

# チュートリアル: ILE RPG アプリケーションの保守: iSeries アプリケーション開発に関する新規レベルのサーバー・ツール統合の紹介

## チュートリアルについて

このチュートリアルでは、一連の練習問題、チェックポイント、および実習を行うことによって、iSeries アプリケーション・サーバーの新規レベルのツール統合について説明します。この新規統合開発環境 (IDE) の iSeries サーバー開発機能には、アプリケーション開発ツールセット (ADTS) または連携開発環境 (CODE) からアップグレードするための納得のいく多くの理由があります。

IDE には、サーバー・アプリケーションと e-business アプリケーションを開発するための一貫性のあるインターフェースが備わっているため、学習期間が短縮されます。これにより、アプリケーション開発の新しいレベルに進むことができます。IDE は、エンドツーエンドのアプリケーション開発ライフ・サイクルをサポートする IBM と複数パートナーの最高品質ツールとのツール統合を保証します。

また、IDE には、iSeries RPG、COBOL、C、C++ アプリケーション、および IBM WebFacing ツールによる Web 使用可能アプリケーションの作成、ビルド、および保守のための堅固で、操作が簡単な開発環境が含まれています。

このチュートリアルでは、RPG で作成された給与計算アプリケーションによる典型的な iSeries アプリケーション開発ライフ・サイクル・タスクに従います。一連の練習問題を通して、新規ツールとその使用方法について説明します。

このチュートリアルを終了するまでに、iSeries サーバー・アプリケーション開発における生産性と使用可能性の著しい向上を実現できるようになり、Java、Web、Web サービス、XML などの新規プログラミング・モデルに移行するための技術が向上します。

このチュートリアルを完了するには、約 2 時間掛かります。

練習問題は順序どおりに行います。練習問題に挿入されている図は、同様のタスクを示しています。名前とアイコンの一部は、練習問題を完了したときに操作している環境と異なる場合があります。

それぞれの練習問題の後でチェックポイント・クイズを行うことをお勧めします。それぞれのクイズを行うことで、練習問題の内容を習得し、次の練習問題に進む準備ができているかどうかを判断することができます。クイズの答えを確認できるように、答えがこのチュートリアルの最後に付いています。

それぞれの練習問題チェックポイントの後に、実習セッションがあります。このセッションでは、練習問題で学んだことを応用します。多くの場合、練習問題で学んだことと同じではないが非常によく似たタスクを行うこととなります。実習のタスクを完了するために、**Development Studio Client for iSeries** のオンライン・ヘルプが役立つことがあります。

チュートリアルは PDF フォーマットで入手できます。PDF バージョンは、**WebSphere Development Studio Client for iSeries** 製品 Web サイト ([ibm.com/software/adwtools/iseries](http://ibm.com/software/adwtools/iseries)) から表示できます。

## チュートリアルのビジネスの問題

あなたは **iSeries RPG** アプリケーション開発者であり、**Microsoft Windows** の基本操作 (デスクトップの操作など) およびマウスの基本操作 (フォルダーのオープンやドラッグ・アンド・ドロップ操作の実行) に精通しています。現在 **iSeries** アプリケーション開発ツール (**ADTS**) 製品 (**PDM**、**SEU**、**SDA**) または連携開発環境 (**CODE**) を使用しています。**ADTS** はサーバー・サイド **iSeries** アプリケーションを開発および保守するための従来のメソッドでした。しかし、現在では **iSeries RPG**、**COBOL**、**C**、**C++**、**CL**、および **DDS** の開発用に、高度に統合された、拡張性の高い新規のツールセットが存在します。この新規ツールは、**Java**、**Web**、**Web** サービス、および **XML** アプリケーションの開発経験と一貫性のある開発経験を提供します。この新規ツールを使用すれば、非常に高度な編集や **DDS** 設計のサポートのためにクラシックな **CODE** ツールを活用することもできます。この新世代のツールは、ファイル・システムの探索、コンパイル/ビルド、編集、実行、およびデバッグのためのさまざまなサポートを提供します。生産性と使用可能性の著しい向上、単独およびチームでの開発のサポート、および **IBM** とパートナーが提供するサーバー開発ツールの緊密な統合のための共通ツールが提供されます。

これは、あなたとあなたの会社が特に **Web** を介して **iSeries** アプリケーションを使用するときに望むことです。**iSeries** アプリケーションを新しくする場合の最初のステップは、次世代の **iSeries** サーバー・アプリケーション開発ツールに移行することです。ここには、この次世代ツールについてあなたとあなたの会社を知る内容があります。

最初のサーバー・アプリケーション開発ツールは、独自のツールとビューを持つリモート・システム・エクスプローラーです。リモート・システム・エクスプローラーは、**QSYS** ファイル・システムにドリルダウンしたり、フィルターを使用して **QSYS** ファイル・システム内の特定のオブジェクトをリストしたりすることができるという点で、プログラミング開発マネージャー (**PDM**) によく似ています。

ただし、リモート・システム・エクスプローラーは **PDM** よりはるかに優れています。また、**iSeries** ジョブとコマンド、および **IFS** ファイル・システムを探索することもできます。さらに、リモート・システム・エクスプローラーを使用すれば、リモートの **Linux**、**UNIX**、および **Windows** システムのファイル・システムを探索できます。

「リモート・システム」ビューは、PDM と同様に 1 次ドリルダウン・ビューです。メンバーをダブルクリックすると、リモート・システム LPEX エディターがオープンし、IDE 内にビルドされ、高度な編集機能を提供します。このエディターは SEU の機能よりはるかに優れていて、CODE エディターのすべての機能に近いものとなっています。構文検査機能が SEU から移植され、コンパイラがエラーの検査用に組み込まれ、参照マニュアルが組み込まれます。このマニュアルは F1 カーソル依存です。プログラム階層は「アウトライン」ビューで参照できます。コンテンツ・アシストは一般的なフィーチャーです。明示的で優れた iSeries 特定のサポートが、RPG、COBOL、C、C++、CL、および DDS の編集のためにだけでなく、それらの検査、コンパイル、実行、およびデバッグのためにも用意されています。これらの操作は、リモート・システム LPEX エディターからだけでなく、独自のリモート・システム・エクスプローラー・パースペクティブ (iSeries サーバー・アプリケーション開発に特定のビューおよびツール用の場所) 内の IDE からも行われます。

PDM ユーザーは「リモート・システム」ビュー (ツリー・ビュー) よりもテーブルに慣れているため、ツリーが表す内容をテーブル・フォーマットで示す iSeries テーブル・ビューが用意されています。列見出しをクリックすれば、列をソートできます。「リモート・システム」ビューでファイルをクリックすれば、テーブルの内容が簡単に置換されます。右クリック・アクションは PDM の右クリック・アクションと同じで、テーブルの下部にあるコマンド行は PDM の場合と同様です。

表示装置ファイルおよび印刷装置ファイルの開発用組み込みツールはまだ用意されていませんが、「リモート・システム」ビューから CODE 設計機能を簡単に起動することができます。

RPG プログラムのデバッグは、リモート・システム・エクスプローラー対話式デバッグ・パースペクティブから簡単に実行できます。このパースペクティブを介して、Java、OPM/ILE、RPG、COBOL、CL、ILE C、および C++ のプログラムをデバッグできます。

この情報を考慮すれば、先に進むことができます。

## 開始する前に

開始する前に、以下をインストールする必要があります。

iSeries:

- OS/400 バージョン 5 リリース 1 以降。リモート・システム・エクスプローラーには、追加の PTF が必要となります。PTF 情報は、[ibm.com/software/adwtools/iseries](http://ibm.com/software/adwtools/iseries) から入手できます。このチュートリアルには iSeries サーバー上のプログラミング・オブジェクトが含まれていて、アプリケーションには iSeries サーバー上のバックエンド・コードが含まれているため、OS/400 が必要となります。
- IBM WebSphere Development Studio for iSeries (5722-WDS)

- IBM WebSphere Development Studio for iSeries には V5R2M0 OS/400 (5722-SS1) が必要です。
- iSeries に復元される RSELAB.savf
- コマンド STRHOSTSVR \*ALL および STRTCPSVR \*DDM で開始される iSeries システム・サーバー

iSeries サーバー・アプリケーション開発用ワークステーション:

- Intel Pentium II プロセッサ
- 最小 256 MB RAM
- 必要ハード・ディスク・スペース: 1650 MB
- 他に 700MB の一時ハード・ディスク・スペース (製品インストール時に必要)
- Windows: VGA グラフィックス・カード (800 x 600 以上推奨、256 色)
- CD-ROM/DVD ドライブ
- マウスなどのポインティング装置

ソフトウェア

- IBM WebSphere Development Studio Client for iSeries バージョン 5.1
- 任意の使用可能な PTFPTF 情報は、[ibm.com/software/adwtools/iseries](http://ibm.com/software/adwtools/iseries) から入手できます。
- eNetwork パーソナル・コミュニケーション・バージョン 5.5 以降
- iSeries システムへの TCP/IP アクセス
- Windows 2000 Professional SP2<sup>®</sup> 以上
- Windows XP Professional SP1<sup>®</sup> 以上
- Microsoft™ Internet Explorer 5.5 SP1 以上

このチュートリアルライブラリーを使用するには、そのライブラリーを iSeries に復元する必要があります。

iSeries で保管ファイルを作成します。

```
CRTSAVF <library_name>/RSELAB
```

ワークステーションでコマンド・プロンプト・ウィンドウをオープンし、保管ファイルをダウンロードしたディレクトリーに移動して以下のようにします。

```
ftp <iSeries_name>  
<user_name>  
<password>  
binary  
put rselab.sav <library_name>/RSELAB  
quit
```

iSeries で RSELAB ライブラリーを復元します。

```
RSTLIB SAVLIB(RSELAB) DEV(*SAVF) SAVF(<library_name>/RSELAB)
```

ライブラリーを復元したら、保管ファイル <library\_name>/RSELAB を削除できます。

## このチュートリアルで使用される規則

このチュートリアルのテキストでは、制御名と入力テキストを区別できるように活版印刷の規則が使用されます。たとえば、以下のようになります。

- メニュー項目は太字フォントです。  
「**メニュー**」 → 「**メニュー**選択」をクリックします。
- フィールド、チェック・ボックス、およびボタンの名前も太字フォントです。  
「**フィールド**」 フィールドにテキストを入力します。
- 入力テキストは、新規行にサンプル・フォントで表されます。

```
This is the text that you type.
```

## 関連情報

このチュートリアルでは、リモート・システム・エクスプローラーを使用して実行することができる最も一般的なタスクを扱います。関連タスクの詳細については、以下の文書を参照してください。

### WebSphere Development Studio Client for iSeries

- 「iSeries アプリケーションの保守と開発」 オンライン・ヘルプ
- ライブラリー・ページ ([ibm.com/software/adwtools/series](http://ibm.com/software/adwtools/series)) にあるリモート・システム・エクスプローラーのプレゼンテーション
- ライブラリー・ページ ([ibm.com/software/adwtools/series](http://ibm.com/software/adwtools/series)) にある iSeries アプリケーション開発クイック・ツアー



## 練習問題 1: WebSphere Development Studio とリモート・システム・エクスプローラーの紹介

この練習問題で、IBM WebSphere Development Studio for iSeries (Development Studio) 製品、およびその製品と IBM WebSphere Development Studio Client for iSeries との関係を紹介しします。ホスト・コンポーネントを構成する製品と、ワークステーション・コンポーネントを構成する製品について学びます。iSeries プログラマー用の Development Studio Client とともにサーバー・アプリケーション開発ツールが含まれていることに気がきます。その際、リモート・システム・エクスプローラー (iSeries サーバー・アプリケーション開発ツールの起動ポイント) を紹介しします。

この練習問題を終了すると、以下のことができるようになります。

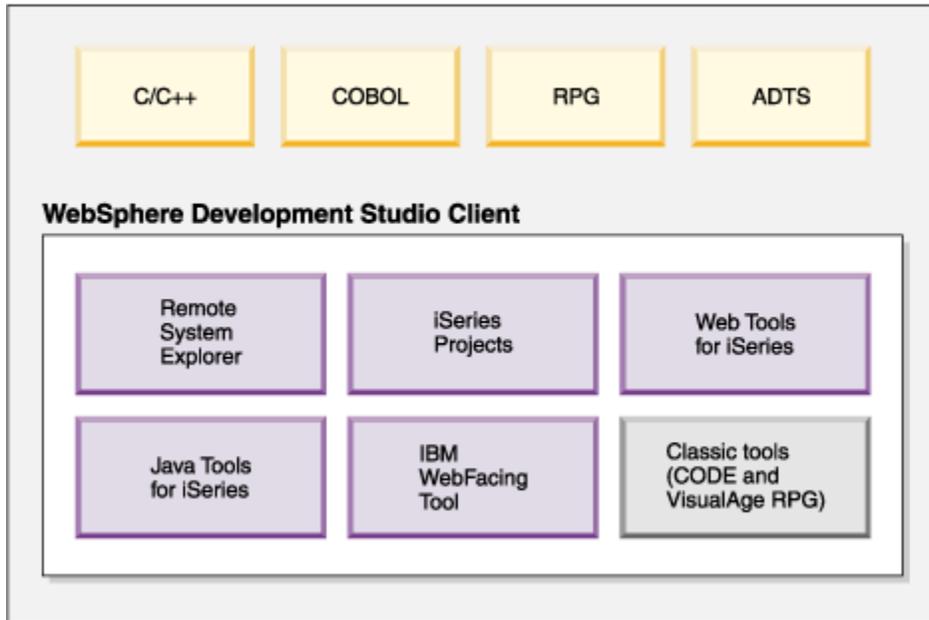
- Development Studio 製品を説明する。
- Development Studio Client 製品を説明する。
- どのようにして Development Studio Client が WebSphere Studio 製品ファミリーに適合するのかを説明する。
- iSeries アプリケーション開発のために iSeries プログラマーが使用できるツールについて説明する。

最初に、Development Studio 製品および Development Studio Client 製品を構成する内容について説明しします。

### Development Studio および Development Studio Client

Development Studio Client を使用すれば、iSeries システムの従来の e-business アプリケーションを素早く開発および展開できます。ご使用のホスト開発に Development Studio を使用する場合は、この強力なツール・スイートの無制限ライセンスを受けていることとなります。次の図は、ホスト・ツールとクライアント・ツールの間の対話を示しています。

## WebSphere Development Studio



Development Studio Client は、以下の作業を行う場合に役立つように設計されています。

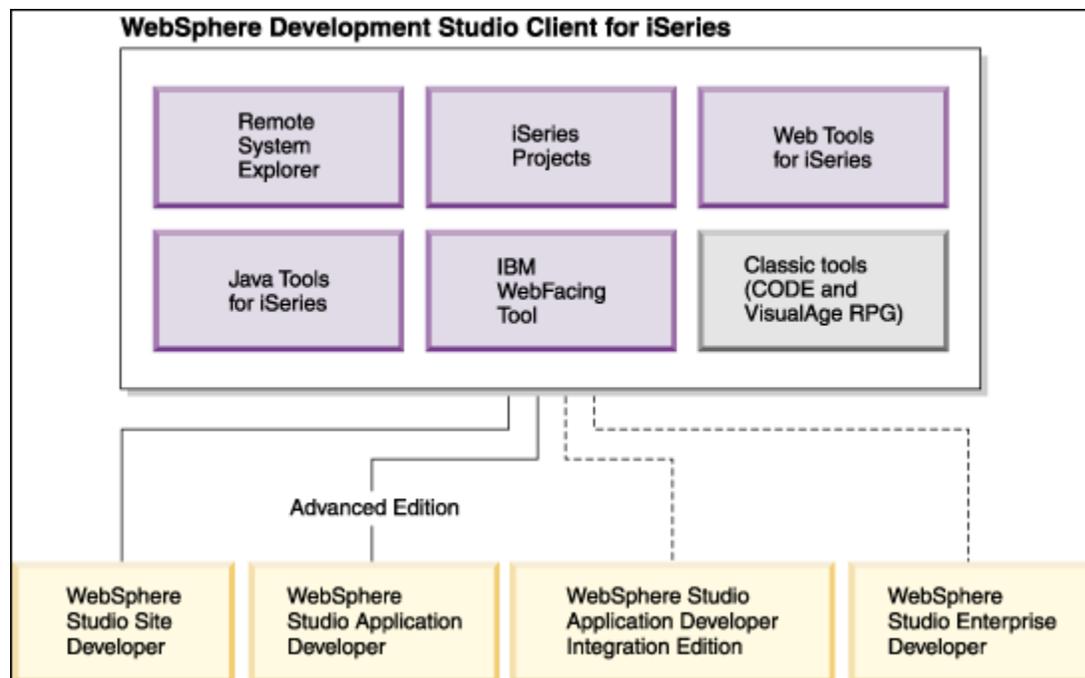
1. リモート・システム・エクスプローラーを使用して **iSeries** アプリケーションを開発および保守する。
2. IBM WebFacing ツールとその他の Web ツールを使用して **iSeries** クラシック・アプリケーション用の Web GUI を開発する。
3. Java ツールを使用して **iSeries** 用のクライアント・サーバー・アプリケーションを開発する。
4. 他の統合サイト開発ツール (XML、Web サービス、SQL、リレーショナル・データベース) を操作する。

これまでに、Development Studio Client がワークステーション・ツールを構成し、Development Studio がホスト・ツールを構成することを学びました。しかし、それだけではありません。WebSphere Studio は、アプリケーションと Web を開発するための IBM のソリューションです。どちらのバージョンの IBM 製品も WebSphere Studio の追加基本製品に同梱されています。

- IBM WebSphere Development Studio Client for iSeries には、WebSphere Studio Site Developer が含まれています。Site Developer には、Web サービスのサポート、XML 開発ツール、および Java と Web の開発ツールのコア・サポートが含まれています。
- **Advanced** IBM WebSphere Development Studio Client Advanced Edition for iSeries には、WebSphere Studio Application Developer が含まれています。この基本製品は、J2EE アプリケーションおよび Web サービスを作成および保守するためのエンドツーエンドのサポー

トを提供します。この製品は、Enterprise Java Bean および Java メッセージ・サービスの拡張サポートも提供します。

さらに、他の WebSphere 製品がインストールされている場合は、この製品のどちらかの版を WebSphere Studio Application Developer - Integration Edition または WebSphere Studio Enterprise Developer に加えてインストールすることもできます。次の図は、この製品が WebSphere Studio 製品ファミリーにどのように適合するのを示します。



## iSeries サーバー・アプリケーション開発ツール

これまでに、Development Studio Client の 2 つの特色とそれぞれの使用理由について学びました。次に、このチュートリアルで最初に述べた次世代の iSeries サーバー・アプリケーション開発ツールについて説明します。そのツールは何か、およびそのツールが何をするのかについて述べます。

### リモート・システム・エクスプローラー

リモート・システム・エクスプローラーでは、開発サイクル・タスクを管理することができます。これは、より柔軟な拡張バージョンのプログラミング開発マネージャー (PDM) です。リモート・システム・エクスプローラーを使用した Windows ベースのワークステーションからの iSeries システムでの開発プロジェクトと iSeries プロジェクトを作成および管理できます。これらのツールを使用すれば、iSeries リモート・ホストに接続し、iSeries ライブラリー、ファイル、およびメンバーを表示することができます。ホスト・コンパイラー、ワークステーション・エディター、プログラ

ム検査プログラム、およびさまざまなデバッガもすべてリモート・システム・エクスプローラーから起動できます。このツールは、UNIX<sup>(R)</sup>、Linux、Windowsなどの他のシステム・タイプもサポートします。

## LPEX エディター

リモート・システム LPEX エディターを使用すれば、プログラム編集タスクが簡素化されます。これは、カスタマイズできる強力な言語依存のエディターです。ソースのトークン強調表示により、さまざまなプログラム・エレメントが明確になります。このエディターには SEU に似た仕様のプロンプトが RPG および DDS 用に用意されていて、桁依存のフィールドに入力するときに役立ちます。RPG、COBOL、および DDS ソースのローカル構文検査と意味検査では、iSeries システムでのクリーン・コンパイルが保証されます。検査エラーが発生した場合、エラー・リストを参照すれば、問題を素早く見つけて解決することができます。オンライン・プログラミング・ガイド、言語解説書、およびコンテキスト依存ヘルプは、必要な情報を見つけ出すときに役立ちます。

## 「リモート・コマンド」ビューでのシェルとコマンド

「リモート・コマンド」ビューを使用すれば、汎用システムでコマンドおよびコマンド・シェルを実行して、コマンドおよびコマンド・シェルと対話することができます。汎用システムには Windows、Linux、および UNIX システム・タイプが含まれます。

特に、このビューは以下の目的で使用します。

- コマンドをコマンド・シェルで実行する。
- プログラムの出力を表示および解釈する。
- プログラムを入力する。
- 同じビューからさまざまなコマンドとシェルを表示および管理する。複数のコマンドを単一シェルで実行し (シェル当たり一度に 1 つのコマンド)、複数のシェルを単一システムで実行し、複数のシステムで複数のシェルを実行することができます。

コマンド・シェルが起動される時、またはコマンドがリモート・システム・エクスプローラーから実行される時は必ず、「リモート・コマンド」ビューに出力が表示され、その出力の操作方法が示されます。

## プログラム検査プログラム

リモート・システム・エクスプローラーの最も強力な固有の機能の1つは、プログラム検査プログラムです。iSeries システムでコードをコンパイルする前に、プログラム検査プログラムを起動して、エラーがないことを確認することができます。検査プログラムはワークステーションで意味 (コンパイル) エラーを検査し、ユーザーが iSeries でクリーン・コンパイルを確実に行うことができるようにします。軽減されるホスト・サイクルについて考えてみてください。特に iSeries システムから切断された状態でコードを作成しているときに、この検査プログラムは役に立ちます。このようなことができるのは、リモート・システム・エクスプローラーがコードの構文解析と検査を iSeries ホスト・コンパイラからワークステーションに移植したためです。「エラー・リスト」ウィンドウでは、検出されたエラーとその重大度がリストされ、エラー・メッセージがソースに直接挿入されます。このウィンドウは、エラーの間をナビゲートするときに役立ちます。

## CODE 設計機能

エディターを使用して表示装置ファイルおよび印刷装置ファイルの DDS ソースを作成および保守する場合、タスクははかどらず、困難になる可能性があります。重要となるのは、画面やレポートを視覚的に設計し、DDS ソースを生成するためのグラフィック設計ツールです。それこそ、CODE 設計機能で実行できる作業です。

CODE 設計機能インターフェースは、DDS 初心者プログラマーが DDS 言語の詳細について考えることなく、画面、レポート、およびデータベースを素早く簡単に作成できるように、また同時に DDS 熟練プログラマーが言語の機能と能力すべてにアクセスできるように設計されました。CODE 設計機能はまだワークベンチに完全には統合されていませんが、個別のツールとしてワークベンチから起動できます。

## iSeries デバッガー

統合 iSeries デバッガーを使用すれば、iSeries システムで実行中のアプリケーションをデバッグできます。このデバッガーには、iSeries プログラムを容易にデバッグおよびテストできるようにする対話式のグラフィカル・インターフェースが用意されています。このデバッガーはワークベンチに完全に統合されています。また、デバッガーを実行する前に、編集集中のソースにブレークポイントを直接挿入することにより、ブレークポイントを設定することもできます。統合 iSeries デバッガー・クライアント・ユーザー・インターフェースでは、プログラムの実行を制御することもできます。たとえば、プログラムの実行、行ブレークポイントの設定、監視ブレークポイントの設定、サービス・エントリー・ポイント・ブレークポイントの設定、プログラム命令のステップスルー、変数の検査、および呼び出しスタックの検査が可能です。異なる言語で作成されている可能性のある複数のアプリケーションを、単一のデバッグ・ウィンドウからデバッグすることもできます。デバッグした各セッションは、「デバッグ」ビューに別々にリストされます。

このチュートリアルでは、ワークベンチ・ツールごとの基本機能と機能についていくつか学びます。Development Studio Client を使用すれば、日々のプログラミング・タスクにおける多くの時間と労力が省かれます。Development Studio Client を使用すれば、より能率的で有能なプログラマーになることができます。同時に iSeries 上でのサイクルが省かれ、さらによいことには、次のステップに進む準備ができて、iSeries アプリケーションが Web に移動されます。

それでは始めましょう。次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

1. WebSphere Development Studio for iSeries:
  - A. 4つのすべてのホスト・コンパイラーと、すべての従来ツール (ADTS) が含まれています。
  - B. 4つのすべてのホスト・コンパイラー、すべての従来ツール (ADTS)、およびワークステーション・ベースのツール Development Studio Client の無制限ライセンスが含まれています。
  - C. ワークステーション・ベースのツール Development Studio Client のみが含まれています。
  - D. 4つのホスト・コンパイラーのみが含まれています。
2. WebSphere Development Studio Client for iSeries バージョン 5 に含まれるもの:
  - A. WebSphere Studio Site Developer バージョン 5 (e-business 開発用)
  - B. 連携開発環境 (CODE)
  - C. VisualAge RPG
  - D. Java ツール
  - E. Web ツール
  - F. WebFacing ツール
  - G. 上記すべて
3. WebSphere Development Studio Client for iSeries アドバンスド・バージョン 5 に含まれるもの:
  - A. WebSphere Studio Application Developer バージョン 5 (e-business 開発用)
  - B. 連携開発環境 (CODE)
  - C. VisualAge RPG
  - D. Java ツール
  - E. Web ツール
  - F. WebFacing ツール
  - G. 上記すべて
4. WebSphere Studio Application Developer でサポートされる内容:
  - A. J2EE アプリケーションの作成と保守
  - B. Web サービスの作成と保守
  - C. Enterprise Java Bean
  - D. Java メッセージ・サービス
  - E. 上記すべて
5. WebSphere Studio Site Developer でサポートされるもの:
  - A. Web サービス
  - B. XML 開発ツール
  - C. Java ツール
  - D. Web ツール
  - E. 上記すべて
6. 開発サイクル・タスクを管理できるもの:

- A. リモート・システム・エクスプローラー
  - B. iSeries プロジェクト
  - C. IBM WebFacing ツール
  - D. 上記すべて
7. リモート・システム・エクスプローラーと iSeries プロジェクトを使用して、iSeries ライブラリー、ファイル、およびメンバーを表示できます。また、ホスト・コンパイラー、ワークステーション・エディター、およびさまざまなデバッガーを起動できます。(T, F)
8. プログラミング編集タスクを簡素化するとき使用するもの:
- A. リモート・システム・エクスプローラー
  - B. iSeries プロジェクト
  - C. IBM WebFacing ツール
  - D. LPEX エディター
  - E. 上記すべて
9. エディターは、ワークステーションまたは iSeries システム上のソース・ファイルに直接アクセスすることができます。コンパイルでエラーが発生したときに、コンパイラー・メッセージから、ソースが表示されるエディターにジャンプできます。エディターがオープンし、問題のソース・ステートメントを訂正できるように、カーソルがその問題のソース・ステートメントの位置に置かれます。(T, F)
10. iSeries システムで実行されているプログラムをワークステーションからデバッグするとき使用するもの:
- A. リモート・システム・エクスプローラー
  - B. iSeries プロジェクト
  - C. IBM WebFacing ツール
  - D. LPEX エディター
  - E. 統合 iSeries デバッガー
  - F. 上記すべて
11. 画面やレポートを視覚的に設計し、DDS ソースを生成するためのグラフィック設計ツール:
- A. リモート・システム・エクスプローラー
  - B. CODE 設計機能
  - C. IBM WebFacing ツール
  - D. LPEX エディター
  - E. 統合 iSeries デバッガー
12. iSeries システムでコードをコンパイルする前に、エラーがないことを確認するために起動するもの:
- A. リモート・システム・エクスプローラー
  - B. CODE 設計機能
  - C. IBM WebFacing ツール
  - D. LPEX エディター
  - E. 統合 iSeries デバッガー
  - F. プログラム検査プログラム
13. 「リモート・コマンド」ビューでできること:
- A. コマンドをコマンド・シェルで実行する。

- B. プログラムの出力を表示および解釈する。
  - C. プログラムを入力する。
  - D. 同じビューからさまざまなコマンドとシェルを表示および管理する。
  - E. 上記すべて
14. 統合 iSeries デバッガーでは、プログラムの実行、行ブレークポイントの設定、監視ブレークポイントの設定、サービス・エンタリー・ポイント・ブレークポイントの設定、プログラム命令のステップスルー、変数の検査、および呼び出しスタックの検査が可能です。(T, F)
15. 製品のアドバンスド・バージョンがワークステーションにインストールされていない場合、「スタート」メニューに「アドバンスド」は表示されません。(T, F)

### ここで学んだ内容

この練習問題では、Development Studio と Development Studio Client、どのように Development Studio Client が WebSphere Studio 製品ファミリーに適合するか、Development Studio Client と Development Studio Client Advanced の相違、および Development Studio Client ワークベンチが提供する iSeries サーバー・アプリケーション開発ツールを紹介しました。

次の練習問題では、Development Studio Client を開始して、リモート・システム・エクスプローラーをオープンします。

## 練習問題 2: Development Studio Client の開始とリモート・システム・エクスプローラーのオープン

この練習問題では、ワークベンチ、ワークスペース、パースペクティブ、特にリモート・システム・エクスプローラー・パースペクティブを紹介します。次に、Development Studio Client の開始方法とリモート・システム・エクスプローラー・パースペクティブのオープン方法について学びます。

この練習問題を終了すると、以下のことができるようになります。

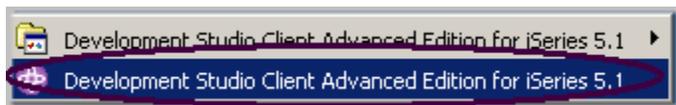
- ワークスペース、ワークベンチ、およびパースペクティブを説明する。
- リモート・システム・エクスプローラー・パースペクティブを説明する。
- Development Studio Client を開始する。
- リモート・システム・エクスプローラー・パースペクティブをオープンする。

まずは Development Studio Client から始めましょう。

### Development Studio の開始

Development Studio Client を開始するには、以下のようにします。

1. デスクトップのタスクバーで「スタート」をクリックします。
2. 「プログラム」->「IBM WebSphere Studio」->「Development Studio Client Advanced Edition for iSeries 5.1」を選択します。



注: 製品の **Advanced Edition** バージョンがワークステーションにインストールされていない場合、「スタート」メニューにワード「**Advanced Edition**」は表示されません。

ウィンドウが表示されることがあります。その場合、ワークベンチで開発するプロジェクトとその他のリソース (フォルダー、サブフォルダー、ファイルなど) が置かれるワークスペースの名前を指定します。

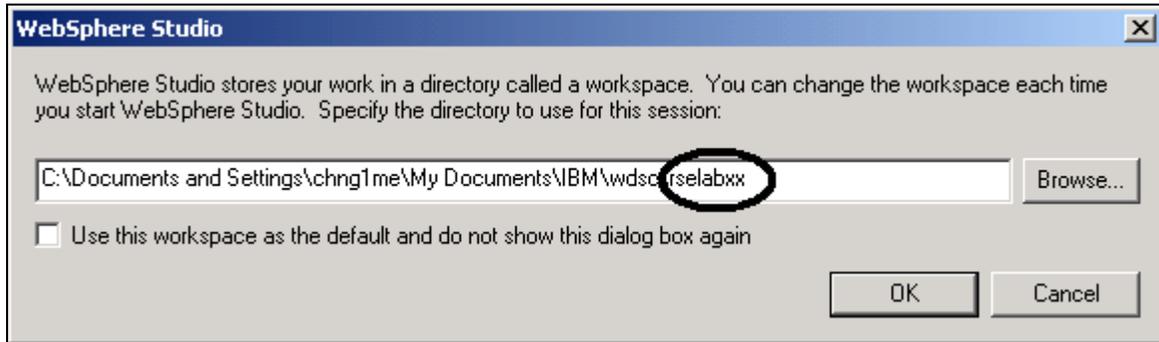


図1: ワークスペース・ディレクトリー名を指定するための「WebSphere Studio」ウィンドウ

3. (任意選択) このウィンドウ内のフィールドを変更して、固有のディレクトリー名 (例: rselabXX (XX は固有の番号)) を使用します。
4. 「**OK**」をクリックします。ロード処理が行われた後に、ワークベンチが表示されます。

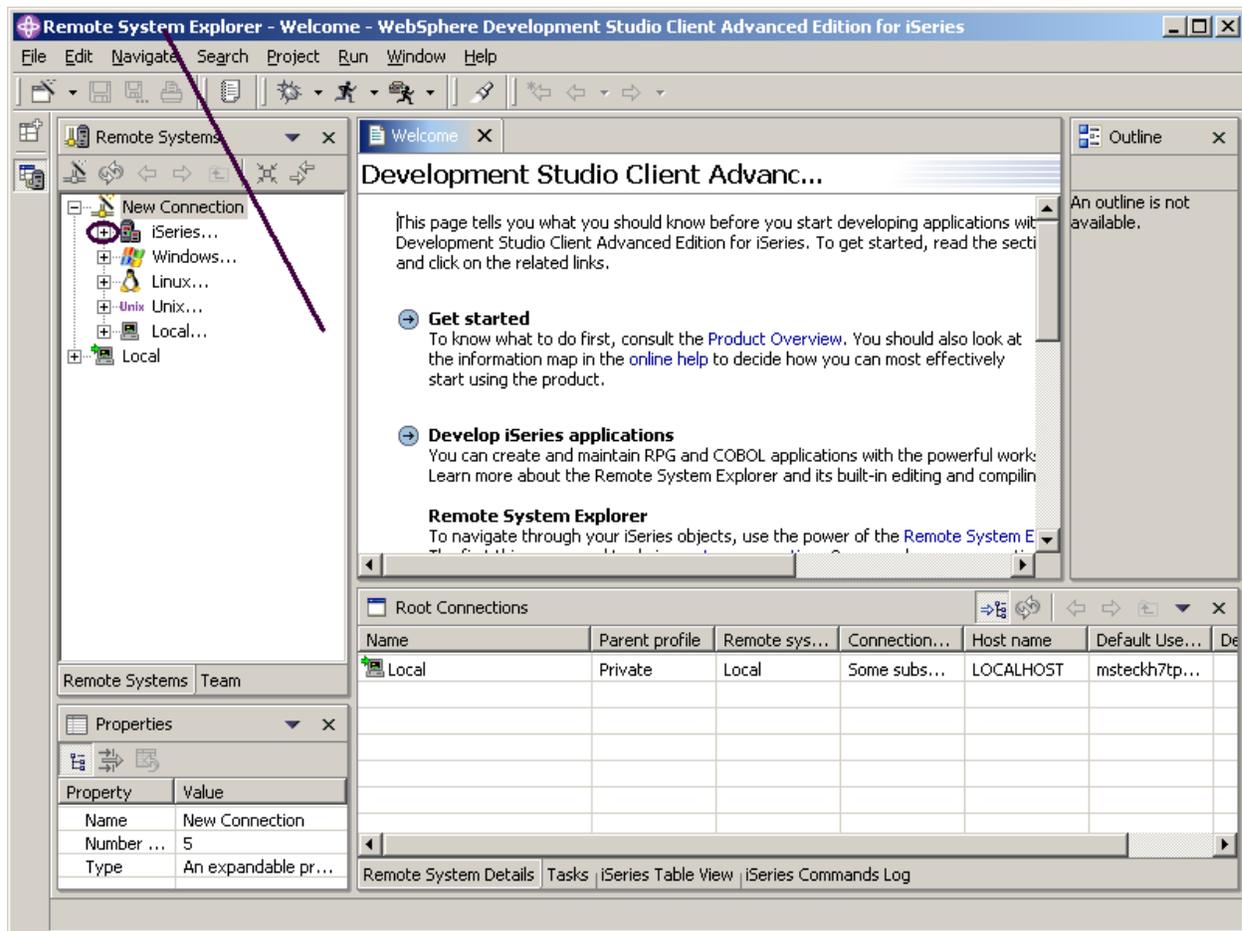


図2: リモート・システム・エクスプローラーでのワークベンチ

ワークベンチは、デスクトップ開発環境に関連しています。ワークベンチの目的は、ワークベンチ・リソースを作成、管理、およびナビゲートするための共通の範例を提供することによって、シームレスなツール統合と制御された開放性を実現することです。ワークベンチ・ウィンドウごとに、1つ以上のパースペクティブが表示されます。

## リモート・システム・エクスプローラー・パースペクティブのオープン

1. パースペクティブの名前を確認します。図2: リモート・システム・エクスプローラーでのワークベンチの矢印は、パースペクティブ名を探す場所を示しています。

パースペクティブは、ワークベンチ・ウィンドウ内のビューの初期セットおよびレイアウトを定義します。このウィンドウ内で、それぞれのパースペクティブは同じエディター・セットを共有します。パースペクティブごとに、特定のタイプのタスクを実行したり、特定のタイプのリソースを処理したりするための機能セットが備わっています。たとえば、Java パースペクティブは、Java ソース・ファイルの編集に共通で使用されるビューを結合し、デバッグ・パースペクティブは、

Java プログラムのデバッグ中に使用されるビューを含みます。パースペクティブはビューとエディターを含み、メニューとツールバーに表示される項目を制御します。

別のパースペクティブを参照する場合、**リモート・システム・エディター**はワークベンチでオープンされず、パースペクティブはありません。

2. ワークベンチ・メニュー・バーから「**ウィンドウ**」を選択します。
3. 「**パースペクティブのオープン**」を選択します。
4. 「**リモート・システム・エクスプローラー**」を選択します。

ワークベンチのリモート・システム・エクスプローラー・パースペクティブで作業します。このパースペクティブは、iSeries プログラマーが、すでに構成された接続を表示し、新規接続を作成し、定義された接続との間で接続および切断を行い、iSeries ファイル、コマンド、ジョブ、および統合ファイル・システム・ファイルを操作するためのものです。このパースペクティブは、新規ワークスペースで **Development Studio Client** を開始したときにアクティブです。以前にワークスペースを使用したことがある場合は、最後にオープンしたパースペクティブでワークベンチが表示されます。これまでに iSeries プログラマー・ツールおよびビューをワークベンチから起動しました。この後の練習問題ではリモート・システム・エクスプローラー・パースペクティブについて詳しく学びます。

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

1. ワークスペース:
  - A. ワークスペースの目的は、ワークベンチ・リソースを作成、管理、およびナビゲートするための共通の範例を提供することによって、シームレスなツール統合と制御された開放性を実現することです。
  - B. ワークベンチ・ウィンドウ内のビューの初期セットおよびレイアウトを定義します。
  - C. デスクトップ開発環境に関連しています。
  - D. ワークベンチで開発するプロジェクトとその他のリソース (フォルダー、サブフォルダー、ファイルなど) が置かれる場所を指定します。
2. ワークベンチ:
  - A. ワークスペースの目的は、ワークベンチ・リソースを作成、管理、およびナビゲートするための共通の範例を提供することによって、シームレスなツール統合と制御された開放性を実現することです。
  - B. ワークベンチ・ウィンドウ内のビューの初期セットおよびレイアウトを定義します。
  - C. デスクトップ開発環境に関連しています。
  - D. ワークベンチで開発するプロジェクトとその他のリソース (フォルダー、サブフォルダー、ファイルなど) が置かれる場所を指定します。

## E. A と C

3. パースペクティブ:
  - A. ワークスペースの目的は、ワークベンチ・リソースを作成、管理、およびナビゲートするための共通の範例を提供することによって、シームレスなツール統合と制御された開放性を実現することです。
  - B. ワークベンチ・ウィンドウ内のビューの初期セットおよびレイアウトを定義します。
  - C. デスクトップ開発環境に関連しています。
  - D. ワークベンチで開発するプロジェクトとその他のリソース (フォルダー、サブフォルダー、ファイルなど) が置かれる場所を指定します。
4. 以下のパースペクティブとその正しい定義を結び付けなさい。
  - A. Java ソース・ファイルの編集に共通で使用されるビューを結合する。
  - B. Java プログラムのデバッグ中に使用されるビューを含む。
  - C. Web アプリケーションの開発中に使用されるビューを含む。
  - D. iSeries アプリケーションの保守中に使用されるビューを含む。
- A. Java パースペクティブ
  - B. Web パースペクティブ
  - C. リモート・システム・エクスプローラー・パースペクティブ
  - D. デバッグ・パースペクティブ
5. リモート・システム・エクスプローラー・パースペクティブでできること:
  - A. 構成された接続を表示する。
  - B. 新規接続を作成する。
  - C. 定義された接続の接続および切断を行う。
  - D. iSeries ファイル、コマンド、ジョブ、IFS ファイルを操作する。
  - E. 上記すべて

## 実習

リモート・システム・エクスプローラー・パースペクティブのオープン方法を学びました。そこで、**Web** パースペクティブをオープンします。**Web** 開発者に使用できるツールとビューを探索します。次に、**Java** パースペクティブをオープンします。**Java** 開発者に使用できるツールとビューを探索します。この時点で **Java** パースペクティブをオープンしていることになっているので、**Web** パースペクティブをオープンします。別の **Web** パースペクティブをオープンしないように注意してください。

**ヒント:** ワークベンチの左にあるフレーム内で **Web** パースペクティブ・アイコンを探してください。そこで **Java** パースペクティブと **Web** パースペクティブの両方をクローズします。

## ここで学んだ内容

この練習問題では、ワークベンチ、ワークスペース、パースペクティブ、特にリモート・システム・エクスプローラー・パースペクティブを紹介しました。次に、**Development Studio Client** の開

始方法とリモート・システム・エクスプローラー・パースペクティブのオープン方法について学びました。

次の練習問題では、**iSeries** システムへの接続を構成し、**iSeries** システムに接続します。

### 練習問題 3: iSeries システムへの接続の構成および iSeries への接続

この練習問題では、iSeries サーバーへの接続を作成し、リモート・システム・エクスプローラー・パースペクティブを使用してオブジェクトを選択します。接続を作成するためのステップを学びます。次に、ライブラリー・リスト内でライブラリーを見つける方法について学びます。最後に、リモート・システム LPEX エディターでメンバーをオープンします。また、「リモート・システム」ビュー、iSeries テーブル・ビュー、および「アウトライン」ビューなど、いくつかのビューについても学びます。

この練習問題を終了すると、以下のことができるようになります。

- 「リモート・システム」ビューを説明する。
- iSeries システムへの接続を構成する。
- iSeries システムに接続する。
- 接続のセットアップに必要な内容を説明する。
- プロファイルを説明する。
- リモート・システム・エクスプローラー・パースペクティブでオブジェクトを検索する。
- サブシステムを説明する。
- オブジェクトを表示し、オブジェクトにアクセスする。
- 「アウトライン」ビューを説明する。
- iSeries テーブル・ビューを説明する。
- iSeries テーブル・ビューと「リモート・システム」ビューの相違を説明する。
- iSeries テーブル・ビューにソース物理ファイル・メンバーを表示する。
- iSeries テーブル・ビューをロックおよびアンロックする。
- 別のソース・メンバーをオープンする。
- ファイルのアウトラインを表示する。

### iSeries システムへの接続の構成

初めてリモート・システム・エクスプローラーをオープンしたときは、ワークステーション上のローカル・ハード・ディスク以外のどのシステムにも接続されていません。リモートの iSeries システムに接続するには、接続を定義する必要があります。接続を定義するときは、リモート・システムの名前または IP アドレスを指定し、ワークスペース内でラベルとして機能する固有の名前を接続に付けて、容易に接続および切断できるようします。iSeries システムに接続するとき、対象のホストにおけるユーザー ID とパスワードを要求するプロンプトがワークステーションから出されます。

初めて iSeries システムに接続するときは、プロファイルを定義する必要があります。すべての接続、フィルター、およびフィルター・プールはプロファイルに属します。フィルターについては、後の練習問題で説明します。プロファイルについては、最初の接続を作成するときに説明します。

それでは、始めましょう。前の練習問題で、既にリモート・システム・エクスプローラー・パースペクティブをオープンしています。

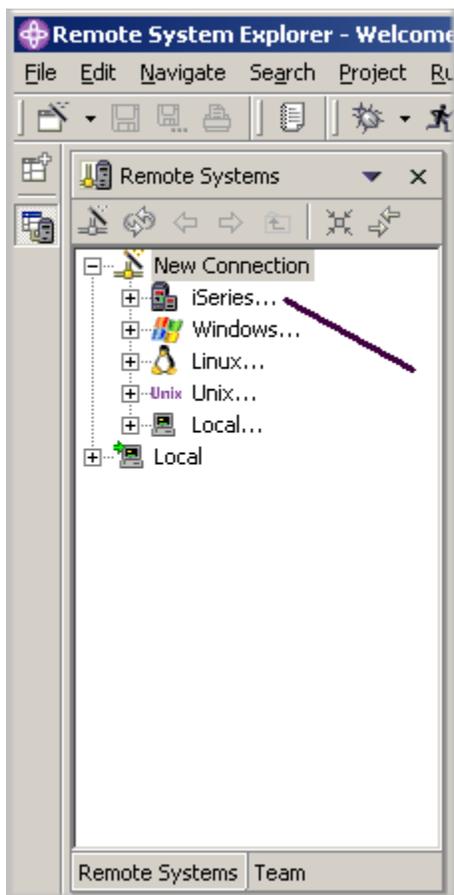


図3: 接続の構成

1. リモート・システム・エクスプローラー・パースペクティブに切り替えます。
2. 「リモート・システム」ビューでは、リモート・システム・エクスプローラーを介して接続できるさまざまなリモート・システム・タイプを表示するように、「**新規接続**」が自動的に拡張されます。

「**iSeries**」を拡張して **iSeries** システムへの接続を構成します。

新規接続ウィザードの「個人用プロファイルの命名」ページが表示されます。

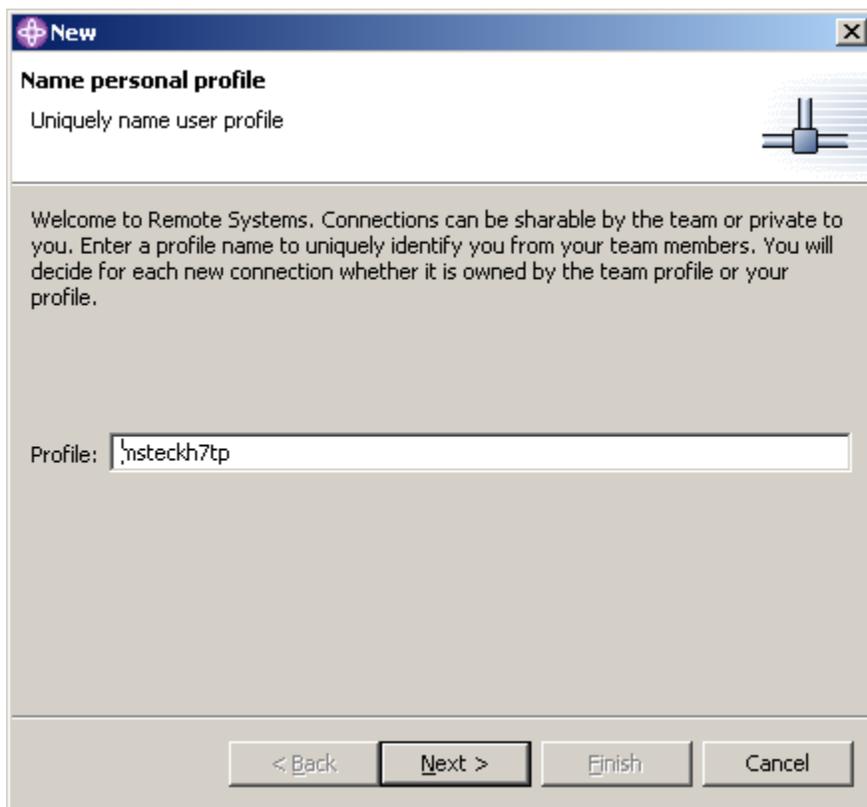


図4: 「新規 – 個人用プロファイルの命名」 ページ

新規接続ウィザードのこのページでは、iSeries システムへの初めての接続であるため、プロファイルを定義する必要があります。すべての接続、フィルター、およびフィルター・プールはプロファイルに属します。プロファイルは接続をグループ化する方法、接続を共有する方法、または接続をプライベートにしておく方法を提供し、多くの接続またはフィルター・プールがある場合にデータを区分できるようにします。最初のプロファイルはローカル・ワークステーション用です。これは個人用プロファイルです。このプロファイルは、接続をプライベートにしておきたいときに使用します。リソースおよび情報を他のユーザーと共有しません。

3. 「プロファイル」情報はそのままにしておきます。「次へ」をクリックします。

新規接続ウィザードの「リモート iSeries システム接続」ページがオープンします。

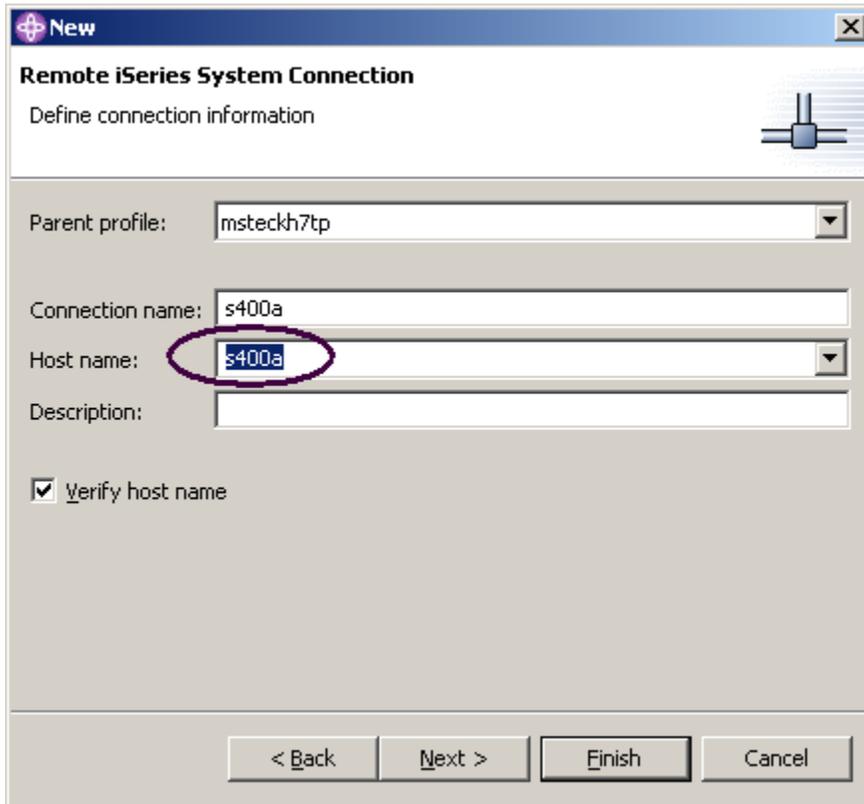


図5: 「接続情報の定義」 ページ

この 2 番目のページで接続の情報を指定します。このページ上のカーソルは「ホスト名」フィールドの位置にあります。

4. ホスト・システムの名前を「ホスト名」フィールドに入力します。

「**接続名**」には、ホスト名が自動的に入力されます。このままにしておきます。この名前は「リモート・システム」ビューに表示されます。これは、プロファイルに固有でなければなりません。

5. 「**親プロファイル**」はそのままにしておきます。この情報を変更する必要はありません。ファイルはプライベートのままであるため、親プロファイルはデフォルトで、前のウィザード・ページで示されたデフォルト・プロファイルになります。リソースを共有している場合は、以前に作成したチーム・プロファイルを選択する必要があります。
6. 「**ホスト名の検査**」チェック・ボックスにチェックマークを入れたままにしておきます。

7. 「終了」をクリックして、システムを定義します。

## iSeries システムへの接続

iSeries システムへの接続を構成した後は、容易に新規接続に接続し、それを拡張して、サブシステムを表示することができます。サブシステムは、リモート・システム内で探査できるさまざまなタイプのリモート・リソースの機能グループです。サブシステムには、以下の 4 つがあります。

- **iSeries** オブジェクトは **PDM** に似たグループであり、ライブラリー、オブジェクト、およびメンバーへのアクセスを可能にします。
- **iSeries** コマンドを使用すれば、頻繁に使用される 1 つ以上のコマンドを含むコマンド・セットを事前定義できます。コマンド・セット内のすべてのコマンドは実行されると、リモート・システムに送信されて実行され、その結果が「コマンド」ビューに記録されます。
- **iSeries** ジョブを使用すれば、ジョブ属性によってサブセット化されたさまざまなジョブを参照し、限られた数の操作をそのジョブで実行できます。
- **IFS** ファイルを使用すれば、リモート **iSeries** システムの統合ファイル・システム内のフォルダーおよびファイルを探査できます。

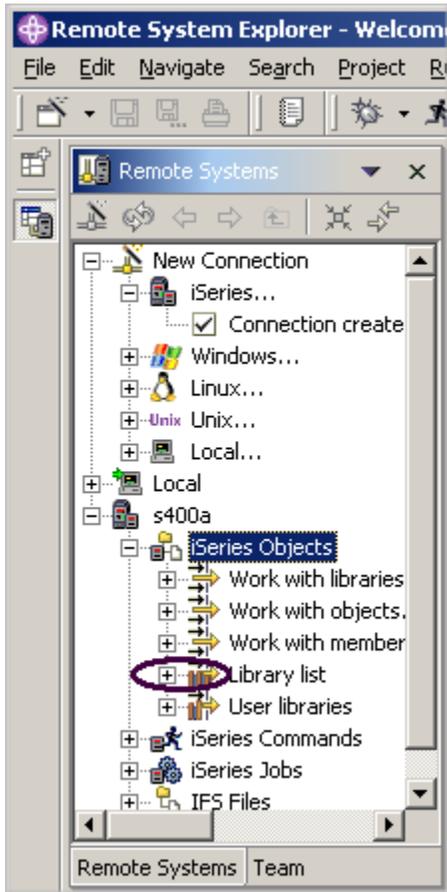


図6: 「リモート・システム」ビューと iSeries 接続

iSeries システムに接続するには、以下のようにします。

1. 「リモート・システム」ビューで、新規接続がサブシステムを表示するように拡張されています。最も頻繁に使用するサブシステムは **iSeries オブジェクト・サブシステム** です。このサブシステムは、**QSYS** ファイル・システム内のオブジェクトにアクセスすることが可能であり、そのオブジェクト上でアクションを実行できるという点で、**PDM** によく似ています。
2. 「iSeries オブジェクト」サブシステムの下にある最初の **3** つのエントリは、**PDM** オプションの名前から名付けられます。これは、機能がよく似ているためです。
  - ライブラリーの操作 (WRKLIBPDM と同様)
  - オブジェクトの操作 (WRKOBJPDM と同様)
  - メンバーの操作 (WRKMBRPDM と同様)

この他に、ライブラリー・リストおよびユーザー・ライブラリーを処理するためのエントリーがあります。

- ライブラリー・リスト (PDM STRPDM オプション 12 をシミュレートするには、事前定義されたライブラリー・リスト・フィルターから開始できます。これを拡張すれば、ライブラリー・リストにすべてのライブラリーが表示されます。)
- ユーザー・ライブラリー

接続自体の下には、操作できるエントリーが他にもあります。そのエントリーから、リモート・システム・エクスプローラーが PDM よりはるかに優れていることが分かります。このため、iSeries のジョブとコマンド、および IFS ファイル・システムを探索することができます。

これから、ライブラリー・リスト内のライブラリーを操作しましょう。

### 3. 「ライブラリー・リスト」の横のプラス符号 (+) をクリックします。

接続が活動化され、ユーザー ID とパスワードを要求するプロンプトが出されます。



図7: 「パスワードを入力」ウィンドウ

4. ユーザー ID とパスワードを入力します。
5. 「ユーザー ID を永続的に変更する」チェック・ボックスにチェックマークを入れます。
6. 「パスワードを保管する」チェック・ボックスにチェックマークを入れます。
7. 「OK」をクリックします。

ユーザー ID が設定され、ライブラリーが自動的にライブラリー・リストに追加されたことを確認します。iSeries コマンドのプロパティを使用すれば、ライブラリーをライブラリー・リストに追加するといった接続情報を設定できます。

注: ライブラリー・リスト内に「RSELABxx」が表示されない場合は、「ライブラリー・リスト」を右クリックして「ライブラリー・リスト・エントリー」ウィンドウを表示し、ライブラリー・リストに RSELABxx を追加します。

リモート・システム・エクスプローラー・パースペクティブ内のワークベンチに戻れば、ジョブのライブラリー・リストにそのライブラリーが表示されています。

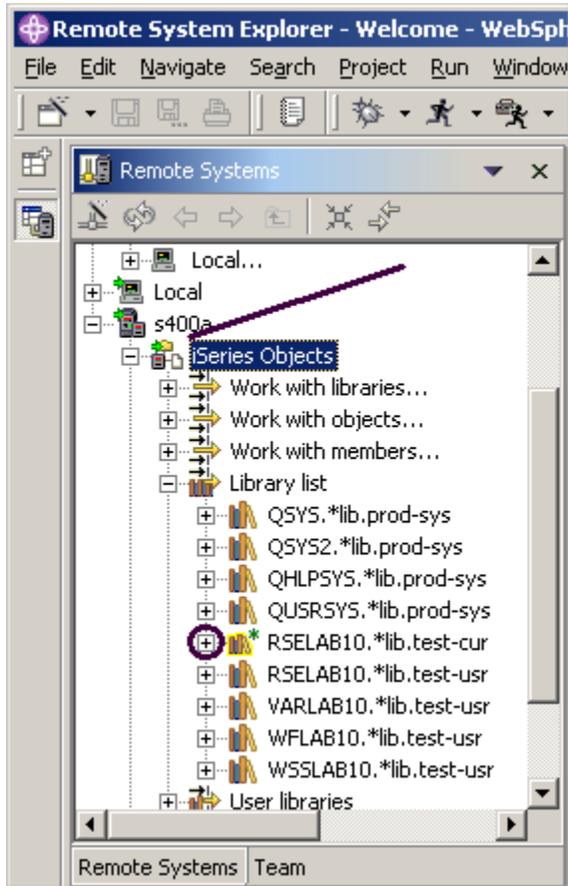


図8: 「リモート・システム」ビューと、ライブラリー・リスト内のライブラリー

現在 s400a ノードのアイコン内に、接続がアクティブであることを示す小さな緑色の矢印があります。

また、「RSELABxx」ライブラリーに、現行ライブラリーであることを示す小さな緑色の星印があります。

ライブラリーごとに、右クリックでいくつかのアクションから選択できます。これらのアクションには、選択されたライブラリー内で新規ソース・ファイルを作成するアクション、ライブラリーが

拡張されている場合にライブラリーの内容をリフレッシュするアクション、ライブラリーの名前を変更するアクション、ライブラリーをコピーするアクション、またはライブラリーを削除するアクションがあります。

## リモート・システム・エクスプローラー・パースペクティブ内のオブジェクトの表示、およびそのオブジェクトへのアクセス

現行ライブラリー RSELABxx 内のオブジェクトを表示し、そのオブジェクトにアクセスする準備を整いました。

オブジェクトを表示するには、以下のようにします。

### 1. ライブラリー「RSELABXX」を拡張します。

このライブラリー内のすべてのオブジェクトが「リモート・システム」ビューに表示されます。オブジェクトごとに、右クリックでいくつかのアクションから選択できます。アクションのリストは、選択されたオブジェクト、および選択されたオブジェクトが 1 つなのか複数なのかにより異なります。ソース・ファイルの場合、ポップアップ・メニューには、選択されたファイル内で新規メンバーを作成するアクション、拡張されている場合にファイルの内容をリフレッシュするアクション、ファイルの名前を変更するアクション、ファイルをコピーするアクション、およびファイルを削除するアクションがあります。これらのアクションは、該当する **iSeries** コマンドをリモートで実行します。コマンドは「コマンド」ビューに記録されます。

### 2. **QDDSSRC** ソース・ファイルが見つかるまで「リモート・システム」ビュー内のファイルをドリルダウンして、そのソース・ファイルを拡張します。

### 3. 同様に **QRPGLESRC** ソース・ファイルが見つかるまで **QDDSSRC** 内のファイルをドリルダウンして、そのソース・ファイルを拡張します。

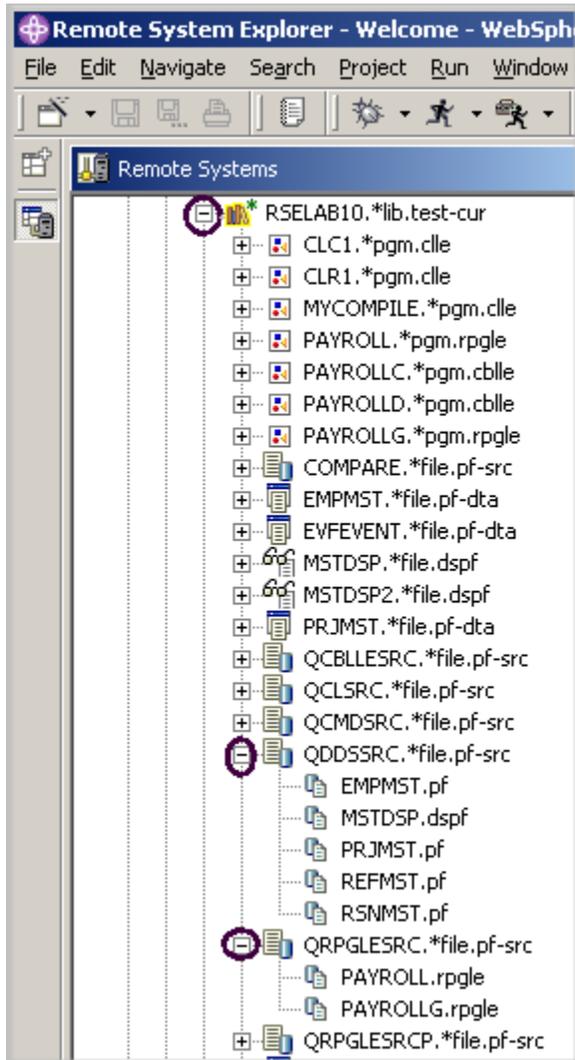


図9: 「リモート・システム」ビューと拡張されたソース・ファイル

現在、これら 2 つのソース・ファイル内のメンバーを参照し、そのメンバーにアクセスすることができます。メンバーごとに、右クリックでいくつかのアクションから選択できます。アクションの正確なリストは、メンバーがデータ・ファイルなのかソース・ファイルなのか、および選択するメンバーが 1 つなのか複数なのかにより異なります。メンバーが 1 つの場合、ポップアップ・メニューのアクションには、編集、名前変更、コピー、移動、削除、およびコンパイルがあります。

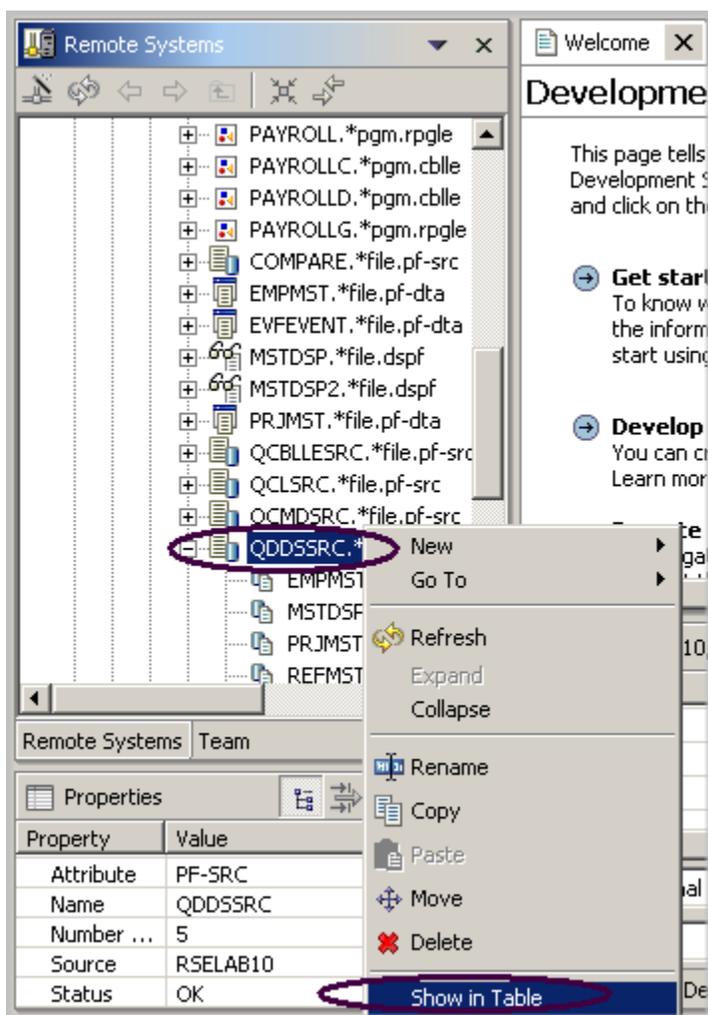


図 10: QDDSSRC ファイルに対する「テーブルへの表示」の選択

先に進んでメンバーを操作する前に、使い慣れた PDM のビューによく似ているので、同じように iSeries テーブル・ビュー内のメンバーも参照しましょう。このビューを使用して、PDM に似たテーブル・フォーマットで項目 (メンバーやオブジェクトなど) のリストを表示します。また、これらの項目に対して編集やコンパイルなどのアクションも実行できます。

#### 4. 「QDDSSRC」ファイルを右クリックして、「テーブルへの表示」を選択します。

iSeries テーブル・ビューは、「リモート・システム」ビュー内の選択オブジェクトを入力として取り入れ、テーブルの内容を表示します。ソース物理ファイルの場合、このステップでは、メンバー、名前、タイプ、属性、テキスト記述、および状況が内部に表示されます。

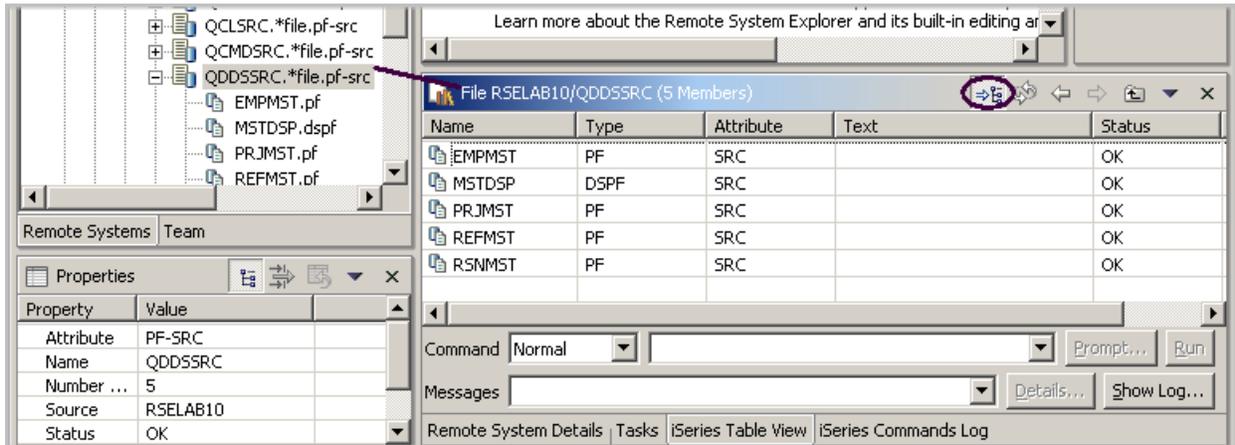


図 11: テーブル・ビューと「ロック/アンロック」ボタン

iSeries テーブル・ビューの上部には、「リモート・システム」ビューと iSeries テーブル・ビューの間の相関関係を制御するロック・アイコン  があります。

ロックが使用不可能になっている場合は、「リモート・システム」ビュー内のオブジェクトまたはライブラリーをクリックすると、該当する項目の関連内容が自動的に iSeries テーブル・ビューから取り込まれます。

ロックが使用可能になっている場合は、「リモート・システム」ビューにあるさまざまな項目をクリックしても iSeries テーブル・ビューへの入力に変更されません。

ロックを使用可能または使用不可能にするには、ロックを一度クリックして、その状態を変更します。

**ヒント:** 列見出しをクリックすれば、ビューが列でソートされます。

この iSeries テーブル・ビューは、アンロックできます。その場合、ビューは、「リモート・システム」ビューでオブジェクトを選択したときのように、「リモート・システム」ビュー内の選択オブジェクトに切り替わります。

5. iSeries テーブル・ビューのツールバーで、「ロック/アンロック」ボタン  がアンロック位置にあることを確認します。これは、別のオブジェクトが「リモート・システム」ビューで選択されている場合、この時点でテーブルが自動的に更新されることを意味します。これは、「リモート・システム」ビューでオブジェクトのポップアップ・メニューを生成し、「テーブルへの表示」を選択するためのショートカットです。

6. 「リモート・システム」ビュー内で「QRPGLESRC」を選択します。テーブルには、QRPGLESRC 内のメンバーが表示されます。

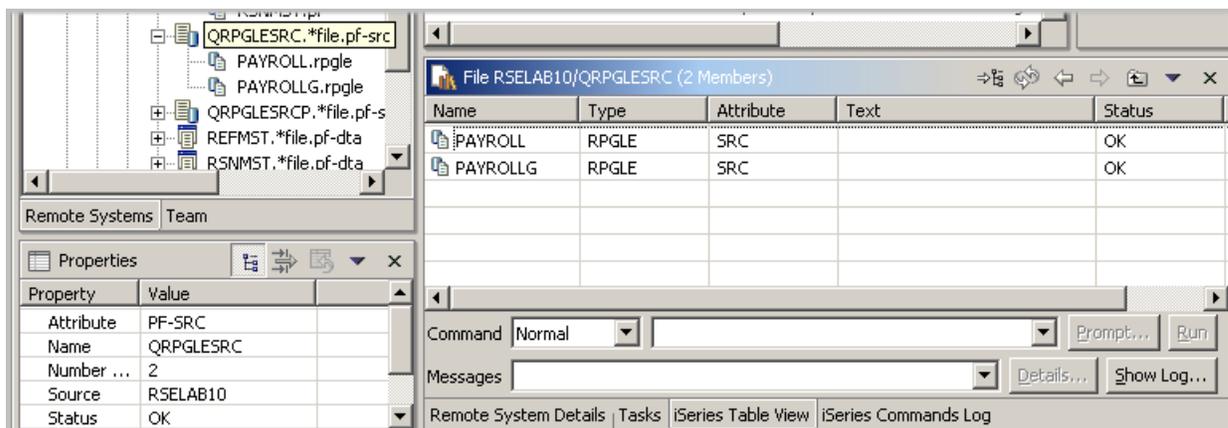


図 12: QRPGLESRC の iSeries テーブル・ビュー

QDDSSRC 内で見つかったメンバー MSTDSP をリモート・システム LPEX エディターで編集する準備が整いました。

7. 「リモート・システム」ビューから QDDSSRC ソース・ファイル内のメンバー「MSTDSP」をダブルクリックします。

**ヒント:** この操作は、「リモート・システム」ビューまたは iSeries テーブル・ビューで実行できます。

リモート・システム LPEX エディターがオープンします。これはワークベンチに組み込まれていて、豊富な編集機能を持ち、iSeries に対応しています。これは、SEU のスーパーセットでもあります。構文検査機能が SEU から移植され、コンパイラがエラーの検査用に組み込まれ、参照マニュアルが組み込まれます。このマニュアルは F1 カーソル依存です。「アウトライン」ビューに、プログラム階層が表示されます。リモート・システム LPEX エディターから RPG、COBOL、C、C++、CL、および DDS を検証、コンパイル、実行、およびデバッグするための明示的で優れた iSeries サポートがあります。

構文検査、自動大文字変換、プログラム検査など、RPG、COBOL、CL、および DDS ソース用に CODE エディターで提供される多くの編集機能が、リモート・システム LPEX エディターで使用可能です。

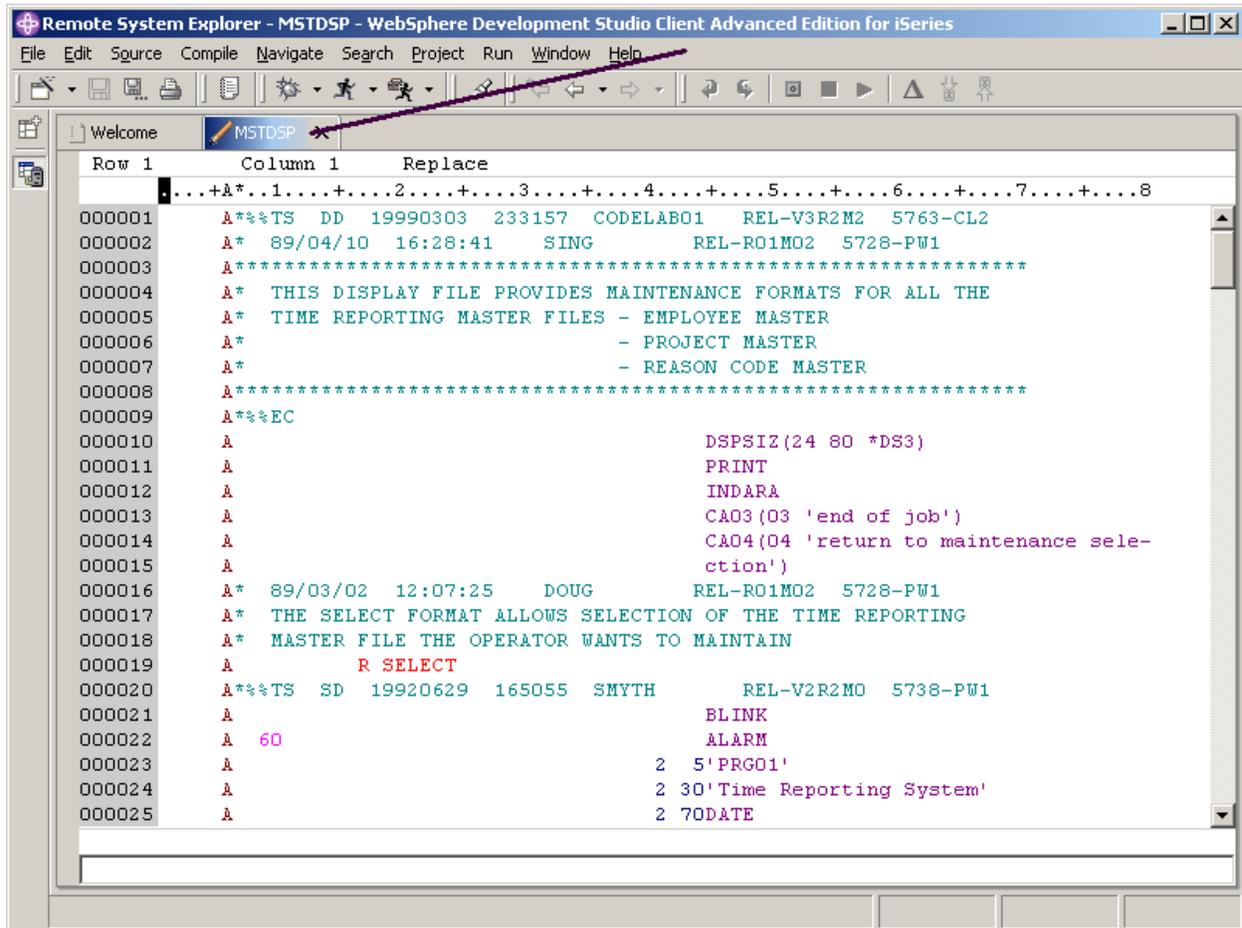


図 13: ソース・エディターと DDS メンバー

8. 図 13: ソース・エディターと DDS メンバー の矢印で示されるように、「MSTDSP」タブをダブルクリックしてエディター・ウィンドウを最大化します。再度「MSTDSP」タブをダブルクリックして、ウィンドウを元のサイズに戻します。

## 別のソース・メンバーのオープン

次に、エディターで別のメンバーをオープンしましょう。「リモート・システム」ビューに戻って、次のようにします。

1. 「QRPGLESRC」ソース・ファイル内のメンバー「PAYROLL」をダブルクリックします。

このメンバーもエディターにロードされます。実際のエディター・ウィンドウは図 14 のようになります。エディター・ペインにある 2 つのタブに注意してください。編集セッションを切り替えるには、それぞれのタブをクリックします。

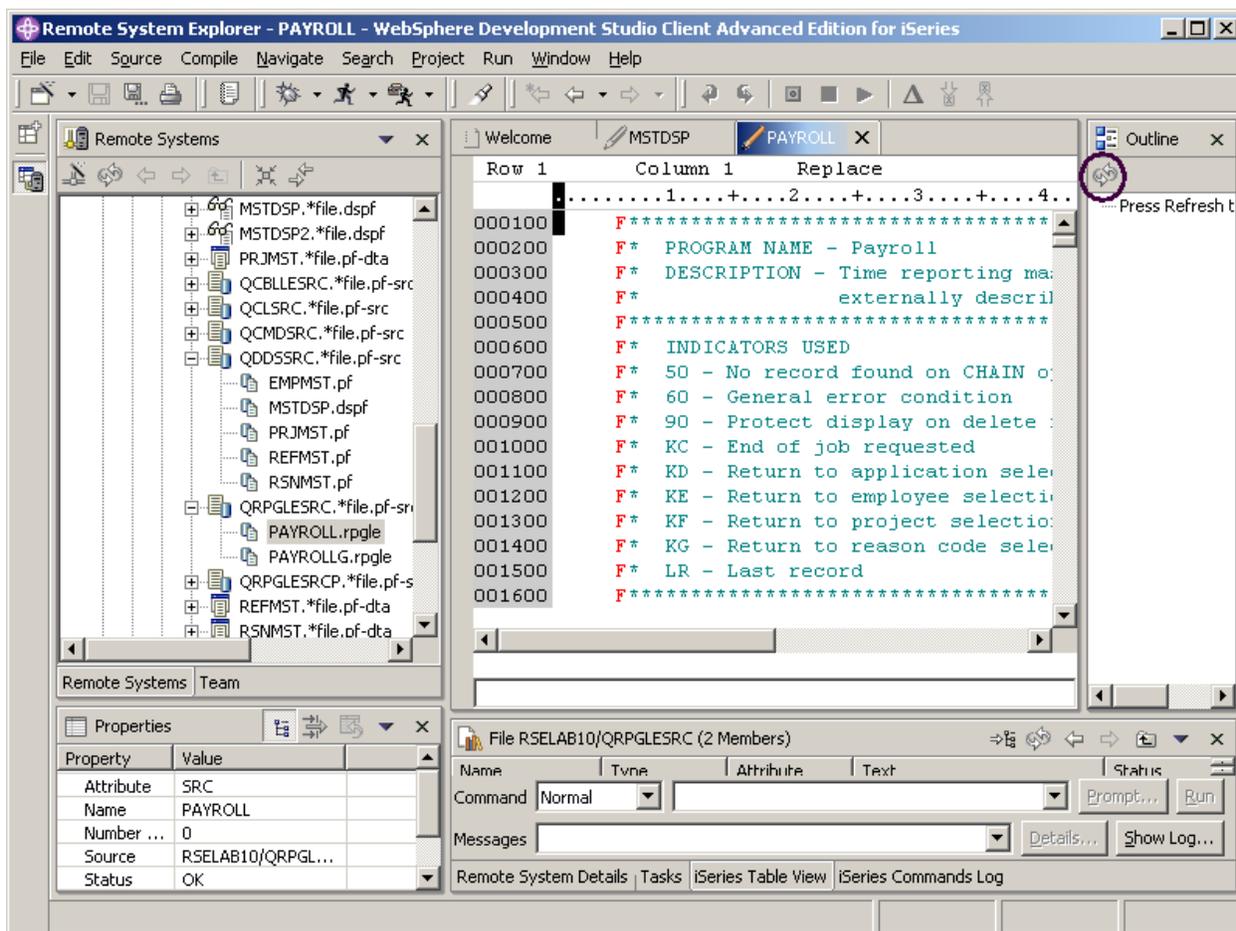


図 14: RPG ソース・メンバーと「アウトライン」ビュー・フレームが表示されたエディター

## 構造化ファイルのアウトラインの表示

次に、「アウトライン」ビューを見てみましょう。このビューには、エディター域で現在オープンされている構造化ファイルのアウトラインが表示され、構造エレメントがリストされます。「アウトライン」ビューおよびツールバーの内容はエディター特定のものです。

RPG ソースの「アウトライン」ビューを参照するには、以下のようにします。

1. 図 14: RPG ソース・メンバーと「アウトライン」ビュー・フレームが表示されたエディターに

丸で示されるように、「アウトライン」ビュー・ツールバーのリフレッシュをクリックします。

「アウトライン」ビューには、ツリー・ビューでソース・プログラムが表示され、ロジックを含む行はありません。

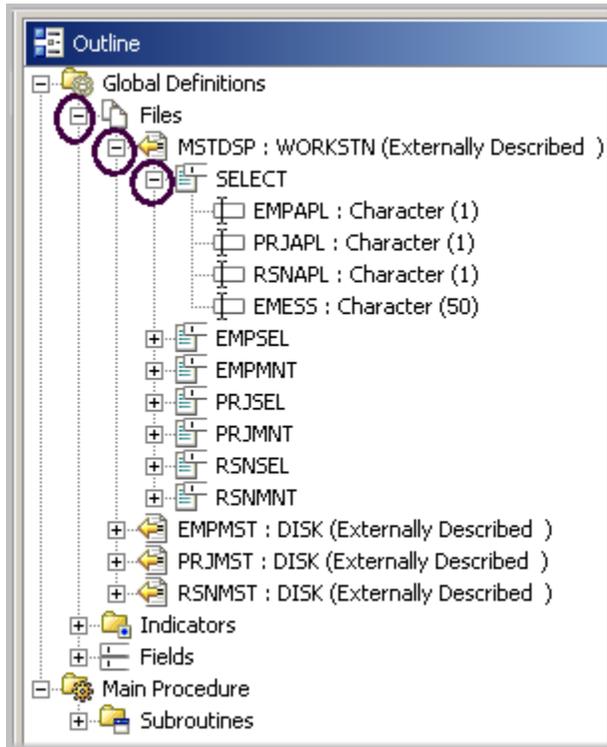


図15: 「アウトライン」ビューと、拡張されたファイルおよびレコード・フォーマット

ここで、ソース・メンバーの詳細を参照します。

2. 「ファイル」を拡張します。
3. 「**MSTDSP**」ワークステーション・ファイルを拡張します。
4. 「**SELECT**」レコード・フォーマットを拡張します。

「アウトライン」ビュー内の任意のエントリをダブルクリックします。これにより、相応の位置にソース・エディターが表示されます。

ここで、ソース・エディター内にロードされた別のメンバーに切り替える場合、そのタブをクリックすれば、対応するメンバーのソース・エディター・ウィンドウがアクティブになります。

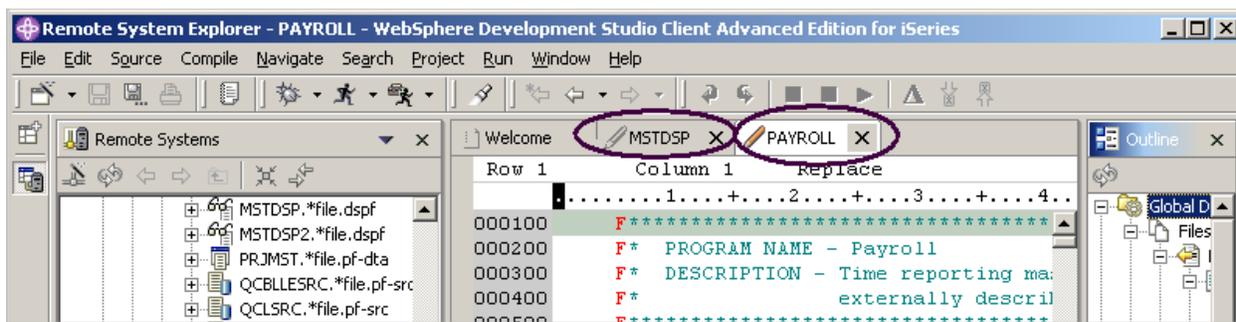


図16: タブをクリックして編集セッションをアクティブにする

ここで、DDS ソース・メンバーに戻ります。

5. 「**MSTDSP**」タブをクリックします。
6. 次の練習問題に備え、「**PAYROLL**」タブをクリックして **PAYROLL** エディター・ウィンドウをアクティブにします。

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

### チェックポイント

1. 初めてリモート・システム・エクスプローラーをオープンしたとき、ローカル・ワークセッション以外のシステムには接続されていません。リモート iSeries システムへの接続には、何が必要ですか。
  - A. リモート・システム・エクスプローラー通信サーバーを開始する。
  - B. 5250 セッションを開始する。
  - C. 接続を定義する。リモート・システムの名前と IP アドレスを指定する。
  - D. プロファイルを定義する。
2. 初めて iSeries システムに接続するときに定義しなければならないもの:
  - A. プロファイル
  - B. フィルター操作
  - C. フィルター・プール
  - D. 接続
  - E. 上記すべて
3. プロファイルで行われること:
  - A. 多くの接続またはフィルター・プールがある場合に、データの区分に役立つ。
  - B. 接続、フィルター、およびフィルター・プールをすべて含む。
  - C. 接続をグループ化する。
  - D. 接続を共用する。

- E. 接続をプライベートにしておく。
  - F. 上記すべて
4. 複数のタイプを持つプロファイル:
- A. チーム
  - B. 個人用
  - C. 両方
5. チーム・プロファイルは、リソースおよび情報を他のユーザーと共有するために使用されます。(T, F)
6. サブシステムに含まれるもの:
- A. iSeries オブジェクト
  - B. iSeries ジョブ
  - C. IFS ファイル
  - D. iSeries コマンド
  - E. 上記すべて
7. iSeries オブジェクトに含まれるもの:
- A. ライブラリーの操作
  - B. オブジェクトの操作
  - C. メンバーの操作
  - D. ライブラリー・リスト
  - E. 上記すべて
8. iSeries テーブル・ビューを使用して行われること:
- A. 項目 (メンバーやオブジェクト) を PDM によく似たテーブル・フォーマットでリストする。
  - B. 編集やコンパイルなどのアクションを項目のリストに対して実行する。
  - C. 両方
9. ロック・アイコンは「リモート・システム」ビューおよび iSeries テーブル・ビューの間の相関関係を制御します。(T, F)
10. 「リモート・システム」ビュー内でオブジェクトまたはライブラリーをクリックしたときに、使用不可ロックが意味する内容:
- A. 対象項目の関連内容は iSeries テーブル・ビューを自動的に取り込む。
  - B. 「リモート・システム」ビューのみが更新される。
  - C. iSeries テーブル・ビューのみが更新される。
  - D. 上記のいずれでもない
11. 「リモート・システム」ビュー内でさまざまな項目をクリックしたときに、使用可能ロックが意味する内容:
- A. 対象項目の関連内容は iSeries テーブル・ビューを自動的に取り込む。
  - B. iSeries テーブル・ビューの入力は変更されない。
  - C. 「リモート・システム」ビューのみが更新される。
  - D. iSeries テーブル・ビューのみが更新される。
  - E. 上記のいずれでもない
12. ウィンドウの見出しをダブルクリックすることにより、エディター・ウィンドウを最大化できます。見出しを再度ダブルクリックすれば、元のサイズに戻すことができます。(T, F)

13. 「アウトライン」ビューで行われること:

- A. エディター内で現在オープンされている構造化ファイルのアウトラインを表示する。
- B. 構造エレメントをリストする。
- C. 内容とツールバーはエディター特定である。
- D. 上記すべて

## 実習

ここでは、自分の **iSeries** システムにアクセスしました。そこで、新規接続を構成し、この **iSeries** システムに接続します。接続を名前変更し、接続を上へ移動し、接続の属性を変更します。次に、**iSeries** オブジェクト・サブシステムを使用して、ライブラリー・リストにライブラリーを表示します。**iSeries** テーブル・ビューを使用して、ライブラリー内のオブジェクトを参照します。**iSeries** テーブル・ビュー内のオブジェクトをサブセット化します。**iSeries** テーブル・ビューからリモート・システム **LPEX** エディターをオープンします。タスクの途中で困ったときは、**Development Studio Client for iSeries** のオンライン・ヘルプを使用してください。

## ここで学んだ内容

この練習問題では、プロファイル、サブシステム、および **iSeries** テーブル・ビューについて学びました。**iSeries** サーバーに対して接続情報を定義し、**iSeries** オブジェクト・サブシステムを選択し、ライブラリー・リストにライブラリーを表示しました。次に、現行ライブラリーを拡張し、**QDDSSRC** の **iSeries** テーブル・ビューを参照しました。**iSeries** テーブル・ビューをロックおよびアンロックしました。次に、リモート・システム **LPEX** エディターで複数のソース・メンバーをオープンしました。エディター・ウィンドウを最大化し、「アウトライン」ビューをオープンし、「アウトライン」ビューから、編集の準備ができたメンバーに切り替えました。

次の練習問題では、ここでオープンしたソース・メンバーを引き続き操作し、リモート・システム **LPEX** エディターでの編集タスクを簡素化する方法について学びます。現在は、「リモート・システム」ビューから編集しています。また、リモート・システム **LPEX** エディターでは、**SEU** に似たフォーマット表示行規則、**SEU** 仕様のプロンプト、および **SEU** スタイルのコマンドを使用できます。



## 練習問題 4: ILE RPG の編集

この練習問題では、「リモート・システム」ビューからリモート・システム LPEX エディターの言語フィーチャーのいくつかを使用して、ILE RPG ソースを編集します。

この練習問題を終了すると、以下のことができるようになります。

- リモート・システム LPEX エディターのフィーチャーを説明する。
- 桁を編集する。
- 接頭部域内で SEU コマンドを起動する。
- 編集の変更を元に戻す、および再実行する。
- RPG プログラミング言語特定のヘルプを起動する。
- 言語特定の情報のプロンプトを出す。
- ソースを字下げする。
- 検索および置換を行う。
- ソースをフィルター操作する。
- 検索ユーティリティーを用いて複数のファイル内でストリングを検索する。
- 比較ユーティリティーを使用して 2 つのファイルを比較する。
- 構文エラーのある行を識別する。
- 構文エラーを訂正する。

リモート・システム LPEX エディターを使用すれば、プログラム編集タスクが簡素化されます。このエディターは、ワークステーションまたは iSeries システム上のソース・ファイルに直接アクセスできます。コンパイルでエラーが発生したときに、コンパイラー・メッセージから、ソースが表示されるエディターにジャンプできます。エディターがオープンし、問題のソース・ステートメントを訂正できるように、カーソルがその問題のソース・ステートメントの位置に置かれます。

以下は、ワークステーション・エディターに求められる基本エディター・フィーチャーの一部のリストです。

- カット、コピー、ペースト
- コピー、移動、および削除操作による行、文字、または長方形のブロック・マーキング
- 強力な検索および置換機能
- 無制限の元に戻すおよび再実行

他に、ワークステーション・エディターには表示されなかった機能がいくつかあります。

- プログラム内で識別しやすくするために、異なる言語構文が異なる色およびフォントを使用して強調表示されている場所でのトークン強調表示

- RPG および DDS など桁依存言語の各桁の目的を示すための、SEU に似たフォーマット行罫線。この罫線は、現行の仕様を反映するように自動的に更新できます。
- SEU に似た、RPG および DDS 用の仕様プロンプト
- シーケンス番号。これにより、接頭部域内で SEU スタイルのコマンドを使用できます。
- 桁依存言語の桁の間での高機能なタブ付け
- 大文字が予期される言語のための自動大文字変換
- 桁依存言語の設定。これにより、テキストの挿入と削除が容易になります。
- オンライン言語参照ヘルプ

ここで、これらのフィーチャーのいくつかを試しましょう。

RPG ソース・メンバー **PAYROLL** (すでにオープンしている) を操作して、下の説明に従ってこれらのフィーチャーの一部を試してください。この練習問題の途中で、エディター・ウィンドウを最大化しなければならない場合があります。

## 桁の編集

LPEX は、桁依存言語での挿入および削除のための特別なサポートを提供します。桁依存編集が選択されると、各桁は別々のエンタリー・スペースと見なされます。たとえば、RPG ソース・メンバーでは、演算項目 2 エンタリー内のストリングとの間で文字を挿入または削除しようとしても、結果フィールド・エンタリーは移動しません。デフォルトのエディター・プリファレンスでは、桁依存編集はオフです。このサポートは、ワークベンチのプリファレンス・ウィンドウに移動することによってオンに切り替えることができます。LPEX エディターには事前定義された設定値がありますが、定義可能な設定を含む関連プリファレンス・ページもあります。カテゴリーの名前は「LPEX エディター」であり、「プリファレンス」ウィンドウのツリー・ビューに表示されます。

桁を編集するには、以下のようにします。

1. ワークベンチ・メニュー・バーから「ウィンドウ」>「プリファレンス」を選択します。  
「プリファレンス」ウィンドウが表示されます。

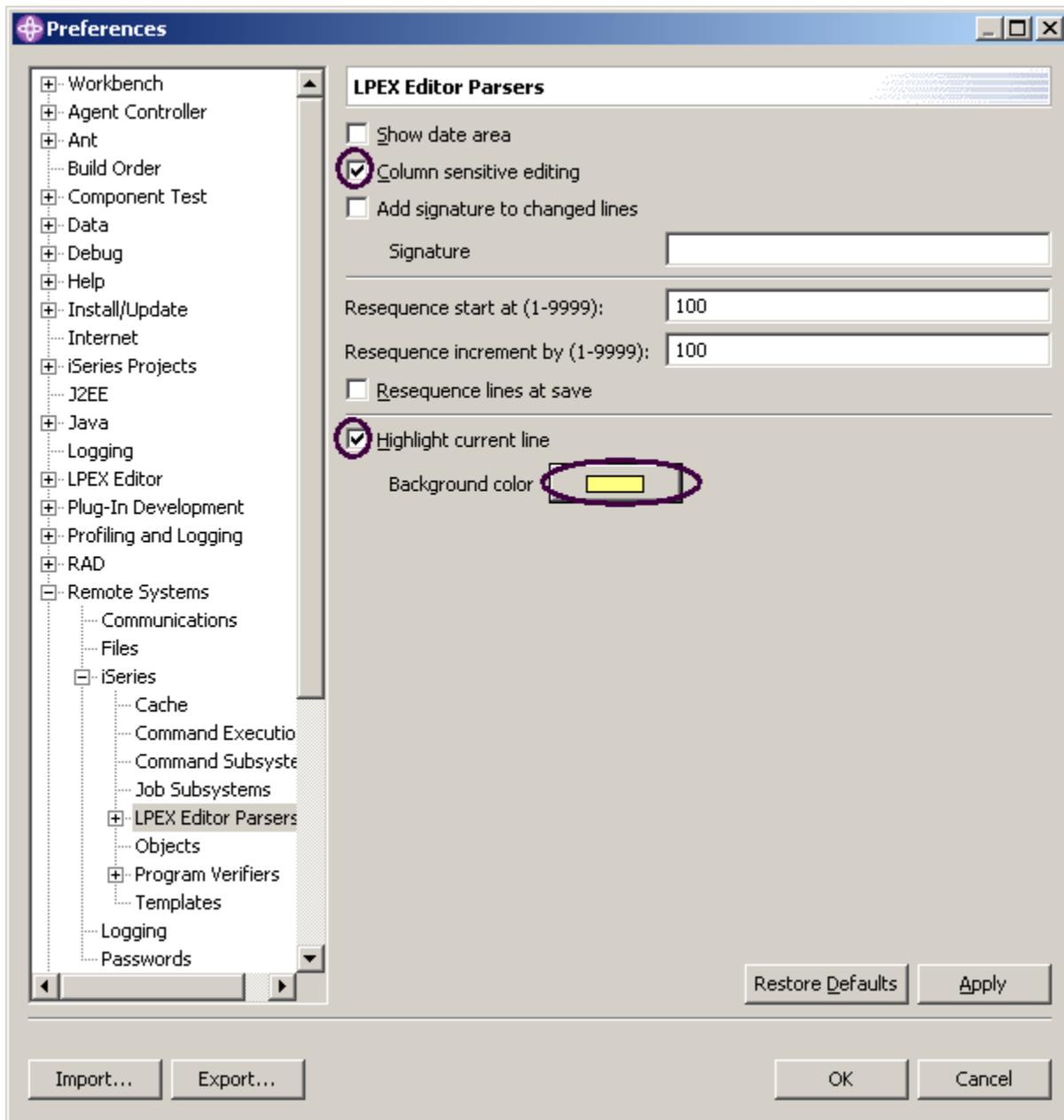


図17: 「プリファレンス」ウィンドウ

左ペインのツリー・ビューで、以下のようにします。

2. 「リモート・システム」を拡張します。
3. 「iSeries」を拡張します。

4. 「LPEX エディター構文解析プログラム」をクリックします。

右ペインでは、このフィーチャーのプリファレンスを設定できます。

5. 「桁依存編集」チェック・ボックスを選択します。

選択すると、各桁は別々のエンタリー・スペースと見なされます。

6. 「現在行の強調表示」チェック・ボックスを選択します。

このオプションは、カーソルが現在ある行を強調表示します。このオプションは、エディター領域でオープンされるすべてのソース・ファイルに適用されます。

7. 「背景色」をクリックします。「色」ウィンドウがオープンします。

8. カラー・パレットから**淡黄色**を選択します。

9. 「OK」をクリックします。「プリファレンス」ウィンドウが表示されます。

**ヒント:** 他にもプリファレンス設定が「LPEX エディター構文解析プログラム」の下にあります。このエンタリーを拡張すると、個々の言語環境のプリファレンス設定が表示されます。

10. 「プリファレンス」ウィンドウでもう一度「OK」をクリックします。

ここで、この結果を確認しましょう。

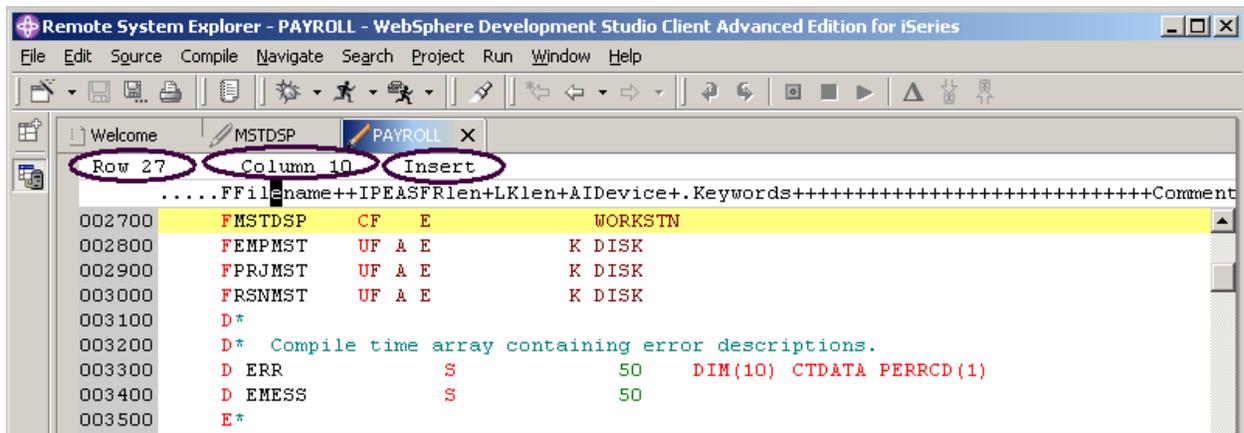


図 18: 挿入モードのエディター(行27 が強調表示されている)

エディター・ウィンドウで、以下のようにします。

11. 27 行目の 10 桁目にカーソルを移動します。

12. エディターが**挿入**モードであることを確認します。

状況域に「置換」と表示される場合、以下のようにします。

13. **Insert** キーを押します。

14. スペース・バーを 3 回押します。

ファイル名のみがシフトされ、右側の他の桁はいずれも影響を受けないことに注意してください。

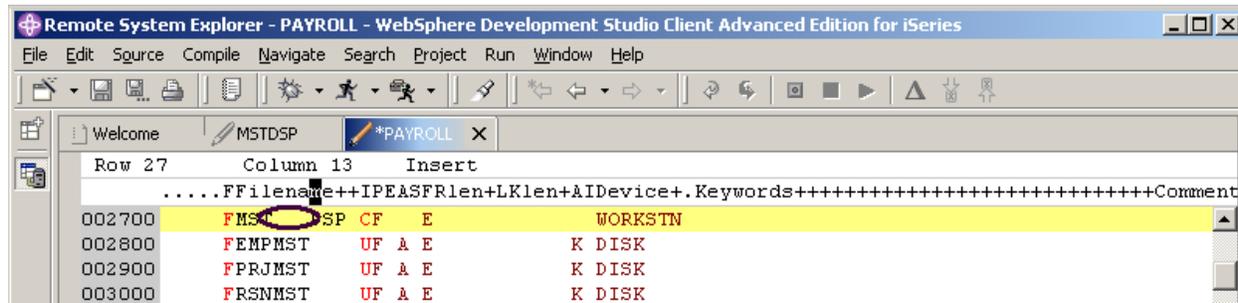


図19: ファイル名に空白が挿入され、他の桁はいずれもシフトされない

15. **Backspace** を 3 回押します。

ファイル名はもう一度影響を受けますが、他の桁は影響を受けません。

## SEU コマンドの入力

一般的な多くのエディターのキーボードおよびコマンド個別設定を採用するように LPEX エディターを構成することができます。多くのエディター・プロファイルは、さまざまなエディター・タスクの実行に使用されるキーおよびコマンドでのみ異なります。以下にリストされた一部の基本エディター・プロファイルでは、各行の先頭に接頭部情報およびコマンド域が追加されます。

- ispf
- seu
- xedit

エディターは、これらのエディター・プロファイルによって使用される接頭部コマンドを認識します。ユーザーは使用するプロファイルに応じて、接頭部域がアクティブのときに **SEU**、**XEDIT**、または **ISPF** コマンドを入力できます。

**SEU** エキスパートは、**SEU** コマンドを使用できることの良さを理解できます。

1. 編集域の左にある灰色のシーケンス番号域にカーソルを移動します。
2. 任意のシーケンス番号上で「dd」と入力します。
3. 数行下に移動し、再び「dd」と入力して **Enter** を押します。

行が削除されたことに注意してください。

4. ここで、シーケンス番号域に「i5」と入力します。
5. カーソルがシーケンス番号域内にあることを確認します。
6. **Enter** を押します。

新しい行が 5 つ挿入されます。

## 元に戻す操作および再実行操作の要求

エディターは、エディター・ウィンドウでファイルに対して行われた変更の各セットを記録します。最後にファイルが保管されたとき以降に行われた変更の数が、状況表示行に表示されます。ファイルに対して行われた変更のセットを元に戻したい場合は、元に戻す操作を使用します。再実行操作を使用すれば、元に戻す操作の効果をキャンセルすることができます。

ここでは、ファイルに対して行った変更のいくつかを元に戻します。次に、再実行操作を使用して変更をキャンセルします。最後に、元の形に戻るようソースを再ロードします。

編集の変更を元に戻す、および再実行するには、以下のようにします。

1. ワークベンチ・メニュー・バーから「**編集**」>「**元に戻す**」を選択します。  
新しい 5 つの行が消えます。
2. **Ctrl+Z** を押して、別の元に戻す操作を行います。  
削除された行が再表示されます。
3. ワークベンチ・メニュー・バーから「**編集**」>「**再実行**」を選択します。  
行が再び削除されます。  
ここで、**iSeries** からソースを再ロードして、元の形に戻っていることを確認します。それには以下のようにします。
4. ワークベンチ・メニュー・バーから「**ファイル**」>「**クローズ**」オプションを選択します。  
「リソースの保管」ウィンドウが表示され、新しい変更を保管するかどうか尋ねられます。

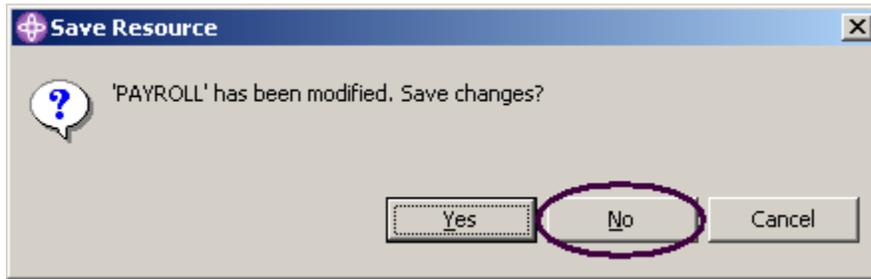


図 20: 確認ウィンドウ

5. 「いいえ」をクリックします。
6. ワークベンチのリモート・システム・エクスプローラー・パースペクティブに戻り、QRPGLESRC ファイル内の **PAYROLL** メンバーを再ロードします。

## 言語依存ヘルプの起動

エディターでは、カーソル依存の言語参照ヘルプが使用できます。RPG 仕様のフィールドの順序または可変フィールドに入力できる値を思い出すことができない場合、下のヘルプが非常に役立ちます。このヘルプは、LPEX エディター・ウィンドウから使用できます。

言語依存のヘルプを表示するには、編集ウィンドウで **F1** を押します。カーソルが命令コード上にある場合は、命令コードのヘルプが表示されます。それ以外の場合は、現行仕様のヘルプが表示されます。

言語依存のヘルプにアクセスするには、以下のようにします。

1. ILE RPG ソースの **112** 行目のワード **MOVE** 上にカーソルを置きます。
  2. **F1** を押します。
- MOVE** 命令コードの言語依存のヘルプが「ヘルプ」ウィンドウに表示されます。

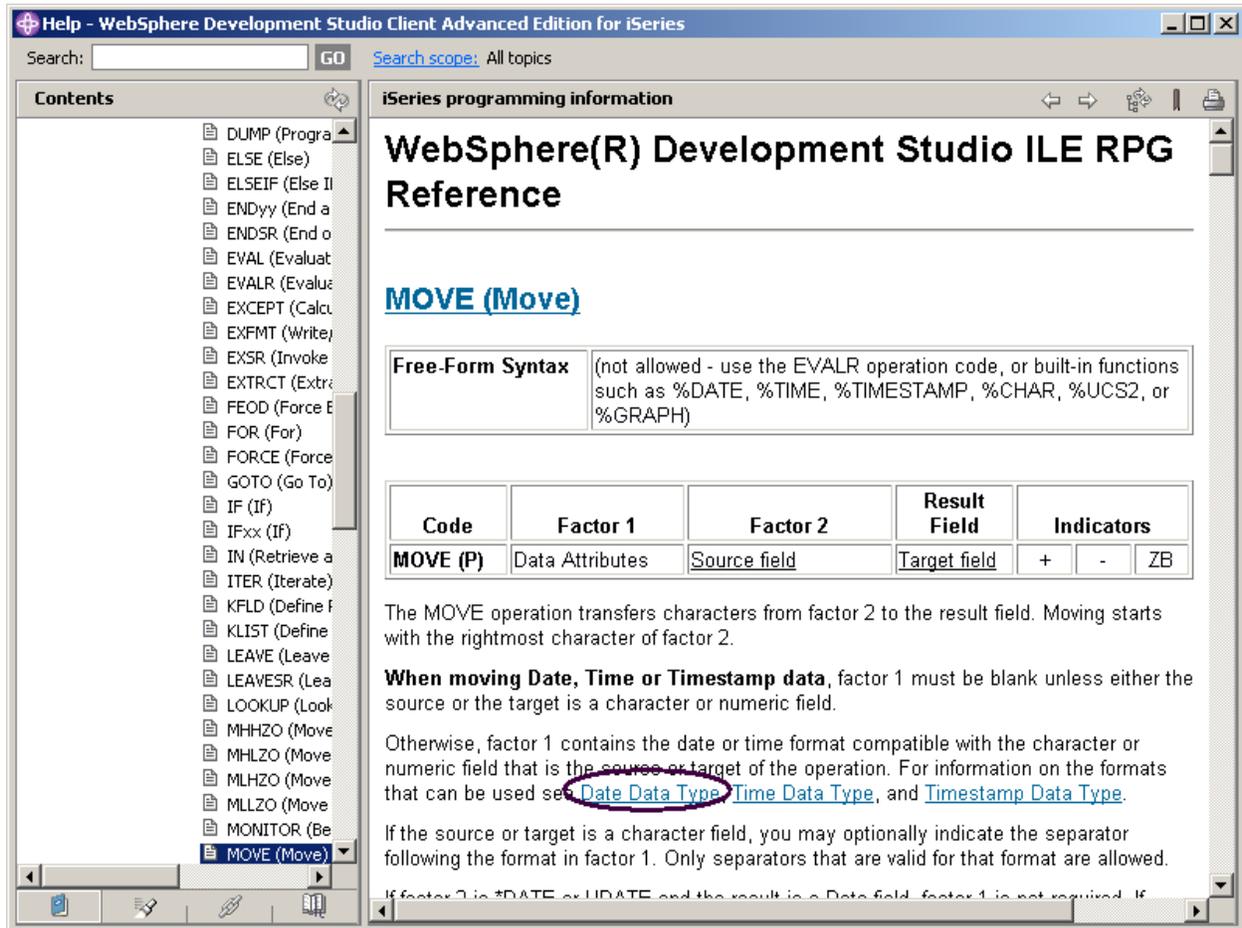
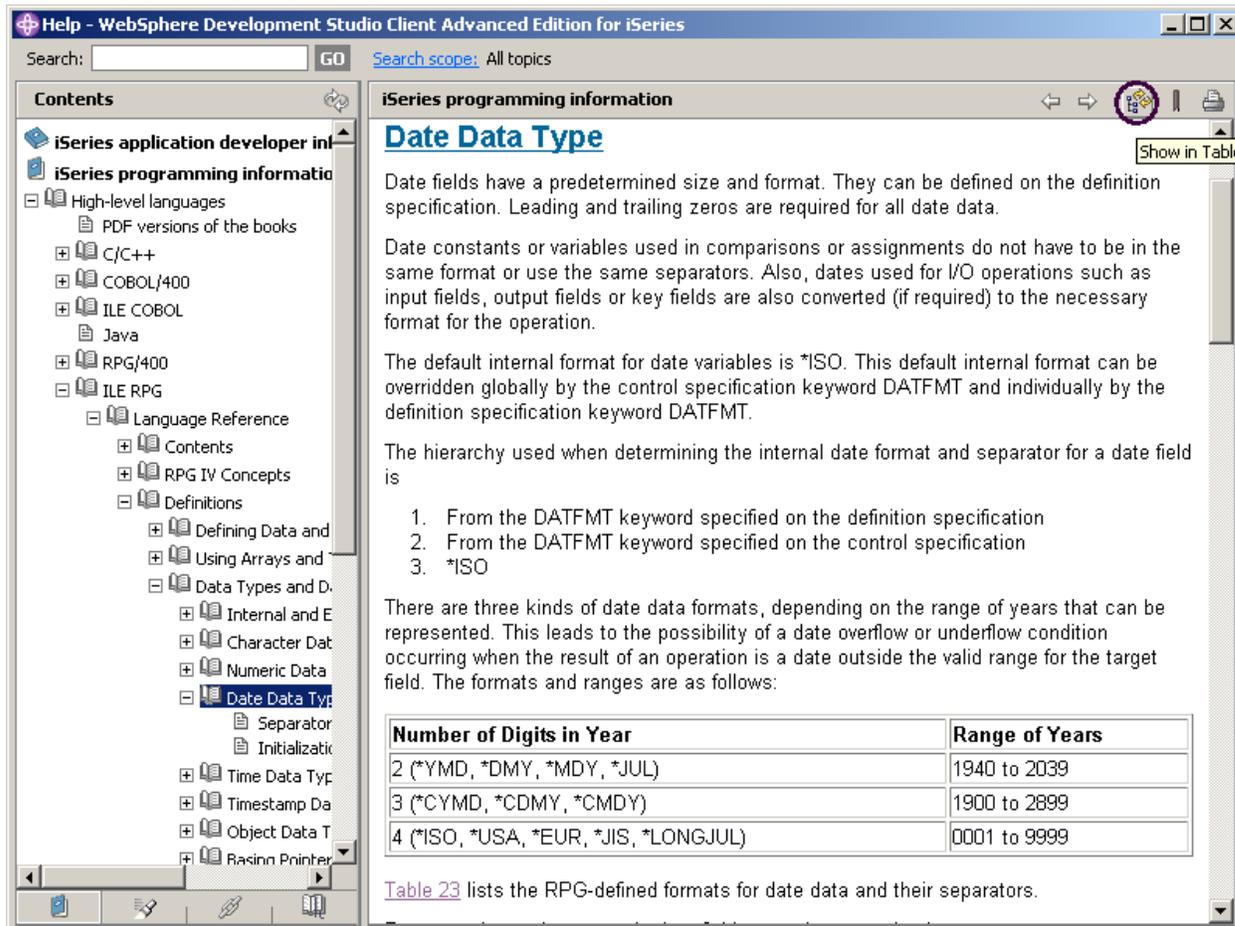


図 21: MOVE のヘルプおよび同期化ボタン

「ヘルプ」ウィンドウに青色でマークを付けられたテキストには、青色のトピックに関する詳細情報へのリンクが含まれています。

3. リンク「日付データ・タイプ」をクリックします。



4. 「ヘルプ」ウィンドウのツールバーにある「目次への表示」をクリックします。  
これにより、左にある「目次」ペイン内のトピックが、メイン・ヘルプ・ペインに表示するヘルプ・トピックに同期化されます。
5. 「ヘルプ」ウィンドウでさまざまな操作を試し、他に使用可能な操作を確認します。
6. 「ヘルプ」ウィンドウを最小化します。
7. ワークベンチ・メニュー・バーで「ヘルプ」>「ヘルプ目次」を選択し、製品で使用できるすべてのヘルプのリストを参照します。



図 22: メイン「ヘルプ」ウィンドウ

ここに目次が表示されるので、関心のあるトピックを選択できます。

8. 「目次」ペインのトピック・リストから「**iSeries プログラミング情報**」を選択します。

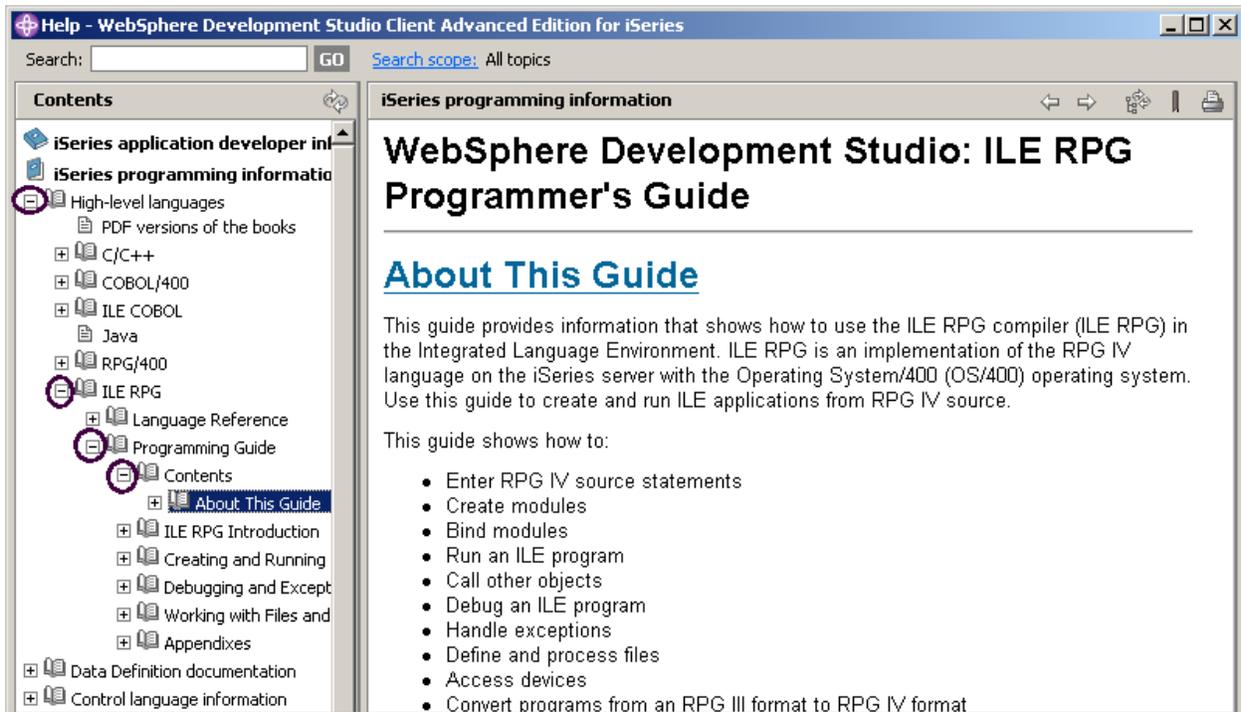


図 23: RPG IV マニュアルの検索

9. 「**高級言語**」を拡張します。
10. 「**ILE RPG**」を拡張します。

11. 「プログラミング・ガイド」を拡張します。

12. 「目次」を拡張します。

13. 「本書について」をクリックします。

マニュアルの最新版が手元があれば、プログラミング情報を容易に検索できます。また、検索ストリングを指定することによりヘルプを検索するためのオプションも用意されています。デフォルトでヘルプ全体が検索されます。検索を特定の文書に制限するには、「検索有効範囲」オプションを選択します。

14. 「検索有効範囲」をクリックします。

15. 「検索有効範囲の選択」ウィンドウで「以下のトピックのみを検索する」ラジオ・ボタンを選択します。

16. 「新規」をクリックします。

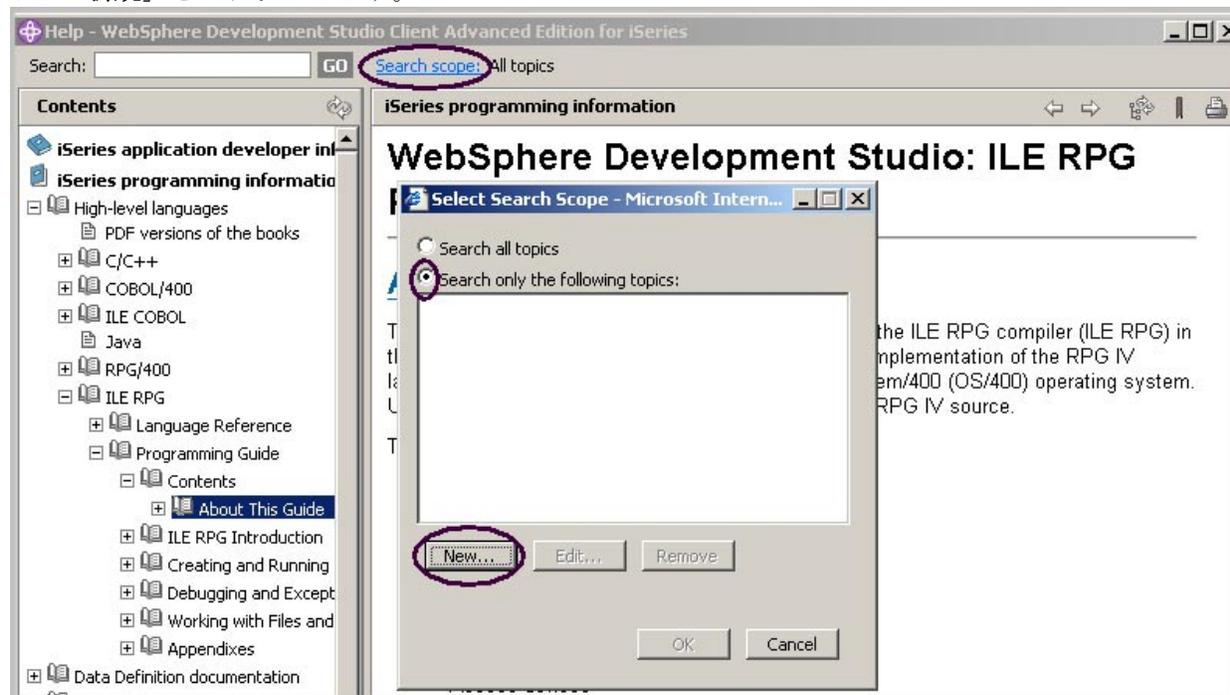


図 24: 検索有効範囲の設定

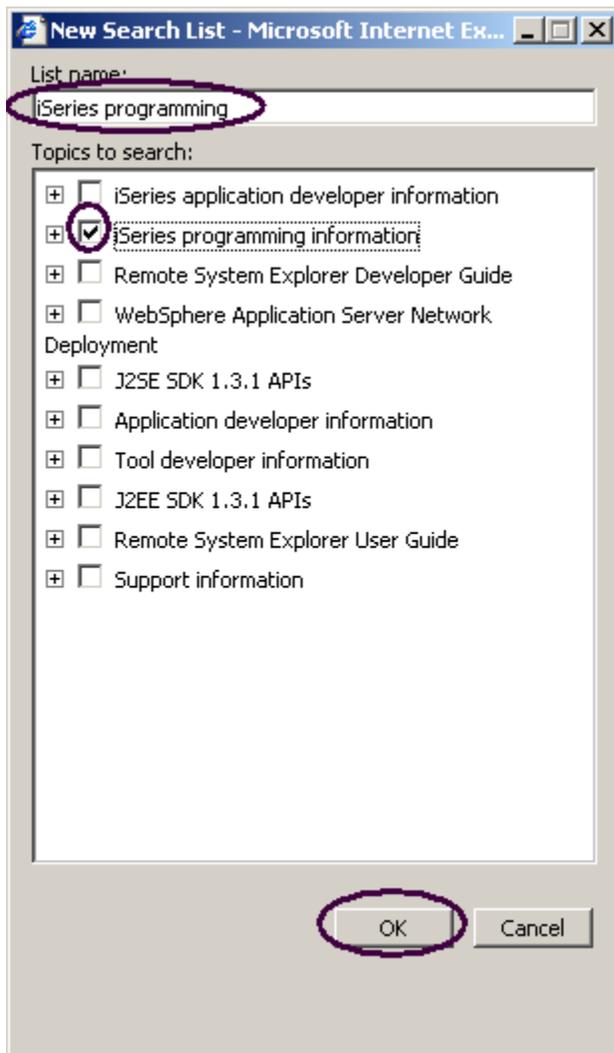


図25: 検索リスト・セットアップ

「新規検索リスト」ウィンドウで、以下のようになります。

17. たとえば、リスト名「iSeries プログラミング情報」を入力します。
18. 検索したい文書の横にあるチェック・ボックスを選択します。
19. 「**OK**」をクリックします。
20. 「検索有効範囲の選択」ウィンドウで「**OK**」をクリックします。

## 言語仕様のプロンプトを出す

エディター・ウィンドウでコードを直接入力または変更する代わりに、プロンプトを使用することができます。行の仕様のプロンプトを要求すると、ウィンドウが表示され、入力フィールドを使用して、該当する行を入力または変更できます。

プロンプトにアクセスするには、以下のようにします。

1. ワークベンチに戻ります。エディター・ウィンドウの **33** 行目にある「D-spec」にカーソルを移動します。
2. ワークベンチ・メニュー・バーから「ソース」を選択します。
3. 「プロンプト」を選択します (または **Shift+F4** を押します)。

エディター・ウィンドウのサイズが縮小され、ワークベンチの下部に「iSeries ソース・プロンプター」ウィンドウを参照できるようになります。「iSeries ソース・プロンプター」ウィンドウには、行の仕様が個々のフィールドに分割されて表示されます。

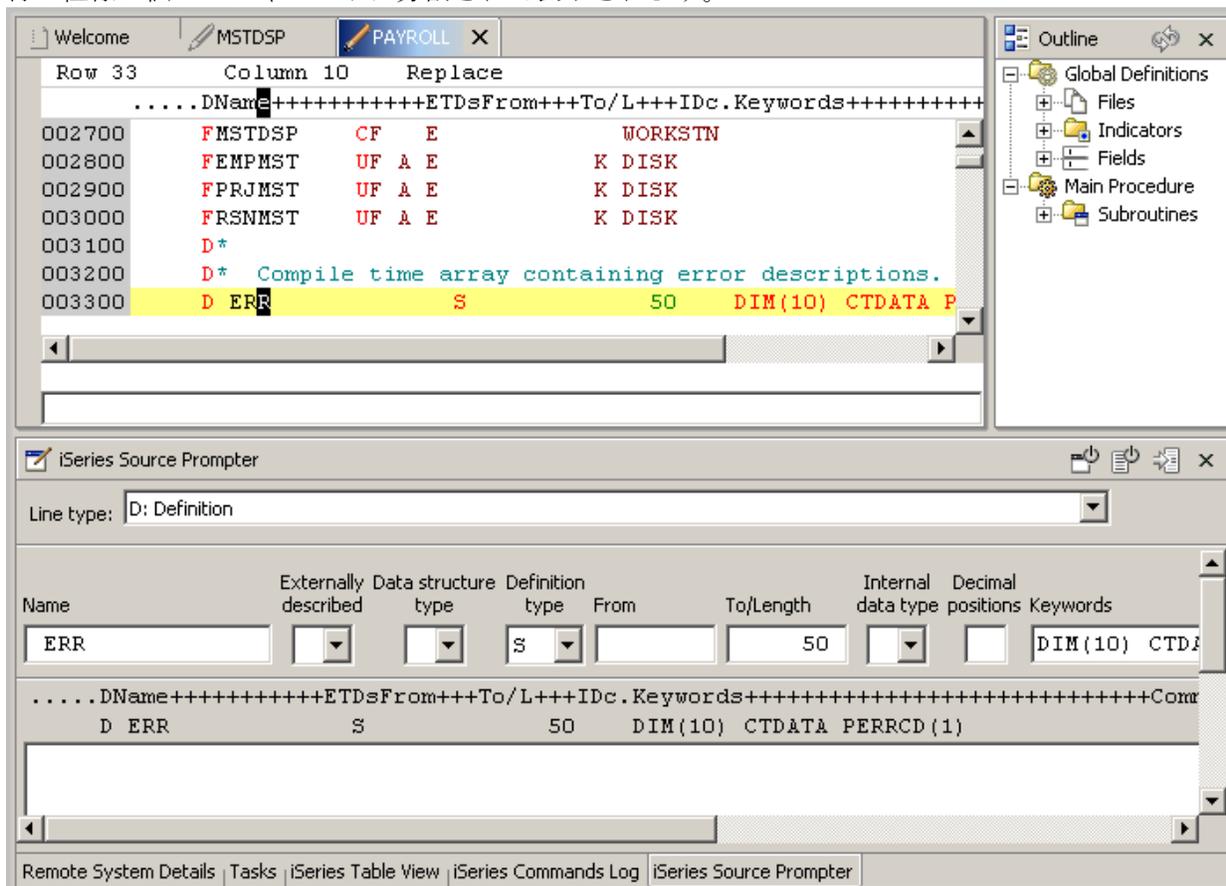


図 26: プロンプト・ウィンドウ

「iSeries ソース・プロンプター」ウィンドウ内の任意のフィールドについてコンテキスト依存ヘルプを表示するには、以下のようにします。

1. **Tab** キーを使用して、「**キーワード**」フィールドにカーソルを移動します。
2. **F1** を押して、このフィールドのヘルプを参照します。

「ヘルプ」ウィンドウに **D** 仕様キーワードのヘルプが表示されます。ウィンドウが自動的に表示されない場合、**Windows** タスクバー上のアイコンをクリックして、ウィンドウを一番上に表示しなければならないことがあります。

通常テキストとは異なる色のワードがヘルプ内に表示されます。それはヘルプ・リンクであり、そのワードまたは句に関して使用可能な追加のヘルプがあることを示します。

3. 任意のリンクをクリックして、その項目の特有のヘルプを参照します。
4. 「ヘルプ」ウィンドウを最小化します。

エディター・ウィンドウのサイズを縮小せずに、最大化されたエディター・ウィンドウから

「iSeries ソース・プロンプター」ウィンドウにアクセスするには、「iSeries ソース・プロンプター」ウィンドウを高速ビューとして左タスクバーに配置します。

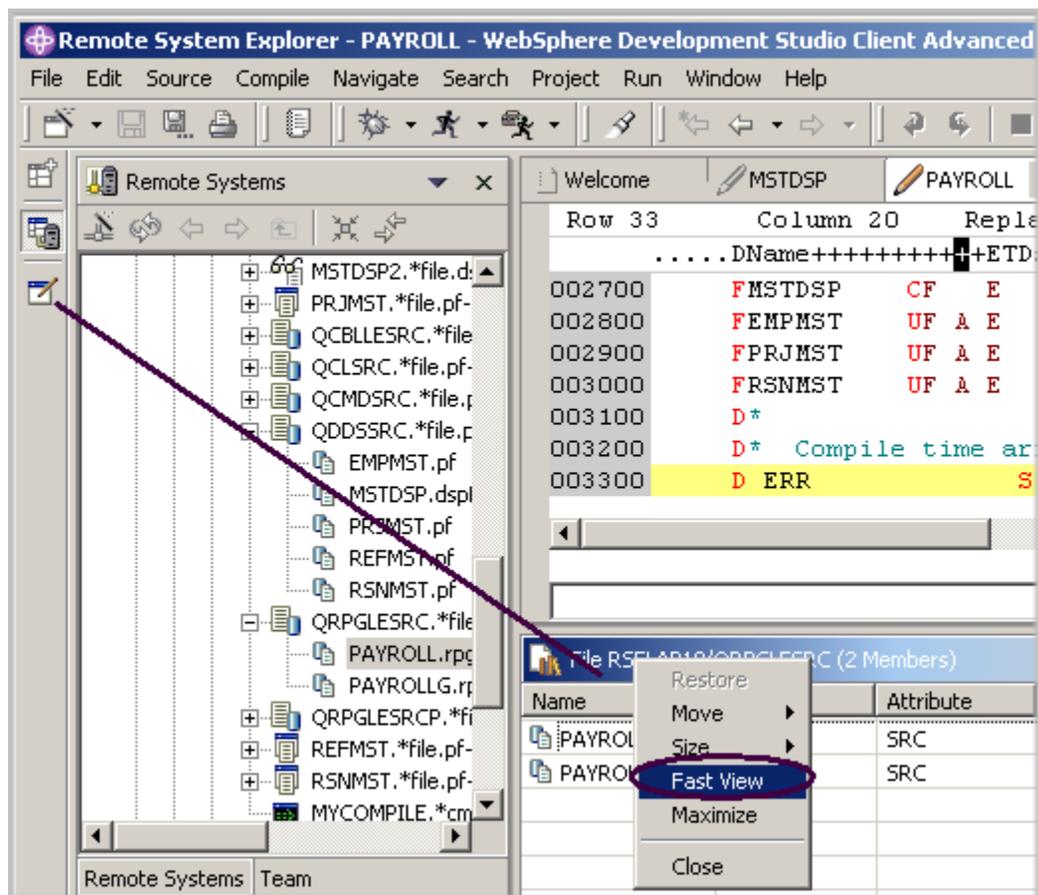


図 27: 「iSeries ソース・プロンプター」ウィンドウをタスクバー上の高速ビューに移動

5. 「iSeries ソース・プロンプター」ウィンドウ見出しを右クリックします。
6. ポップアップ・メニューから「高速ビュー」を選択します。

ワークベンチの左側のタスクバーに iSeries ソース・プロンプター・アイコンが表示されます。

7. エディター・ウィンドウの見出しをダブルクリックして、エディター・ウィンドウを最大化します。
8. エディター・ウィンドウで行を選択し、**Shift+F4** を押して行のプロンプトを出します。

「iSeries ソース・プロンプター」ウィンドウがエディター・ウィンドウの上に表示されます。

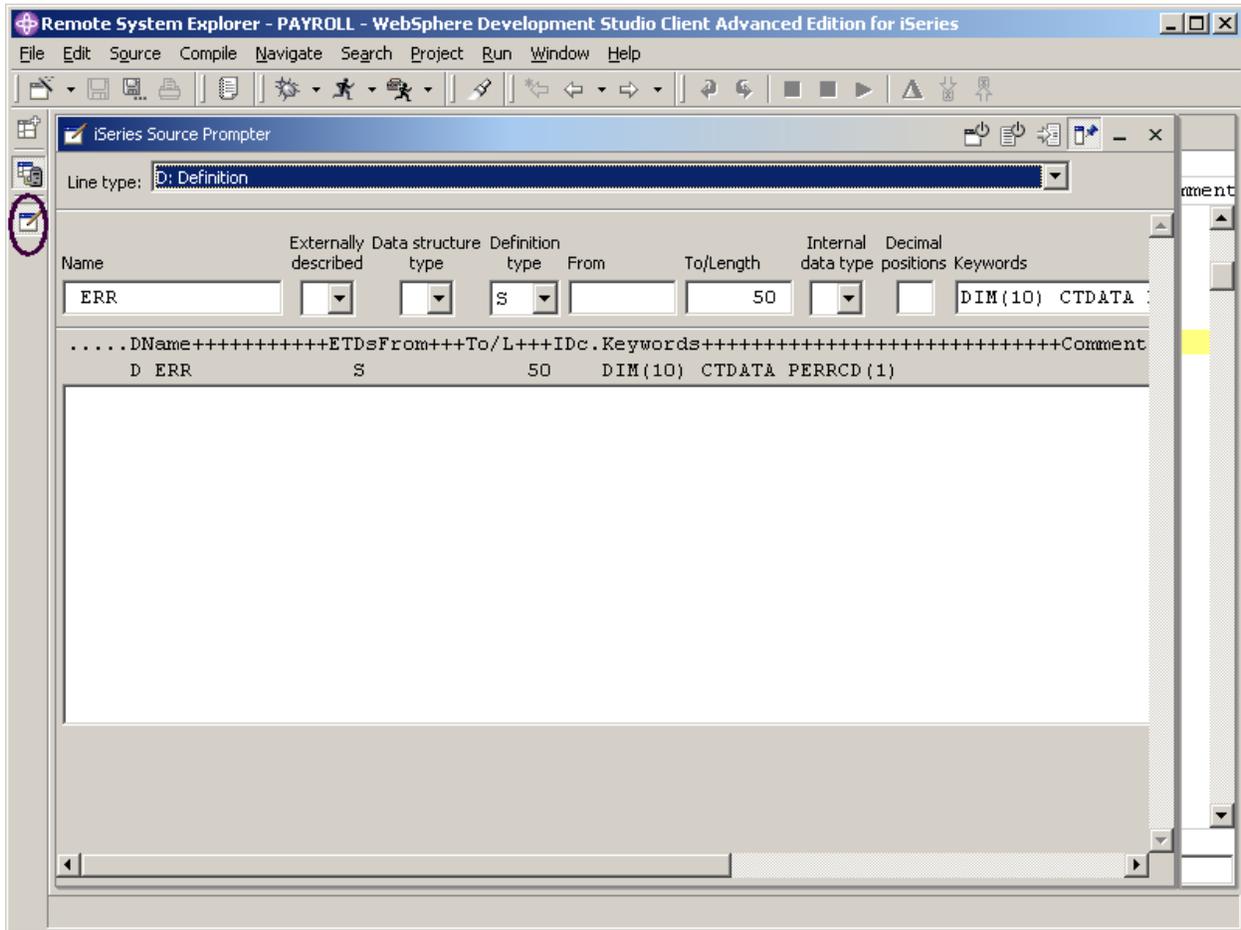


図 28: エディター・ウィンドウの上に表示されたプロンプト・ウィンドウ

9. タスクバー上の **iSeries** ソース・プロンプター・アイコンをクリックして、プロンプト・ウィンドウを最小化します。

注: 「iSeries ソース・プロンプター」ウィンドウのツールバーにある 3 つの押しボタンで、ソース・プロンプト・ビューを使用不可にする、構文検査を使用不可にする、および挿入モードに変更することができます。

10. エディター・ウィンドウの見出しをダブルクリックして、エディター・ウィンドウを元のサイズに戻します。

## ソースの字下げ

ILE RPG ソースの編集時に、構文の開始と終了を判別することが困難な場合があります。字下げオプションを使用すれば、構文が使用されているソースを字下げモードで表示できます。デフォルトでは、字下げオプションを使用すると、画面が横方向に分割され、字下げされたビューが下のペインに表示されます。

ソースを字下げするには、以下のようにします。

1. ワークベンチ・メニュー・バーから「ソース」>「字下げの表示」を選択します。

字下げされたビューがワークベンチの下部に表示されます。

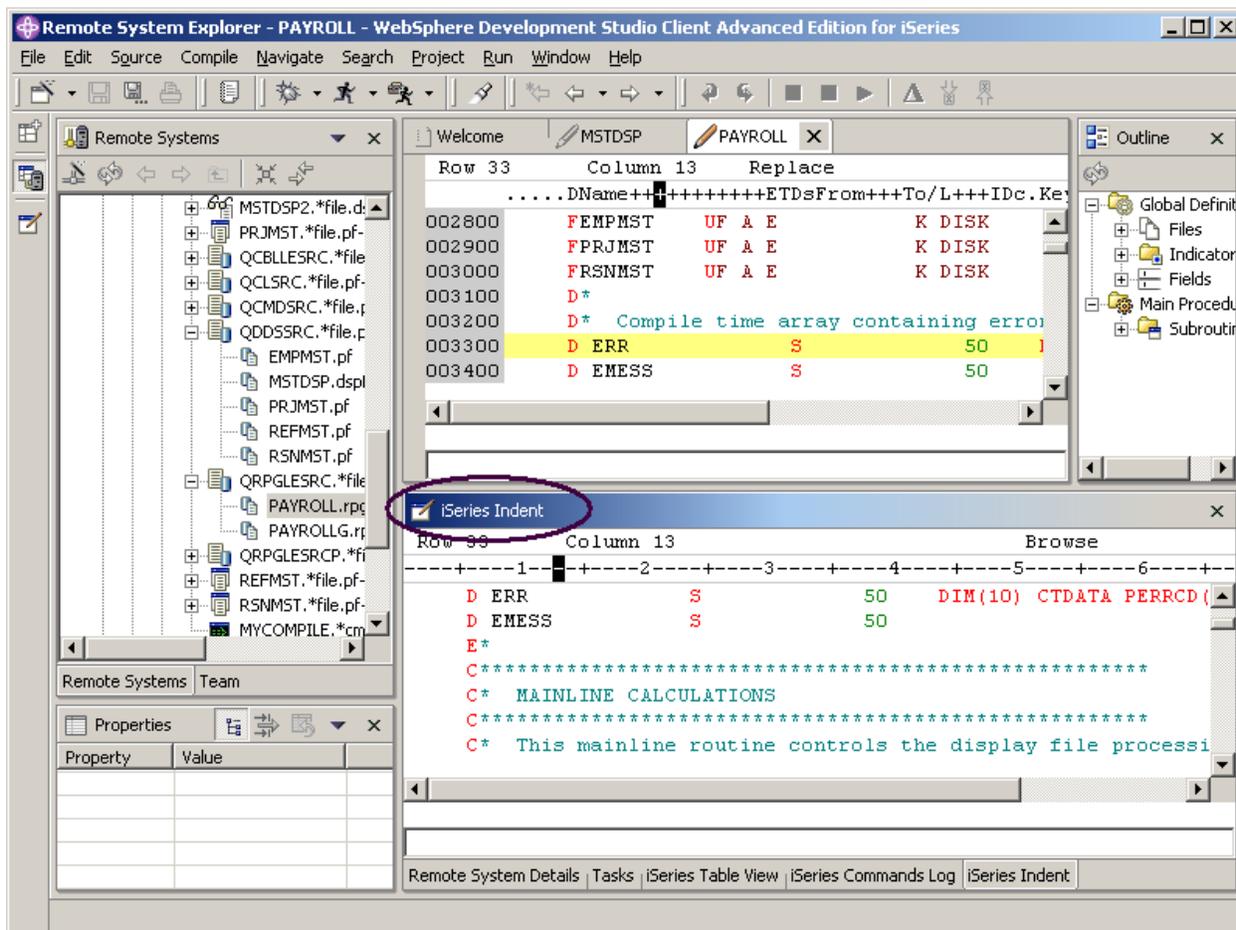


図 29: ワークベンチの下部に表示された字下げビュー

字下げされたビューは完全なビューとして表示できます。

2. 字下げされたビューの「iSeries 字下げ」ウィンドウのタイトル・バーをダブルクリックします。

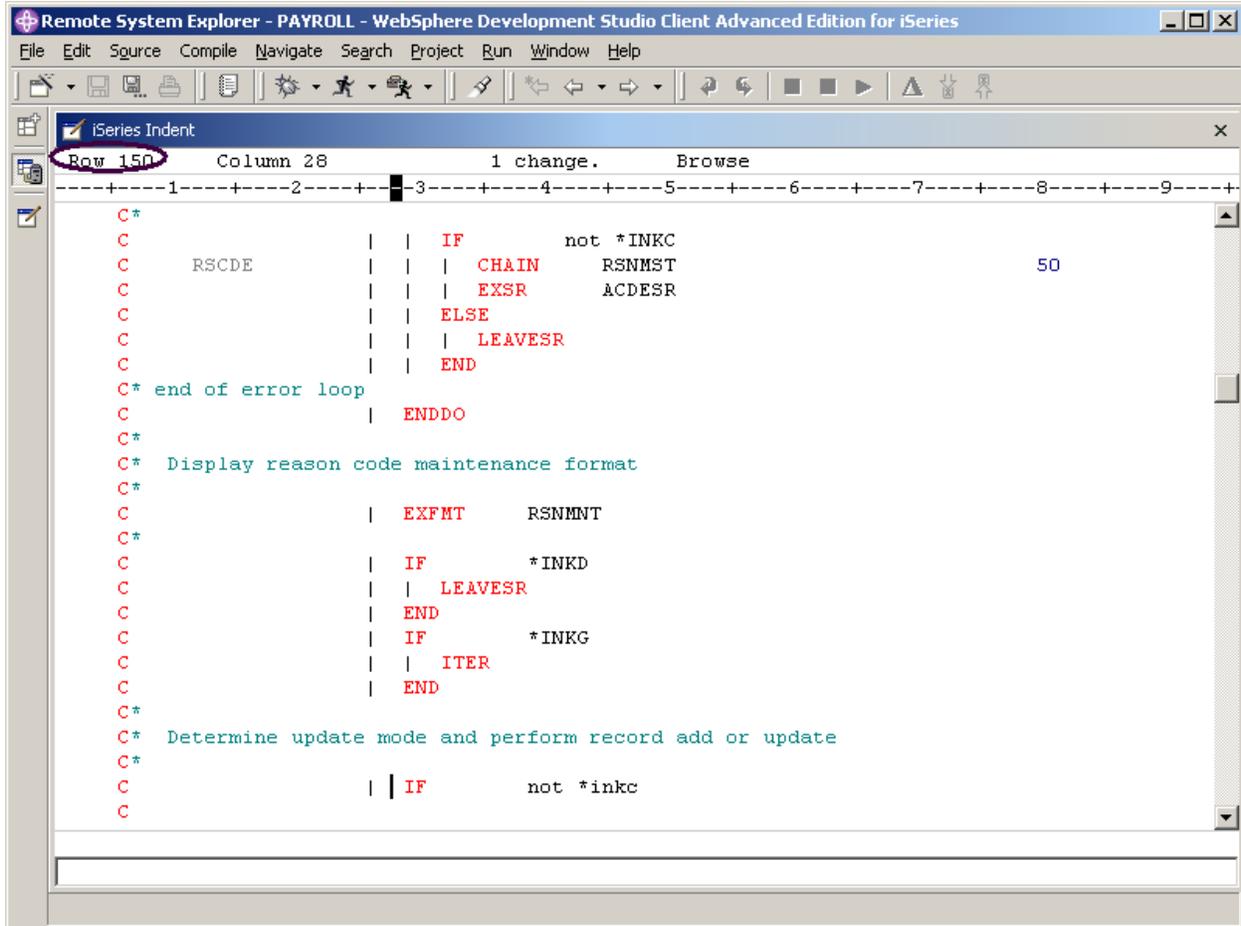


図 30: 最大化された字下げビュー

3. 150 行目にスクロールダウンします。

この領域では、いくつかの条件がネストされ、行が字下げされています。これは、条件の開始と終了を認識するのに役立ちます。

ヒント: このビューは、高速ビューとして左のタスクバーに移動することもできます。

注: 字下げされたビューは参照モード専用であり、編集できません。

4. ビューの右上隅にある「X」をクリックして、字下げされたビューをクローズします。これにより、**PAYROLL** プログラムが表示されているエディター・ウィンドウに戻ります。

## テキストの検索および置換

LPEX エディターには、テキストの検索および置換の強力なフィーチャーもあります。項目の検索には、「検索および置換」ウィンドウを使用します。ワード、ワードの一部、またはこれらの一連を検索できます。パターンが正規表現の規則に従う場合は、一致させたいパターンを入力することもできます。検出された検索項目は置換できます。入力したテキストまたはパターンが見つかる、選択された検索方向に従って、カーソルが検索項目の前後のオカレンスに移動し、その選択内容に従って検出テキストを置換します。

Payroll のエディター・ウィンドウでテキストの検索および置換を行うには、以下のようにします。

1. **Ctrl+Home** を押して、ファイルの先頭に移動します。

**ヒント:** **Ctrl+Home** を押して先頭に移動するか、または **Ctrl+End** を押して末尾に移動するとき、カーソル位置にクイック・マークが設定されます。これにより、**Alt+Q** を押すことで移動前の行に戻ることができます。**Ctrl+Q** を押しても、クイック・マークが設定されます。

2. ワークベンチ・メニュー・バーから「**編集**」>「**検索および置換**」を選択するか、**Ctrl+F** を押します。

検索および置換のウィンドウが、エディター・ウィンドウの下部に表示されます。

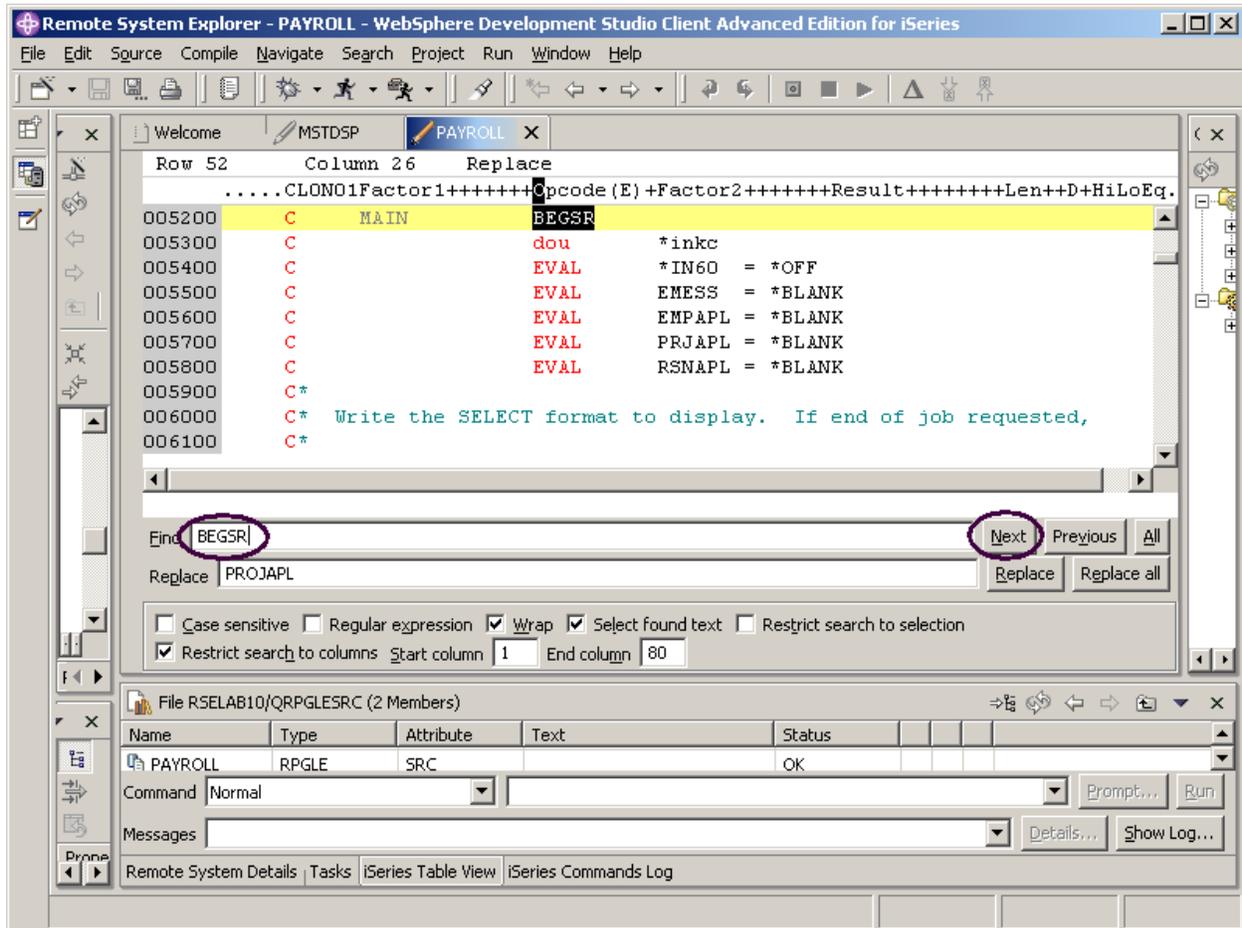


図 31: 検索および置換のウィンドウ

このウィンドウの下部には、特定の桁内でのみ検索するなどの選択オプションがいくつかあります。「BEGSR」の最初のオカレンスを見つけることにします。

3. 「検索」フィールドに「BEGSR」と入力し、サブルーチンの開始を検出します。「置換」フィールドが空白であることを確認します。このフィールドは、テキストを置換する場合に使用します。

エディターでアクティブな行が 52 行目に移動します。この行には、ファイル内の最初の「BEGSR」句が含まれています。

4. 「次へ」をクリックして、ファイル内の次の位置にある「BEGSR」に移動します。

## ストリングによる行のフィルター操作

エディターを使用すれば、ソースをフィルター操作またはサブセット化し、一定のストリングを含む行のみを参照することができます。行にフィルターを掛けることにより、ソースをスクロールせずに素早く容易に行を検索できます。



図 32: 選択された命令コード「BEGSR」

ストリングによりソースをフィルターに掛けるには、以下のようにします。

1. 編集ウィンドウで「**BEGSR**」命令コードをダブルクリックして選択します。
2. ワークベンチ・メニュー・バーから「編集」>「選択内容」>「選択内容をフィルターに掛ける」を選択します。

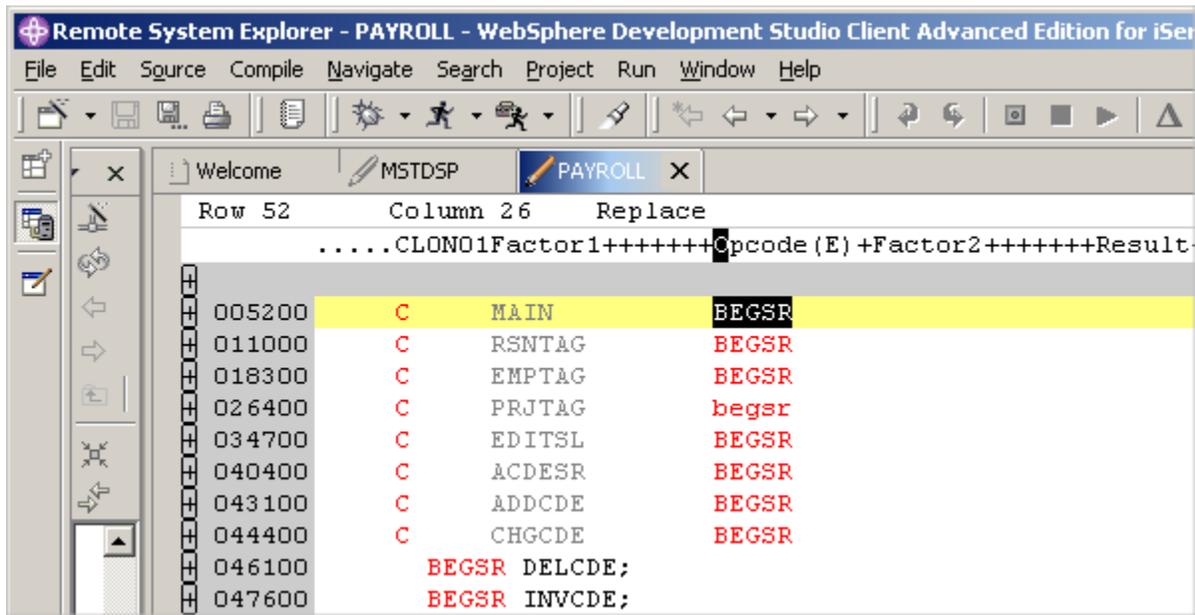


図 33: 「BEGSR」句がある行をすべて表示するエディター・ウィンドウ

3. 数行下の **347** 行目にカーソルを移動します。
4. 347 行目の横にあるプラス符号 (+) をクリックして、このセクションを拡張します。

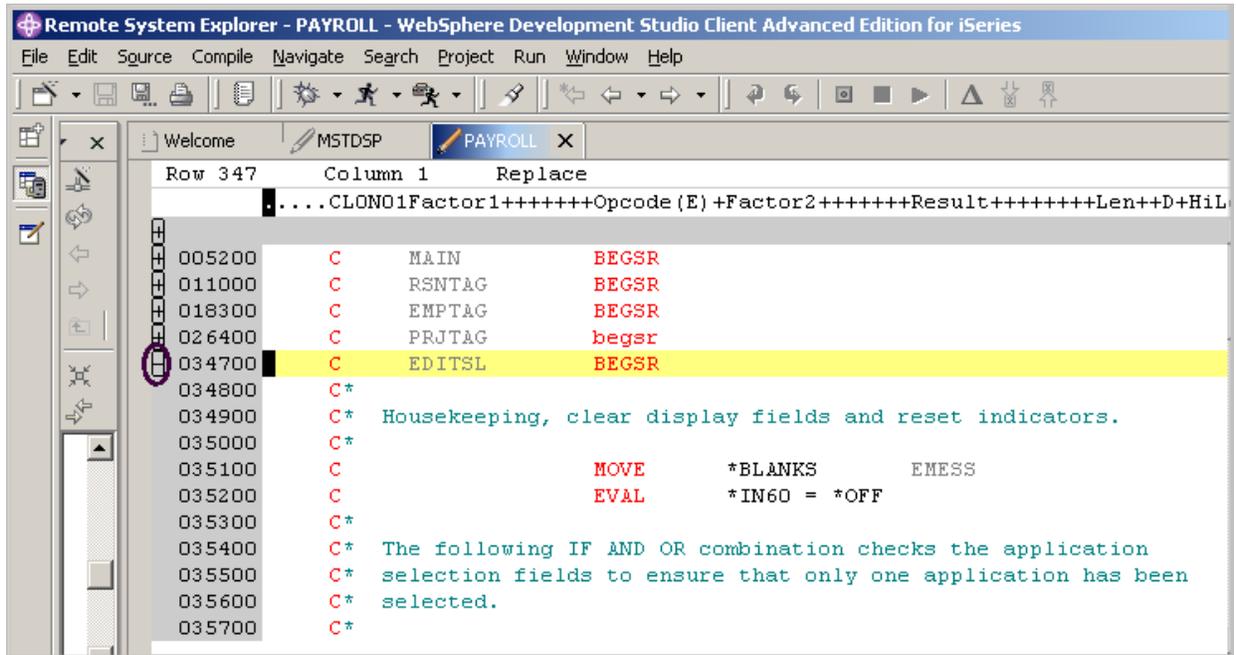


図 34: 拡張されたセクション

ここで、再びソース全体を表示します。

1. ワークベンチ・メニューから「編集」>「すべて表示」を選択するか、**Ctrl+W** を押します。  
すべての行が現在表示されていますが、カーソルは移動したときと同じ行にまだあります。

### タイプを基にした行のフィルター操作

エディターでは、ILE RPG ソースを素早くナビゲートできるように、行タイプを基にして行をフィルターに掛けることができます。すべてのサブルーチンが定義されているソース内の場所を参照したいとします。

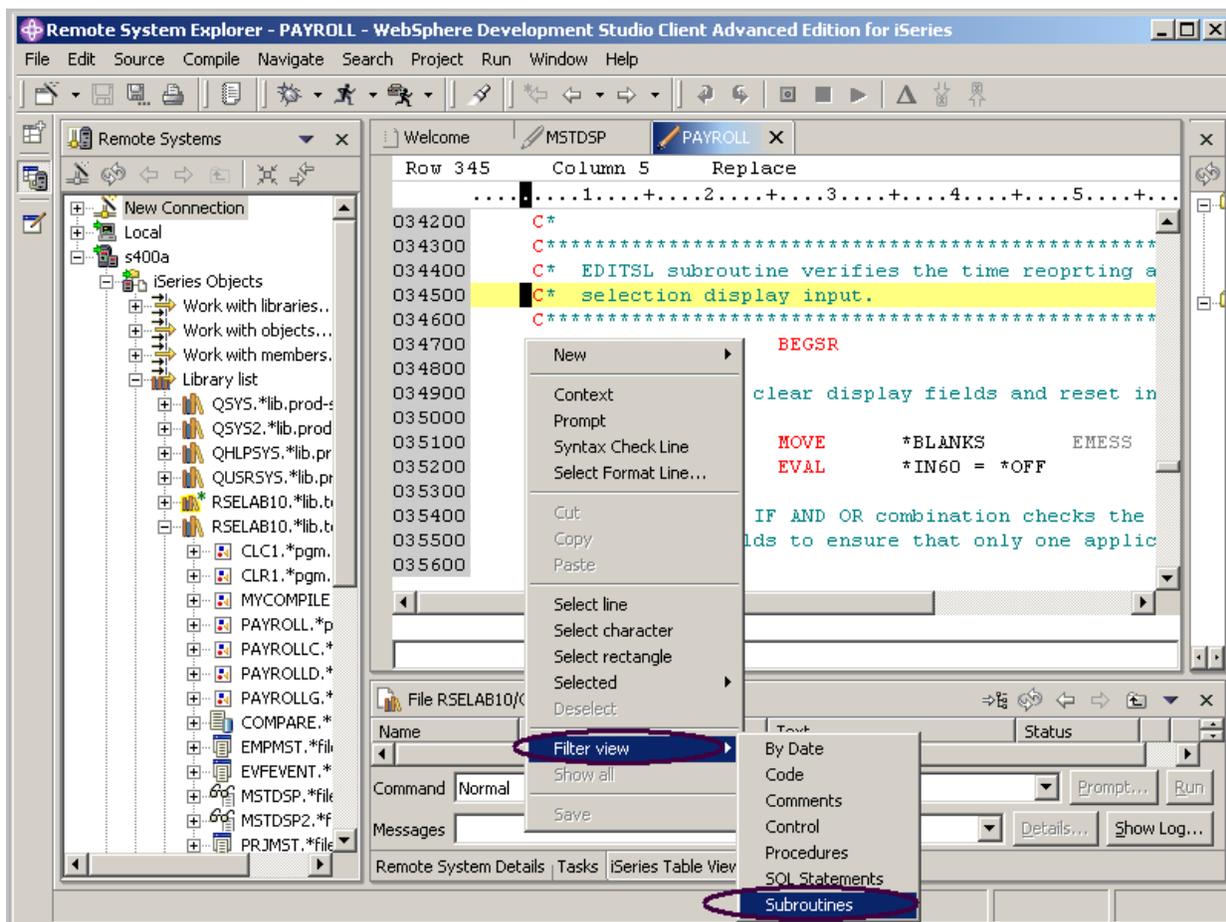


図 35: エディター・ポップアップ・メニューとフィルター・サブメニュー

1. エディター・ウィンドウ内で右クリックします。
2. 「ビューをフィルターに掛ける」>「サブルーチン」を選択します。

すべてのサブルーチン仕様が表示され、参照したいサブルーチンがあるファイル内の領域に素早く容易に移動できるようになります。

Row 347	Column 1	Replace
	.....CLONO1Factor1++++++Opcode (E) +Factor2+++++	
017600	C	ENDsr
018300	C	EMPTAG
025700	C	ENDSR
026400	C	PRJTAG
033500	C	endsr
034700	C	EDITSL
040400	C	ACDESR
042700	C	ENDSR
043100	C	ADDCDE
044000	C	ENDSR
044400	C	CHGCDE
045400	C	ENDSR
046100		BEGSR DELCDE;
047200		ENDSR;
047600		BEGSR INVCDE;
047900		ENDSR;

図36: サブルーチン・フィルターの適用

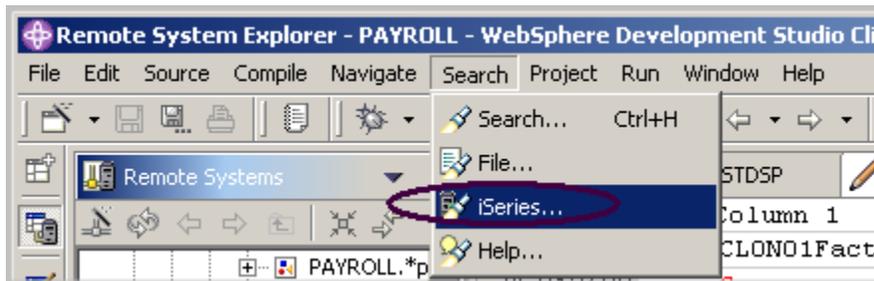
3. サブルーチン宣言 **CHGCDE** がある行にカーソルを移動します。(444 行目)
4. この宣言の横のプラス符号 (+) をクリックして、このサブルーチンのすべての行を表示します。  
これで、このサブルーチン内のソースを操作できるようになりました。

## 複数ファイルの検索

ソース物理ファイル内のメンバーを介して、またはローカル・ディレクトリー内のファイルを介して検索を行う場合、検索ツールを使用できます。マルチファイル検索ユーティリティーにより、ホスト上の多くのメンバーで特定のテキスト・ストリングを検索できます。この機能はローカル・ファイル上でも使用できます。

複数のファイルを検索するには、以下のようにします。

1. ワークベンチ・メニュー・バーから「検索」を選択します。



「検索」ウィンドウが表示されます。

2. 「検索ストリング」フィールドに「ENHRS」と入力します。

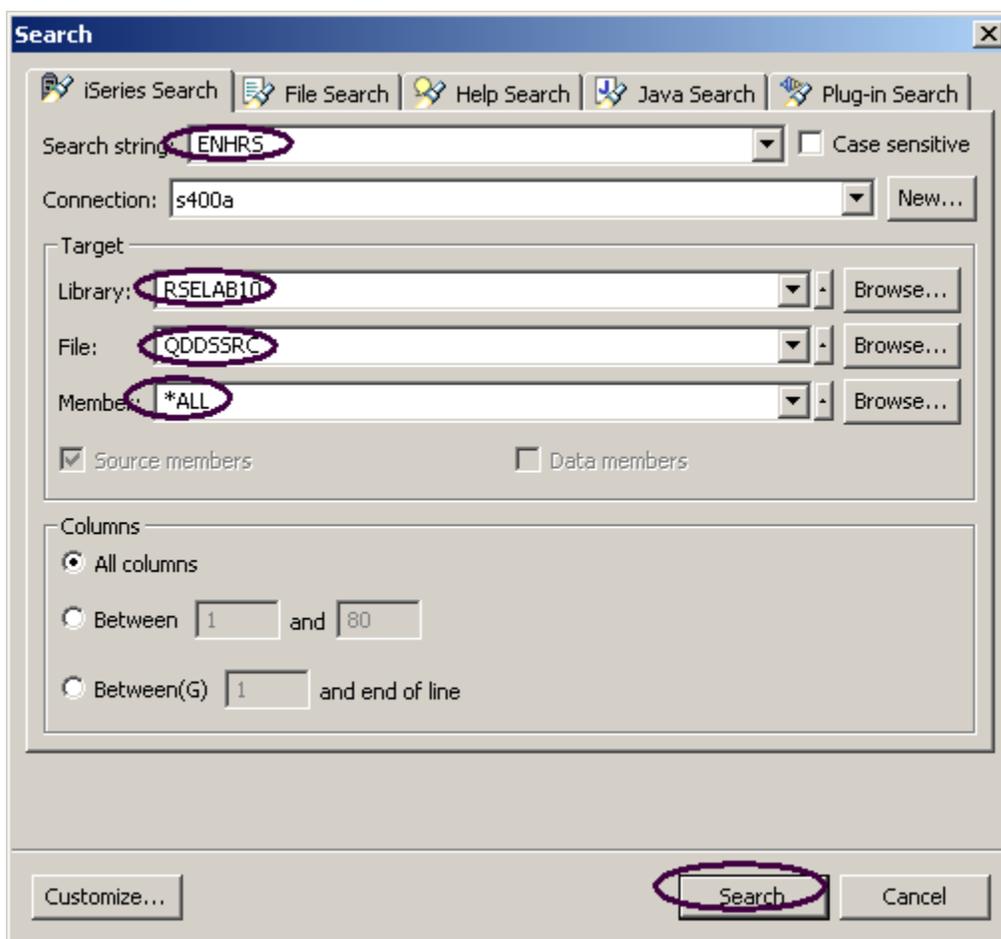


図 37: ファイル検索ウィンドウ

「接続」入力フィールドには、ご使用の iSeries サーバー名が表示されます。表示されない場合は、そのフィールドにサーバー名を入力してください。

「ターゲット」域で、以下のようになります。

3. 「ライブラリー」フィールドに「RSELABxx」と入力します。
4. このソース物理ファイル内のすべてのメンバーを検索するために、「ファイル」フィールドに「QDDSSRC」と入力します。
5. 「メンバー」フィールドに「\*ALL」と入力します。
6. 「検索」をクリックします。

マルチファイル検索ウィンドウに、ENHRS を参照する、すべてのファイル内の行がすべてリストされます。

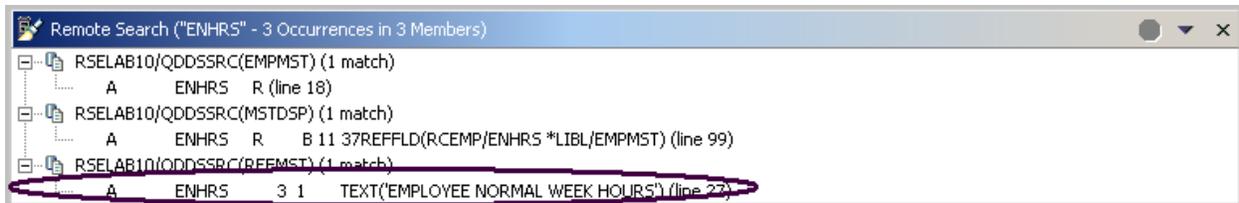


図 38: 検索結果

7. リストの最後の行をダブルクリックします。

A	ENHRS	3	1	TEXT('EMPLOYEE NORMAL WEEK HOURS')
---	-------	---	---	------------------------------------

Row	Column 1	Replace
000020	A	COLHDG('EMP'
000021	A	EDEPT 5 TEXT('EMPLOY
000022	A	COLHDG('EMPL
000023	A	ELOCN 30 TEXT('EMPLOY
000024	A	COLHDG('EMPL
000025	A	EUSRI 8 TEXT('EMPLOY
000026	A	COLHDG('EMPL
000027	A	ENHRS 3 1 TEXT('EMP
000028	A	COLHDG('NORM
000029	A	EPHRC 5 1 TEXT('PROJEC
000030	A	COLHDG('PRJ
000031	A	EPHRY 7 1 TEXT('PROJEC
000032	A	COLHDG('PRJ
000033	A	EPHRP 7 1 TEXT('PROJEC
000034	A	COLHDG('PRJ

Remote Search ("ENHRS" - 3 Occurrences in 3 Members)

- A ENHRS R (line 18)
- RSELAB10/QDDSSRC(MSTDSP) (1 match)
- A ENHRS R B 11 37REFFLD(RCEMP/ENHRS \*LIBL/EMPST) (line 99)
- RSELAB10/QDDSSRC(REFMST) (1 match)
- A ENHRS 3 1 TEXT('EMPLOYEE NORMAL WEEK HOURS') (line 27)

メンバー REFMST がエディターに自動的にロードされ、カーソルが正しい行に置かれます。これで完了です。

## 「リモート・システム」ビューからのファイルの相違の比較

ご使用の製品で多くの変更が行われている場合、比較ユーティリティーが役立ちます。このユーティリティーを使用すれば、さまざまなバージョンのプログラムを比較して、その相違を見つけることができます。比較を行う方法は2つあります。ワークベンチで比較ユーティリティーを使用するか、または CODE ツールで比較ユーティリティーを使用します。CODE ツールでの比較の方が直感的ですが、CODE エディターをワークベンチの外で開始する必要があります。

ワークベンチで比較ユーティリティーを使用する場合、2つのファイルを比較することによってその相違を表示できます。異なるファイルを比較することも、ワークベンチ内のバージョンをリポジ

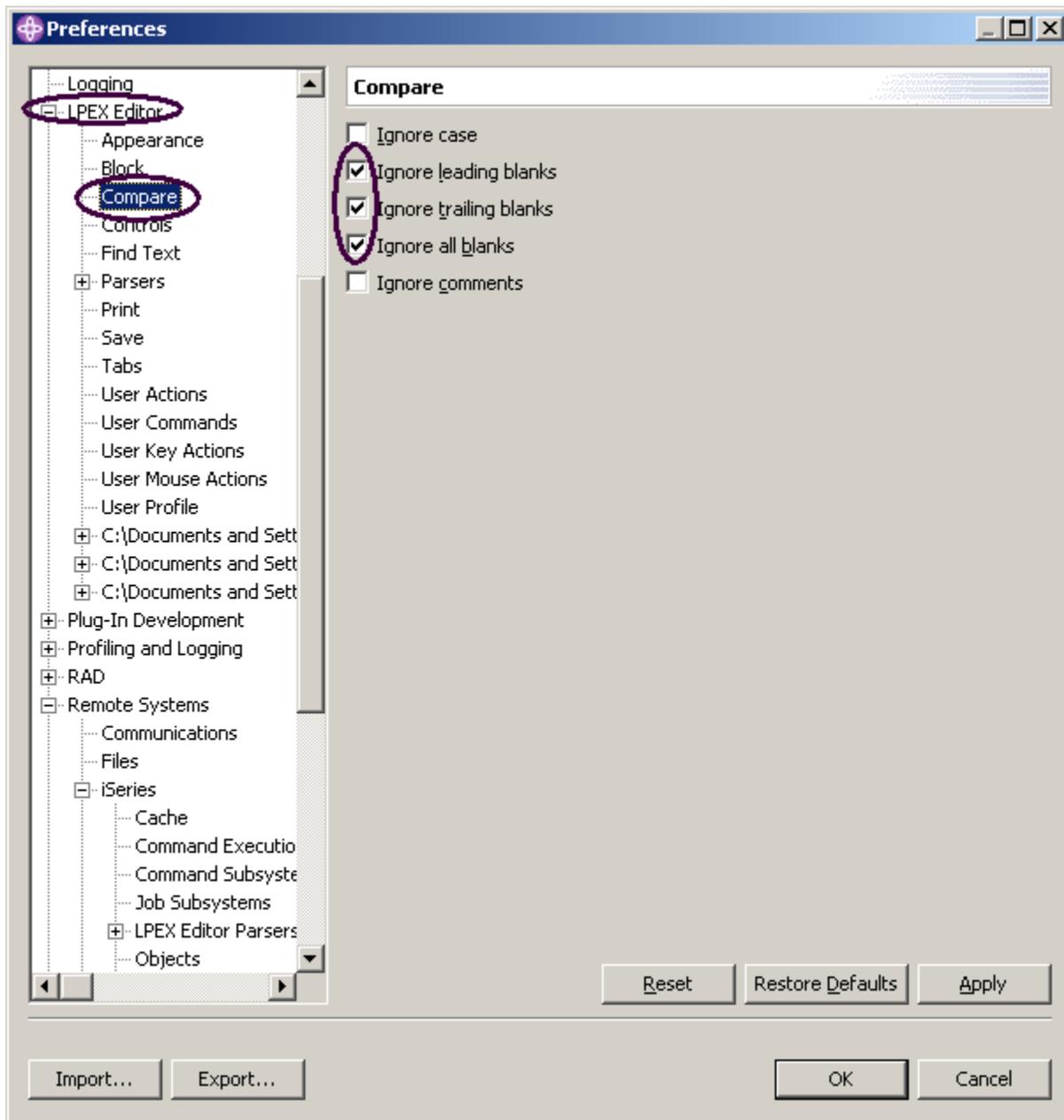
トリー内のバージョンまたはローカル編集履歴と比較することもできます。場合によっては、**3**つのファイルと比較することもできます (共通の旧バージョンが存在する場合)。

比較が行われた後、エディター域で比較エディターがオープンします。比較エディターでは、すべての相違を参照し、比較されたリソース間で、強調表示された相違をコピーできます。比較エディターで行われたリソースの変更は保管できます。

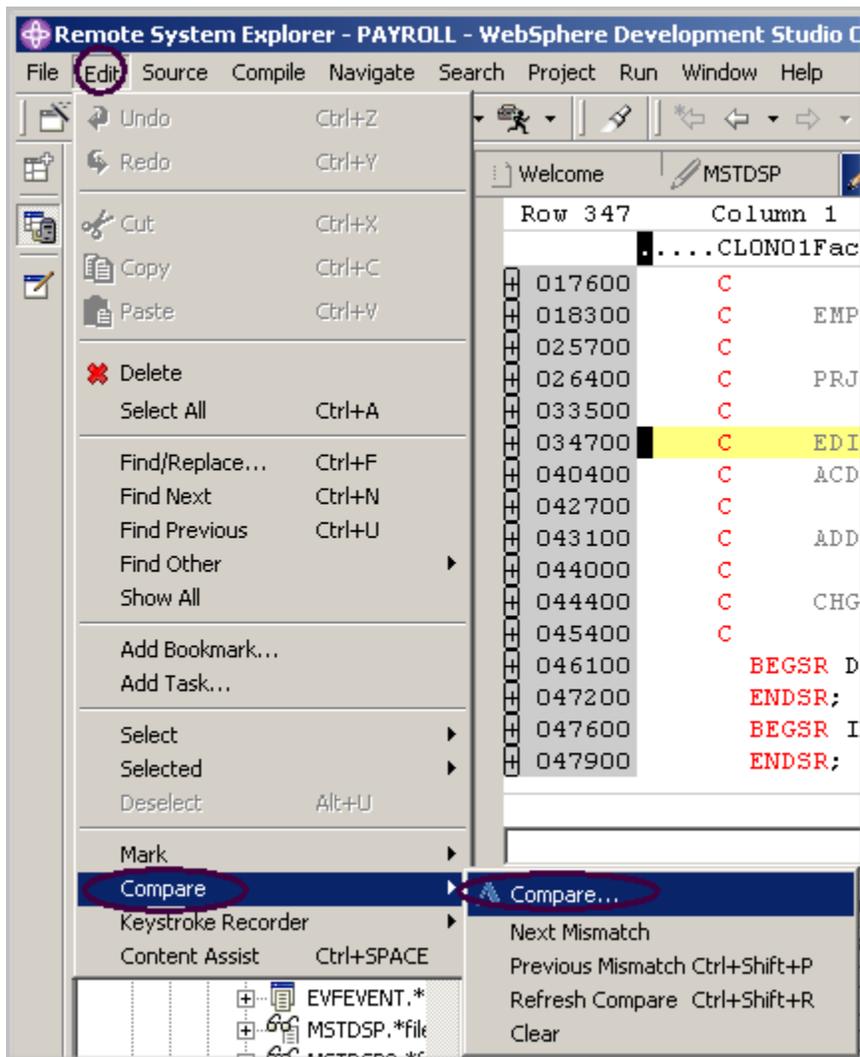
**CODE** で比較ユーティリティを使用する場合も、**2**つのファイルと比較することによってその相違を表示できます。**CODE** エディター・ビューで、ファイルと比較するファイルの名前を入力します。ファイル名を入力するか、またはエディターですでにオープンされているファイルのリストから**1**つを選択することができます。まだエディターでオープンされていないファイルの名前を入力すると、そのファイルがエディターにロードされます。ファイルが指定されない場合、現行ファイルが新規の無表題ファイルと比較されます。現行ファイルが「比較」ビューの左側に表示され、指定されたファイルが右側に表示されます。「比較」メニューを使用して、前後のミスマッチを表示したり、オプション (大文字小文字の無視、フォント、ビューの保護、およびミスマッチのみの表示など) を選択したりします。

ワークベンチでファイルを比較するには、以下のようにします。

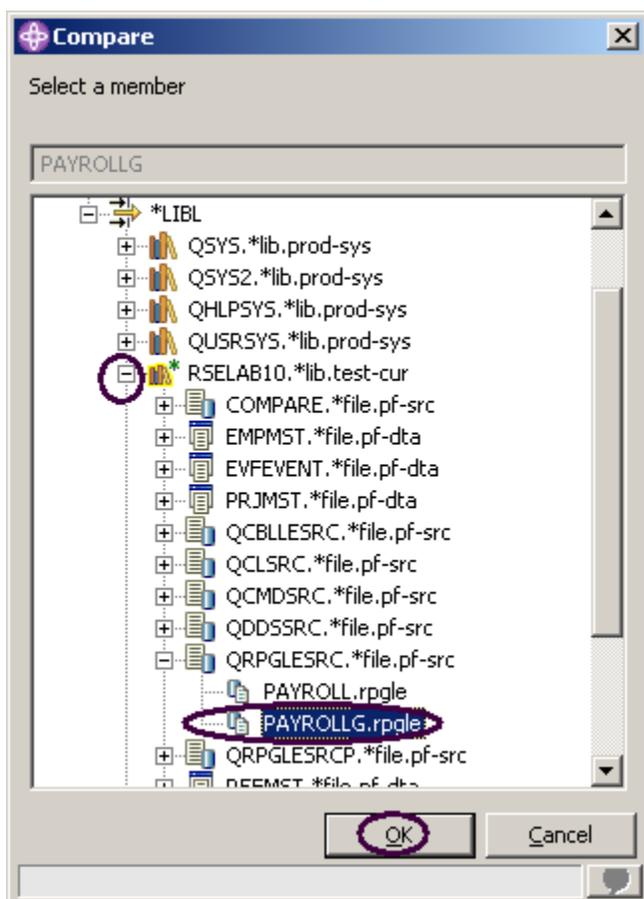
1. ワークベンチ・メニュー・バーから「**ウィンドウ**」>「**プリファレンス**」を選択します。
2. 「プリファレンス」ウィンドウのツリー・ビューで「**LPEX エディター**」を拡張します。
3. 「LPEX エディター」の下の「**比較**」を選択します。



4. 「比較」ウィンドウで、**ブランクの無視**に関するチェック・ボックスが選択されていることを確認します。
  5. 「プリファレンス」ウィンドウで「**OK**」をクリックします。
- PAYROLL メンバーの編集ウィンドウに戻ります。



6. 「PAYROLL」タブをダブルクリックします。
7. ワークベンチ・メニュー・バーから「編集」>「比較」>「比較」を選択します。



「比較」ウィンドウで、以下のようにします。

8. 接続を拡張します。
9. 「\*LIBL」を拡張します。
10. 「RSELABxx」を拡張します。
11. 「QRPGLSRC」を拡張します。
12. メンバー「PAYROLLG」を選択します。
13. 「比較」ウィンドウで「OK」をクリックします。

エディターに、PAYROLL および PAYROLLG という 2 つのメンバーの相違が表示されます。

「編集」メニュー・オプションの下の「比較」メニュー・オプションに戻るか、ショートカット・キー CTRL+Shift+N を使用すると、ミスマッチからミスマッチに移動できます。

PAYROLL と PAYROLLG でのミスマッチは、ミスマッチの行が属するファイルを指し示すことができるように異なる色で強調表示されます。

```

Welcome | MSTDSP | REFMST | PAYROLL X
Row 52   Column 1   Replace
.....1.....2.....3.....4.....5.....
004700   C*
004800   C           EXSR       MAIN
004900   C* If MAIN is done program ends
005000   C           eval       *INLR = *on
005100   * MAIN SUBROUTINE
005200   C           MAIN      DECSR
005300   C           dou       *INKC
005400   C           EVAL     *IN60 = *OFF
005500   C           EVAL     EMESS = *BLANK
005600   C           EVAL     EMPAPL = *BLANK
005700   C           EVAL     PRJAPL = *BLANK
005800   C           EVAL     RSNAPL = *BLANK
005900   C*
006000   C* Write the SELECT format to display. If end of
One or more mismatches found.

```

次に、比較セッションを終了します。

14. ワークベンチ・メニュー・バーから「編集」>「比較」>「消去」を選択します。

### CODE エディターからのファイル変更の比較 (任意選択)

CODE ツールには、比較されているメンバーの横並びビューが用意されています。このようなビューを使用したい場合は、下で説明されているステップに従います。上で使用した比較ツールで十分な情報が得られる場合は、このセクションをスキップしても構いません。

ここでは、2つのファイルオープンし、それを編集して、CODE 比較ユーティリティを使用します。

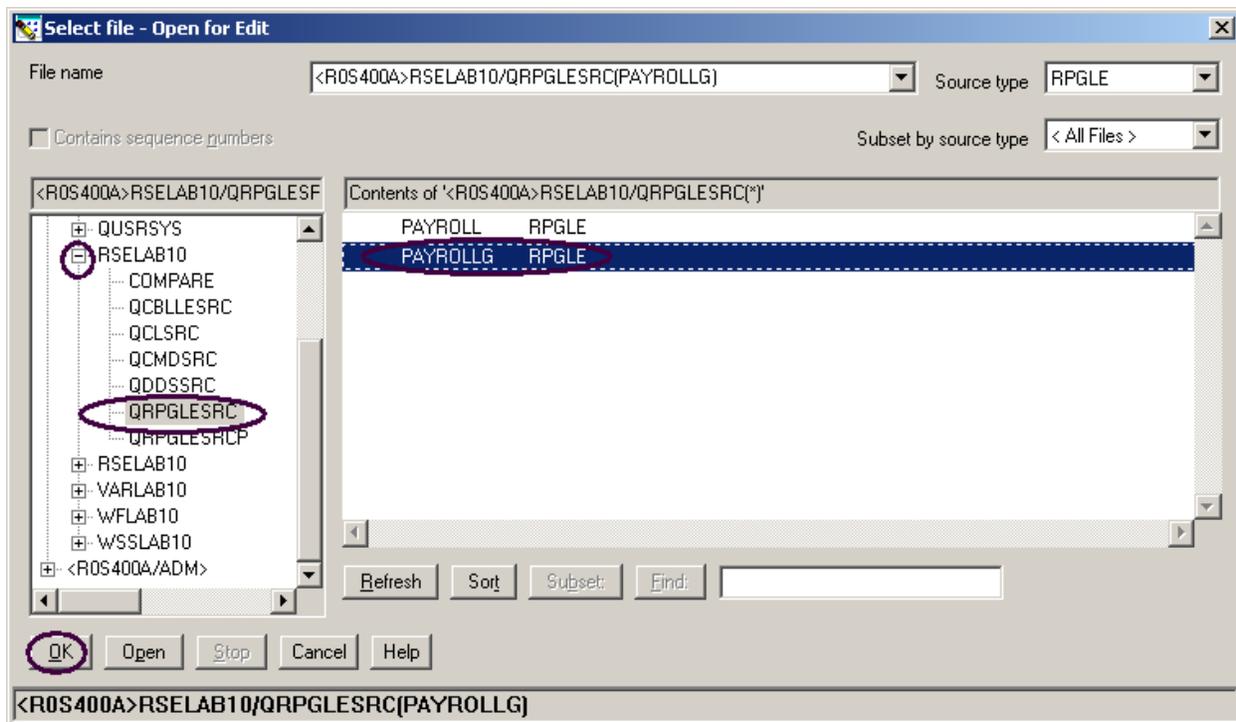
「リモート・システム」ビューで、以下のようにします。

1. 「QRPGLESRC」内のメンバー「PAYROLL」を右クリックします。
2. ポップアップ・メニューから「アプリケーションからオープン」>「CODE エディター」を選択します。

CODE エディター・ウィンドウがオープンし、PAYROLL メンバーが表示されます。このウィンドウは、LPEX エディター・セッションによってロックされているため、参照モードで表示されます。

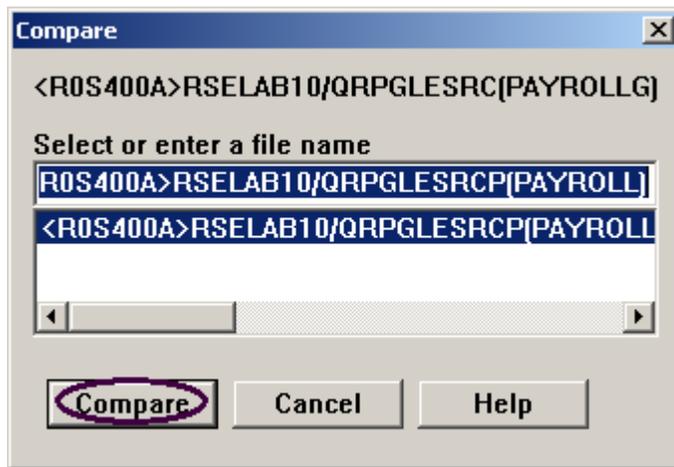
CODE エディターで PAYROLLG メンバーをオープンします。

- CODE エディター・メニュー・バーから「ファイル」>「オープン」オプションを選択します。



オープン・ウィンドウで、以下のようにします。

- 「ROS400A」接続 (リモート・システム・エクスプローラー接続) を拡張します。
- ライブラリー「RSELABxx」を拡張します。
- ファイル「QRPGLESRC」を選択します。
- 右側のメンバー・リストからメンバー「PAYROLLG」を選択します。
- 「ファイルの選択 - 編集用にオープン」ウィンドウで「OK」をクリックします。
- 「アクション」メニューから「比較」を選択します。「比較」ウィンドウが表示されます。



すべてのエントリーがプリロードされます。

10. 「比較」をクリックします。

これで、エディターの左側に PAYROLLG メンバーがロードされ、右側にメンバー PAYROLL がロードされました。2つのメンバーの間には2つの長い垂直で青色の線があり、2つのメンバーの相違が黄色の横線と赤色の横線で強調表示されています。

CODE - <R0S400A>RSELAB10/QRPGLESRC(PAYROLLG): 2

File Edit View Actions Compare Options Windows Help

<R0S400A>RSELAB10/QRPGLESRC(PAYROLLG): 2

Row 1	Column 1	Replace
000001	F*	*****
000002	F*	PROGRAM NAME - Payroll
000003	F*	DESCRIPTION - Time report
000004	F*	externally
000005	F*	*****
000006	F*	INDICATORS USED
000007	F*	50 - No record found on C
000008	F*	60 - General error condit
000009	F*	90 - Protect display on d
000010	F*	KC - End of job requested
000011	F*	KD - Return to applicatio
000012	F*	KE - Return to employee s
000013	F*	KF - Return to project se
000014	F*	KG - Return to reason cod
000015	F*	LR - Last record
000016	F*	*****
000017	F*	SUBROUTINES USED
000018	F*	EDITSL - Edit application
000019	F*	ACDESR - Edit action code
000020	F*	*****
000021	F*	This program uses all ext
000022	F*	used are - MSTDSP - main
000023	F*	- EMPMST - empl
000024	F*	- PRJMST - proj
000025	F*	- RSNMST - reas
000026	F*	*****
000027	FMSTDSP	CF E
000028	FEMPMST	UF A E K
000029	FPRJMST	UF A E K
000030	FRSNMST	UF A E K
000031	D*	
000032	D*	Compile time array contai
000033	D ERR	S
000034	D EMESS	S
000035	E*	

IBM Live Parsing Editor

11. 垂直スクロール・バーを使用してファイル内を移動します。スクロールするにつれて、メンバーの相違の場所が表示されます。RPG エキスパートであれば、PAYROLL にいくつかエラーがあることに気がきます。これらをすぐに修正しましょう。

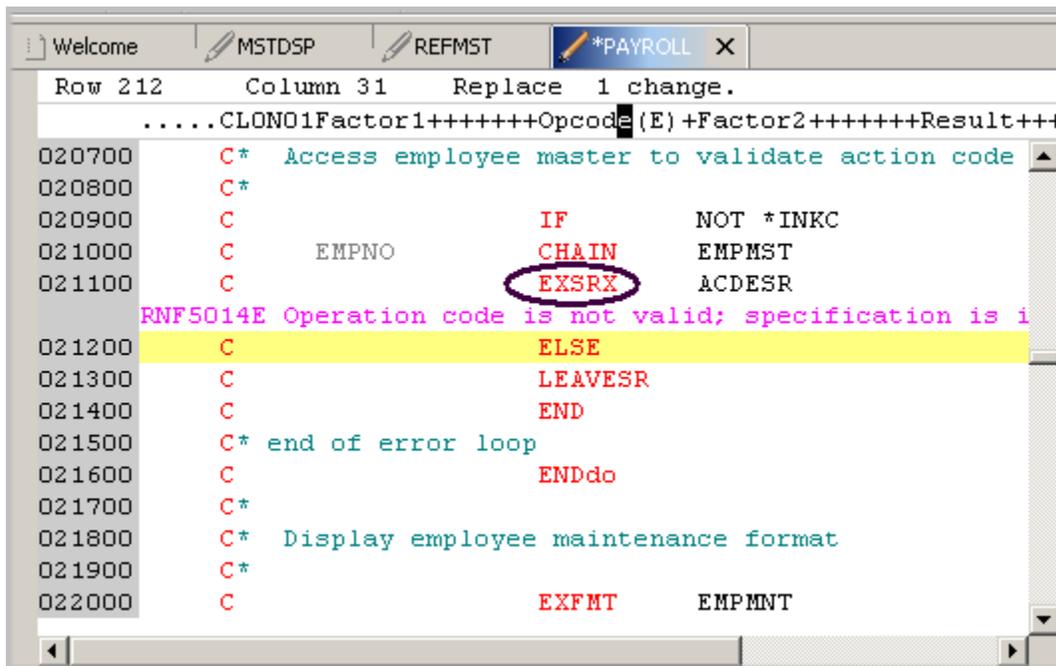
12. 「比較」メニュー (このアクションの実行中に挿入されました) から「比較の終了」を選択して、元のビューに戻ります。
13. CODE エディターをクローズします。メニュー・バーから「ファイル」>「終了」を選択します。ワークベンチでの作業を続けます。

## 構文の検査

LPEX エディターが SEU と共用する強力なフィーチャーの 1 つは、ソースの構文検査機能です。構文検査を行うことができるのは、現在選択されているソースまたはファイル全体で、各ソース行またはすべての行からカーソルが離れたときです。

ここでは、構文エラーを発生させ、プロンプトでそのエラーを訂正します。

1. PAYROLL エディター・ウィンドウで、「EXSR        ACDESR」と表示されている 211 行目にカーソルを移動します。  
すでにその行にカーソルが移動している場合もありますが、移動していない場合は、シーケンス番号桁に行番号を入力するか、またはスクロールダウンします。
2. 「EXSR」命令コードに「X」を追加し、「EXSRX」にします。
3. その行からカーソルを移動します。  
エラーに注意を引くためのエラー・メッセージが表示されます。



```
Row 212      Column 31      Replace 1 change.
.....CLON01Factor1+++++Opcode(E)+Factor2+++++Result+++
020700      C*   Access employee master to validate action code
020800      C*
020900      C           IF           NOT *INKC
021000      C           EMPNO      CHAIN      EMPMST
021100      C           EXSRX      ACDESR
RMF5014E Operation code is not valid; specification is i
021200      C           ELSE
021300      C           LEAVESR
021400      C           END
021500      C*   end of error loop
021600      C           ENDdo
021700      C*
021800      C*   Display employee maintenance format
021900      C*
022000      C           EXFMT      EMPMNT
```

図 39: 構文エラーが表示されたエディター・ウィンドウ

4. ピンクのエラー・メッセージ上にカーソルを移動します。
5. **F1** を押します。

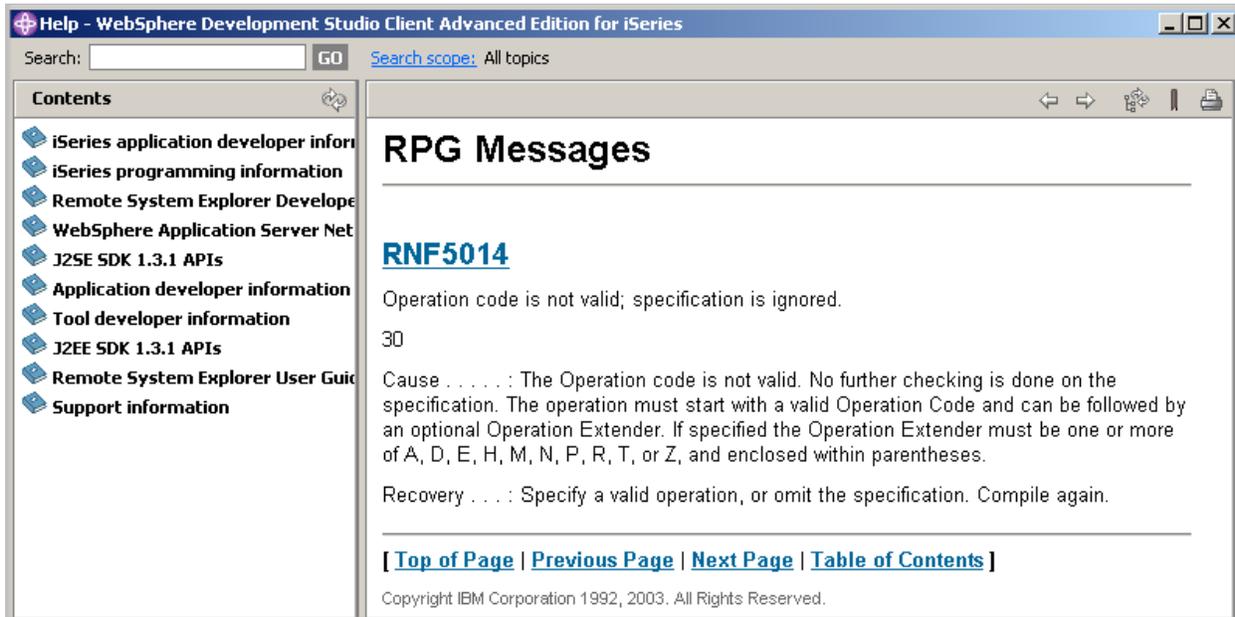


図40: 構文エラーの2次レベル・ヘルプ

ウィンドウがオープンし、エラーの2次レベル・ヘルプが表示されます。

6. 「ヘルプ」ウィンドウを最小化します。
7. 「EXSRX」を「EXSR」に変更して、エラーを訂正します。
8. 修正した行からカーソルを移動します。

エラー・メッセージがエディターから自動的に除去されます。

**ヒント:** 自動構文検査は切り替えることができます。切り替えるには、ワークベンチ・メニュー・バーの「ウィンドウ」>「プリファレンス」オプションを使用して、「プリファレンス」ツリー・ビューで「リモート・システム」>「iSeries」>「LPEX エディター構文解析プログラム」>「ILE RPG」を拡張します。

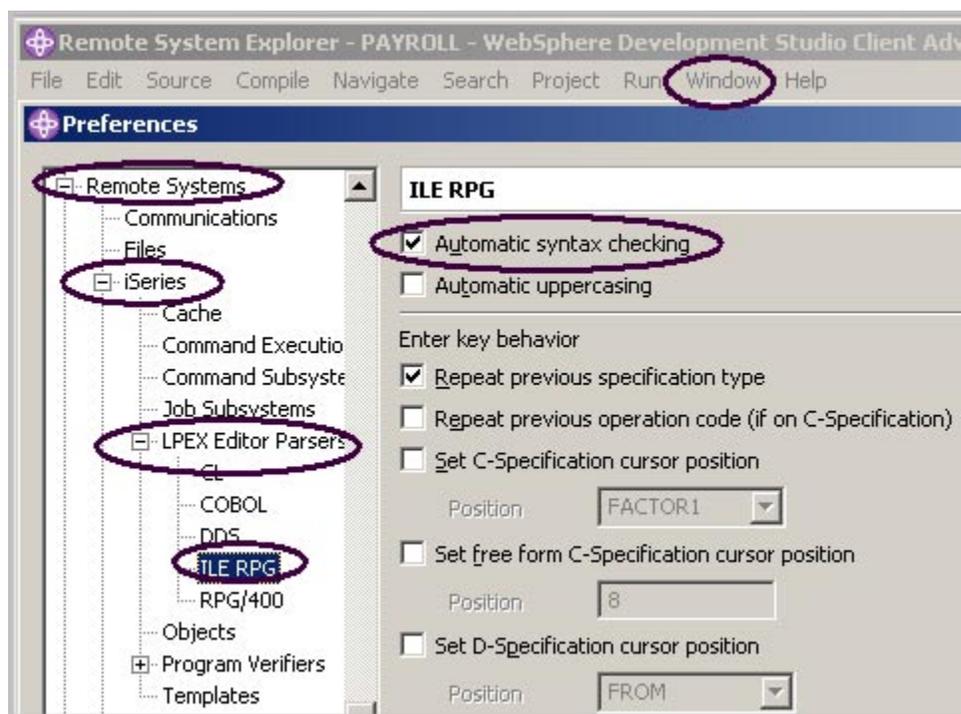


図 41: 構文検査のオンオフを切り替える「プリファレンス」ウィンドウ

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

1. 桁依存エディターが選択されたときに行われること:
  - A. それぞれの桁は別々のエンタリー・スペースとみなされる。
  - B. 演算項目 2 エンタリー内のストリングとの間で文字を挿入または削除しようとしても、結果フィールド・エンタリーは移動しない。
  - C. デフォルトのエディター・プリファレンスでは、桁依存編集はオフ。
  - D. 上記すべて
2. LPEX エディターには事前定義された設定値がありますが、定義可能な設定を含む関連プリファレンス・ページもあります。(T, F)
3. LPEX エディターのプリファレンスを設定する場所:
  - A. 「プリファレンス」ウィンドウ
  - B. エディター・ウィンドウ
  - C. 新規ウィザード
  - D. 「リモート・システム」ビュー

4. 一般的な多くのエディターのキーボードおよびコマンド個別設定を採用するように LPEX エディターを構成することができます。(T, F)
5. ファイルに対して行われた一連の変更を元に戻す場合は、\_\_\_\_\_ 操作を使用しなければなりません。空欄の操作名を指定하십시오。
  - A. 挿入
  - B. 置換
  - C. 再実行
  - D. 元に戻す
6. \_\_\_\_\_ 操作を使用すれば、元に戻す操作の効果をキャンセルすることもできます。空欄の操作名を指定してください。
  - A. 挿入
  - B. 置換
  - C. 再実行
  - D. 元に戻す
7. RPG 仕様のフィールドの順序または可変フィールドに入力できる値を思い出すことができない場合、下のヘルプが非常に役立ちます。そのヘルプ名を指定してください。最も適切な答えを選択してください。
  - A. コンテキスト依存ヘルプ
  - B. 言語依存ヘルプ
  - C. 表示ヘルプ
  - D. フィールド・ヘルプ
8. 言語依存のヘルプを表示するには、\_\_\_\_\_ キーを編集ウィンドウで押します。空欄のキー名を指定してください。
  - A. F2
  - B. F3
  - C. F1
  - D. F4
9. カーソルが命令コードの \_\_\_\_\_ にある場合に、命令コードのヘルプが表示され、それ以外の場合は、現行仕様のヘルプが表示されます。
  - A. 前
  - B. 後
  - C. 上
  - D. 離れた位置
10. エディター・ウィンドウでコードを直接入力または変更する代わりに、\_\_\_\_\_ を使用できません。
  - A. プロンプト
  - B. フィルター
  - C. SEU コマンド
  - D. フォーマット表示行
  - E. 上記すべて

11. \_\_\_\_\_ ビューは隠しビューであり、素早く可視にすることができます。その動作は通常のビューと同じですが、隠されたときにワークベンチ・ウィンドウの画面スペースを使用しないという点が異なります。
- A. リモート・システム
  - B. ナビゲーター
  - C. アウトライン
  - D. 高速
12. \_\_\_\_\_ オプションを使用すれば、構文を持つソースを字下げモードで表示できます。デフォルトで、\_\_\_\_\_ オプションは画面を横方向に分割し、下のペインに字下げされたビューを表示します。
- A. 高速ビュー
  - B. アウトデント
  - C. 字下げ
  - D. 編集
13. 字下げされたビューは参照モードのみであり、編集できません。(T, F)
14. 項目の検索には \_\_\_\_\_ ウィンドウを使用します。最も適切な答えを選択しなさい。
- A. 検索
  - B. 検出
  - C. 編集
  - D. 検索および置換
15. 検索できるもの:
- A. ワード
  - B. ワードの一部
  - C. 一連のワード
  - D. パターン (正規表現の規則に従う場合)
  - E. 上記すべて
16. LPEX エディターを使用すれば、ソースを \_\_\_\_\_ またはサブセット化し、一定のストリングを含む行のみを参照することができます。
- A. 検索
  - B. 検出
  - C. ソート
  - D. フィルター操作
17. ILE RPG ソースを素早くナビゲートできるように、エディターでは \_\_\_\_\_ を基にして行にフィルターを掛けることができます。最も適切な答えを選択しなさい。
- A. ストリング
  - B. 行タイプ
  - C. 行番号
  - D. 上記すべて
18. ソース物理ファイル内のメンバーを介して、またはローカル・ディレクトリー内のファイルを介して検索を行う場合、\_\_\_\_\_ ツールを使用できます。最も適切な答えを選択しなさい。
- A. 比較
  - B. 検索

- C. 検出
  - D. 編集
19. \_\_\_\_\_ ツールを使用すれば、さまざまなバージョンのプログラムを比較し、その相違を見つけることができます。最も適切な答えを選択しなさい。
- A. 変換
  - B. マイグレーション
  - C. 比較
  - D. 検索
20. ファイルを比較する方法は 2 つあります。それは、ワークベンチの比較ツールと CODE エディターの比較ツールです。(T, F)
21. 構文検査を行うことができるのは、現在選択されているソースまたはファイル全体で、各ソース行またはすべての行からカーソルが離れたときです。(T, F)

## 実習

この練習問題ではリモート・システム LPEX エディターの使用方法を学びました。そこで、次の新規タスクを試してみます。トークン強調表示が ILE RPG ソースでどのように機能するのかを確認します。フォーマット表示行を表示し、それを使用してソースを入力します。コメントでソースにフィルターを掛けます。自動大文字変換をオンに設定します。タスクの途中で困ったときは、Development Studio Client for iSeries のオンライン・ヘルプを使用してください。

## ここで学んだ内容

この練習問題では、リモート・システム LPEX エディターをオープンし、エディター・フィーチャーについて学びました。学んだエディター・フィーチャーは、桁依存編集、SEU コマンド、元に戻す操作と再実行操作、言語依存ヘルプ、プロンプト、ソースの字下げ、検索および置換オプション、ストリングによる行のフィルター操作、行タイプによる行のフィルター操作、ファイルの検索、ファイルの比較、および構文検査です。

次の練習問題では、ソースを検査して、iSeries システムでクリーン・コンパイルを確実に行います。このアプローチで、iSeries のサイクルが軽減されます。「リモート・システム」ビューから検査とコンパイルの両方を実行してください。

## 練習問題 5: RPG の検査とコンパイル

この練習問題では、リモート・システム LPEX エディターで ILE RPG を検査およびコンパイルします。プログラムの検査とプログラムのコンパイルは、プログラムまたはモジュールをコンパイルするときの最終ステップです。どちらかでエラーが検出されると、iSeries エラー・リストが表示されます。iSeries エラー・リストは、リモート・システム LPEX エディターで検査ユーティリティーとコンパイル・ユーティリティーによって検出されたエラーを管理する強力なツールです。これらのツール、iSeries エラー・リストのさまざまな機能、および作成した RPG プログラムについて学びます。

この練習問題を終了すると、以下のことができるようになります。

- プログラム検査プログラムの目的を説明する。
- プログラム検査プログラムを起動する。
- iSeries エラー・リストを使用して、メッセージを挿入し、エラーが修正されたかどうか検査する。
- リモート・システム・エクスプローラーで iSeries テーブル・ビューを使用して、iSeries コマンドを実行依頼する。
- コンパイル・ユーティリティーを使用して、コンパイル・オプションを選択し、ILE RPG ソースをコンパイルする。
- 生成された ILE RPG プログラム PAYROLL について説明する。

### 検査プログラムの起動

ここでは、リモート・システム・エクスプローラーの最も強力な固有のフィーチャーの 1 つであるプログラム検査プログラムを取り上げます。iSeries でコードをコンパイルする前に、プログラム検査プログラムを起動することでエラーがないことを確認できます。検査プログラムはワークステーションで意味 (コンパイル) エラーを検査し、ユーザーが iSeries でクリーン・コンパイルを確実に行うことができるようにします。軽減されるホスト・サイクルについて考えてみてください。特に、iSeries から切断された状態でコードを作成しているときに、この検査プログラムは役に立ちます。このようなことができるのは、リモート・システム・エクスプローラーがコードの構文解析と検査を iSeries システム・コンパイラからワークステーションに移植したためです。「iSeries エラー・リスト」ウィンドウでは、検出されたエラーとその重大度がリストされ、エラー・メッセージがソースに直接挿入されます。このウィンドウは、エラー間をナビゲートするときに役立ちます。

検査プログラムを起動するには、次のようにします。

1. ワークベンチ・メニュー・バーから「ソース」>「検査」を選択します。

しばらくすると、検査プログラムによりエディター・ウィンドウの下に iSeries エラー・リストが表示されます。

ID	Message	S...	Line	Location	Connection
RNF5178	ENDSR operation is missing for subroutine...	30	1	RSELAB10/QRPG...	s400a
RNF7030	The name or indicator EMES is not defined.	30	365	RSELAB10/QRPG...	s400a
RNF7030	The name or indicator RSNTAX is not defi...	30	95	RSELAB10/QRPG...	s400a
RNF7018	The operand RSNTAX of EXSR is not a su...	30	95	RSELAB10/QRPG...	s400a
RNF5184	Factor 2 must be '0' or '1' when the Resul...	30	364	RSELAB10/QRPG...	s400a
RNF7515	Move operands ERR and EMES have type...	30	365	RSELAB10/QRPG...	s400a
RNF7089	RPG provides Separate-Indicator area for...	0	27	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator RSNTAG is not refere...	0	110	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator *IN03 is not refere...	0	1	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator *IN04 is not refere...	0	1	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator *IN05 is not refere...	0	1	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator *IN06 is not refere...	0	1	RSELAB10/QRPG...	s400a
RNF7031	The name or indicator *IN07 is not refere...	0	1	RSELAB10/QRPG...	s400a

図42: 検査プログラム・エラー・リスト

エラー・リストには、以下の内容が表示されます。

1. エラー・メッセージ
2. 重大度
3. 行番号
4. ソース位置
5. 接続名

#### エラーの修正

次は、ソース内のエラーを修正します。

ソース内のエラーを修正するには、エラー・リストを表示します。

1. エラー「**RNF7030**」をダブルクリックします。  
エディター・ウィンドウのエラー発生行に自動的に戻ります。365 行目のエラーはタイプミスです。

変数 EMES は、実際は EMESS です。正しい名前を検索するための良い方法の 1 つとして、コンテンツ・アシストがあります。

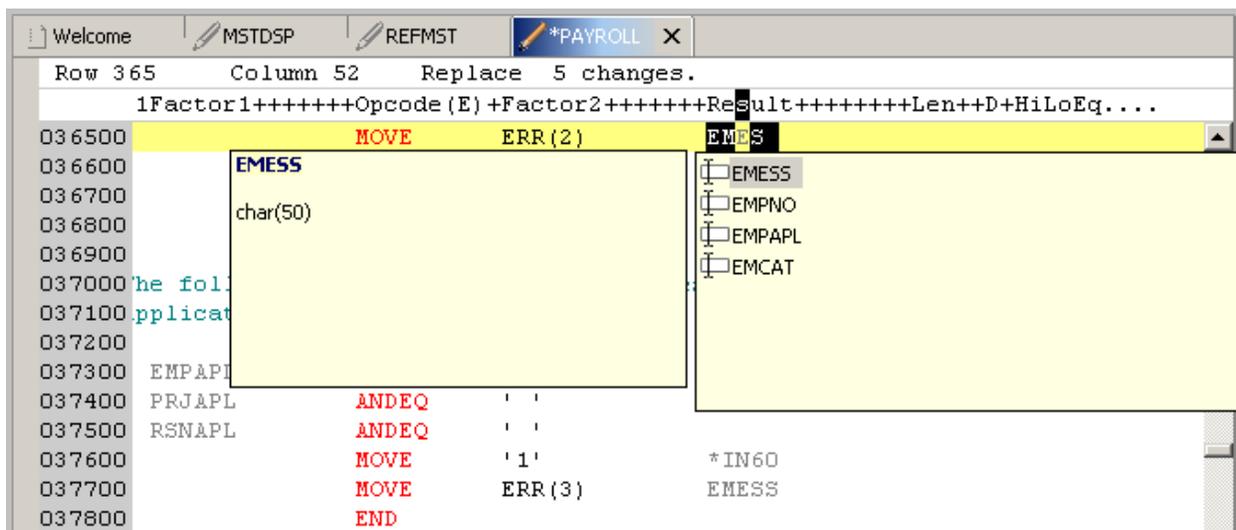


図 43: コンテンツ・アシスト

2. ミススペルの変数を選択し、**CRTL+spacebar** を押します。  
 変数が正常に開始された場合は、その選択項目に正しい名前が使用されています。リスト内の変数 EMESS をダブルクリックして、変数名を訂正します。  
 変数名を検索する別の方法として、「アウトライン」ビューを使用して、宣言されている変数を参照する方法があります。

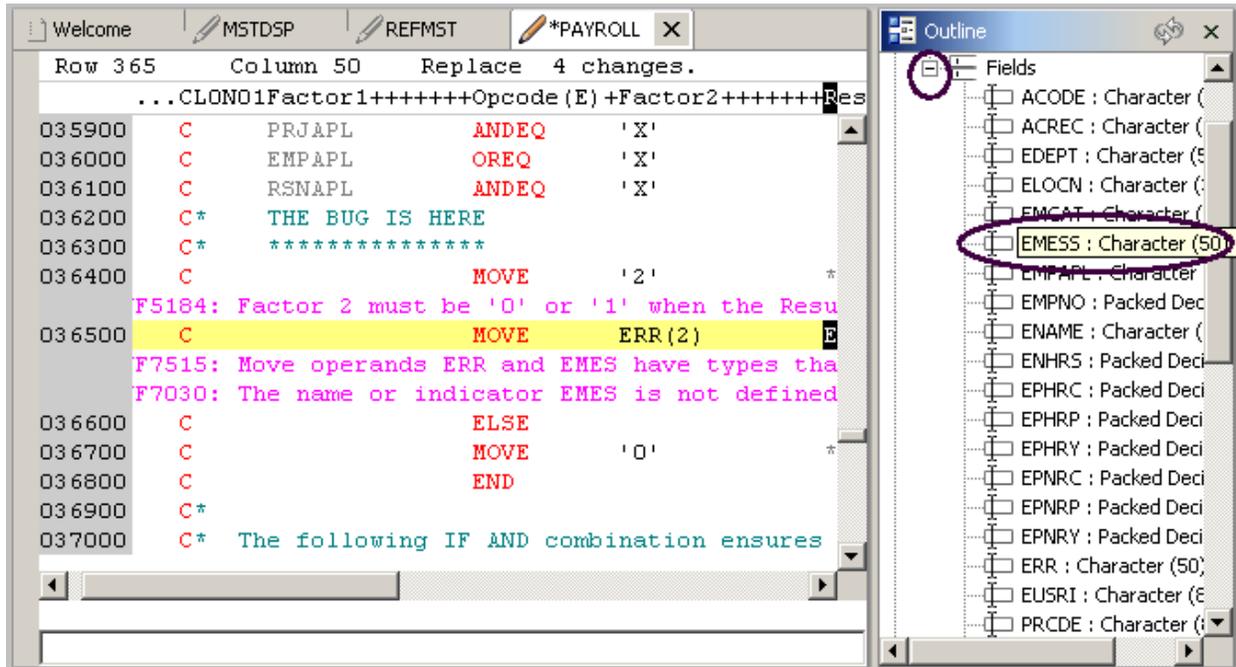


図44: 「アウトライン」ビューと拡張されたフィールド

次のエラーも **RNF7030** です。

- これをダブルクリックします。  
エディターで修正します。

- 「**RSNTAX**」は正しくは「**RSNTAG**」です。適切に変更します。

- 次のエラー **RNF5184** に進みます。

次のエラーは、実際は **RNF7018** ですが、最初のエラーに関連しているため、無視することができます。これは同じ行番号を示していて、両方のエラーが関連していることを表しています。

- 「**RNF5184**」をダブルクリックします。明らかに、自分の RPG を知らないユーザーが、値 2 を標識に割り当てようとした。

- 「2」を「1」に置き換えてこれを修正します。

エラー **RNF7515** は、修正した最初のエラーに関連しています。このエラーは行番号が同じです。そのため、無視することができます。

重大な残りのエラーは **RNF5178** のみです。このエラーは、「**ENDSR**」が欠落していることによって引き起こされます。

注: 「ENDSR」が欠落している場所を検査プログラムが判別できなかったため、報告された行番号は1です。そのため、このエラー・メッセージを単にダブルクリックすることはできません。欠落しているステートメントが属している場所を調べる必要があります。

```

Welcome | MSTDSP | REFMST | *PAYROLL X
Row 398 Column 7 Replace 8 changes.
.....1.....2.....3.....4.....5.....6.....7.....+
039000 C MOVE '1' *IN60
039100 C MOVE ERR(1) EMESS
039200 C END
039300 C RSNAPL IFNE ' '
039400 C RSNAPL ANDNE 'X'
039500 C MOVE '1' *IN60
039600 C MOVE ERR(1) EMESS
039700 C END
039800 C * ENDSR
039900 C *
040000 C *****
040100 C* ACDESR subroutine verifies the time reporting action codes for
040200 C* all maintenance selections.
040300 C *****
040400 C ACDESR BEGSR

```

図45: 誤ってコメント・アウトされた ENDSR

8. エディターで 398 行目に移動します。
9. C 仕様から「\*」を除去します。

ヒント: Tab キーを使用すれば、適切な桁に素早く移動できます。

これで、すべての非情報エラーが修正されました。

フィルター・メニューを使用すれば、異なる重大度をフィルターに掛けることができます。

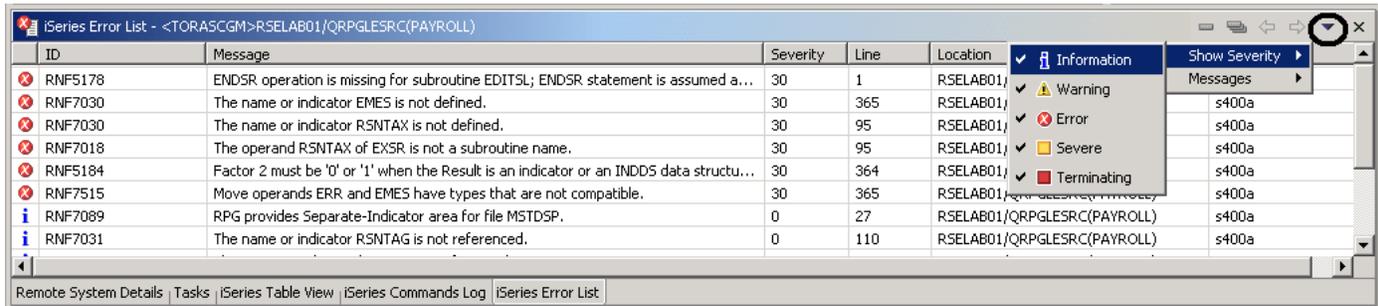


図 46: 重大度レベルに対するフィルター操作

- 図 46: 重大度レベルに対するフィルター操作 に示されるように、ウィンドウ・バーの矢印をクリックします。
- ドロップ・ダウン・メニューから「重大度の表示」を選択します。
- リストに表示したくない重大度 (たとえば「情報」など) を選択解除します。

### ソース・メンバーの保管

ここでは、変更内容を失わないように保管します。PAYROLL メンバーが選択されていることを確認します。次に、ソースをもう一度検査して、すべてのエラーが修正されていることを確認します。次のいずれかの方法で、メンバーを保管できます。

- ワークベンチ・メニュー・バーから「ファイル」>「保管」を選択します。
- ワークベンチ・ツールバーから保管ボタン  をクリックします。
- Ctrl+S** を押す。

変更内容が iSeries にアップロードされます。

- ソースをもう一度検査する。

ID	Message	S...	Line	Location	Connection
RNF7089	RPG provides Separate-Indicator area for...	0	27	RSELAB10/QRPL...	s400a
RNF7031	The name or indicator *IN03 is not refere...	0	1	RSELAB10/QRPL...	s400a
RNF7031	The name or indicator *IN04 is not refere...	0	1	RSELAB10/QRPL...	s400a
RNF7031	The name or indicator *IN05 is not refere...	0	1	RSELAB10/QRPL...	s400a
RNF7031	The name or indicator *IN06 is not refere...	0	1	RSELAB10/QRPL...	s400a
RNF7031	The name or indicator *IN07 is not refere...	0	1	RSELAB10/QRPL...	s400a

図 47: エラー・リスト (情報メッセージのみが残されている)

すべての処理が正常に終了しました。これで、プログラムをコンパイルする準備が整いました。

## リモート・コンパイルの起動

リモート・コンパイル機能は、リモート・システム・エクスプローラーの一部です。この機能は、ワークステーション・インターフェースを提供します。このインターフェースを介して、ホスト上でオブジェクトをコンパイル、バインド、またはビルドするための要求を **iSeries** に実行依頼します。これにより、サポートされているすべての **CRTxxx** コマンドに使用できるすべてのコンパイル・オプションへ容易にアクセスできます。

ローカル・プログラム検査プログラムを使用した場合は、ホストのコンパイルが正常に行われます(無駄な **iSeries** サイクルがありません)。ただし、エラーがある場合、ホスト・コンパイラーはエラー情報をワークステーションに戻し、エラーは「**iSeries** エラー・リスト」ウィンドウにロードされ、そのウィンドウはプログラム検査が行われたときと同じ状態になります。

プログラムをコンパイルするときのデフォルトでは、バッチ・ジョブ待ち行列にコンパイルが実行依頼されます。この練習問題では、対話式のコンパイルを実行できます。

プリファレンスを変更して対話式にコンパイルするには、以下のようにします。

1. ワークベンチ・メニュー・バーから「**ウィンドウ**」>「**プリファレンス**」を選択します。

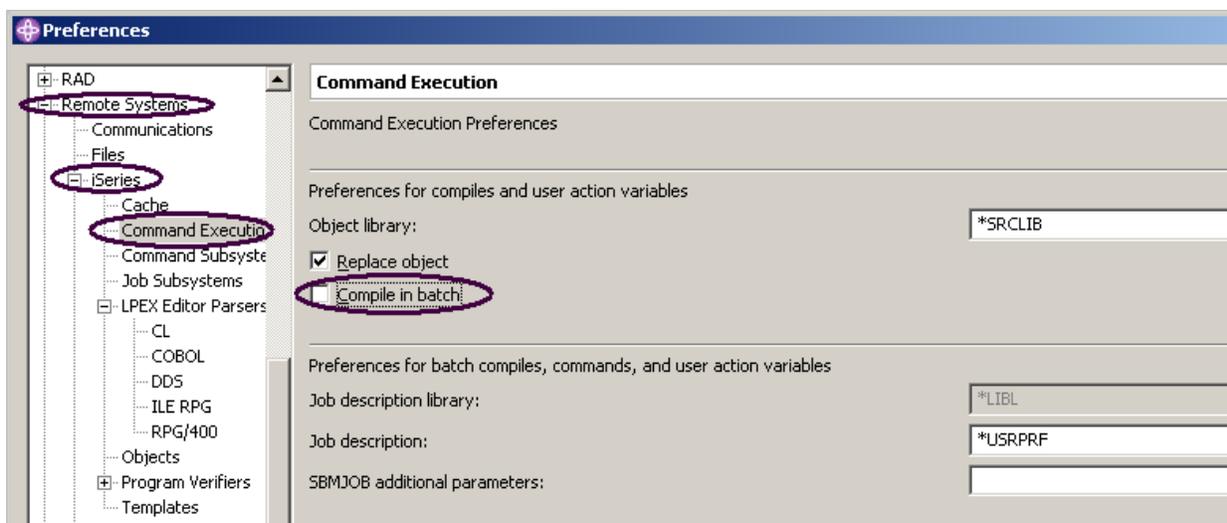


図 48: コンパイル・コマンドのデフォルトの変更

2. 「**リモート・システム**」を拡張します。
3. 「**iSeries**」を拡張します。
4. 「**コマンド実行**」を選択します。
5. 「**バッチでのコンパイル**」チェック・ボックスを選択解除します。

6. 「OK」をクリックして、リモート・システム・エクスプローラー・パースペクティブに戻ります。

## コンパイルの開始

コンパイルの開始方法は、リモート・システム・エクスプローラーで他のコマンドを使用する方法と同様です。ソース・メンバー用のポップアップ・メニューには、コンパイル・サブメニューが含まれています。コンパイル・サブメニューには、さまざまなメンバー・タイプ用のコンパイル・コマンドがいくつかあります。コンパイル名は、コンパイル・コマンドをリモート・システムで実行するための ID です。たとえば、コンパイル・ラベル CRTBNDRPG は、バインドされた RPG プログラムを作成するように iSeries システムに指示します。コンパイル名はソース・メンバー・タイプ (メンバー (\*MBR) およびモジュール (\*MODULE)) にのみ関連付けられます。このため、これらのタイプのエレメントを右クリックしたときのみ、コンパイルするためのオプションが表示されます。コンパイルする方法には、プロンプトありとプロンプトなしの 2 つがあります。どちらのコンパイル・アクションでも、コンパイル名のリスト付きサブメニューが起動されます。

拡張されたライブラリー RSELABxx を示す「リモート・システム」ビューで、以下のようにします。

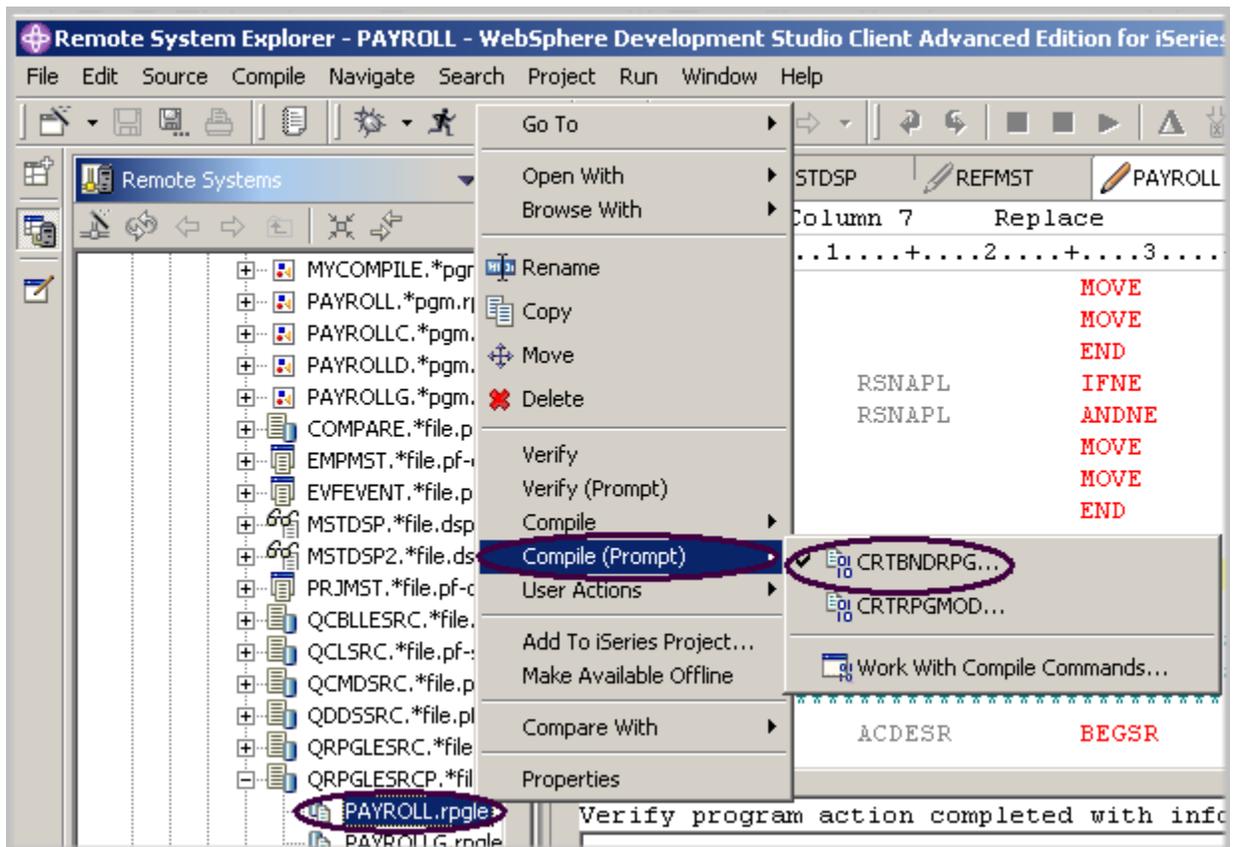


図 49: コンパイルするメンバーの選択

1. 「QRPGLSRC」内のメンバー「PAYROLL」を右クリックします。
2. ポップアップ・メニューから「コンパイル(プロンプト)」>「CRTBNDRPG」オプションを選択します。

## ソースのコンパイル

ここでは、CRTBNDRPG コマンドのプロンプトを使用して、コンパイル・パラメーターを指定します。

名前に関係するすべての入力フィールドには、正しい情報がすでに表示されています。

Program: PAYROLL Name

Library: RSELAB10 Name

Source file: QRPGLESRCP Name, QRPGLESRC

Library: RSELAB10 Name

Source member: PAYROLL Name

Generation severity level: 10 0-20

Text 'description': \*SRCMBRTXT Character value...

Default activation group: \*YES

Compiler options: Advanced Parameters

\*EVENTF

Debugging views: \*ALL

Advanced (X)  All Parameters(Y)  Keywords(Z)

CRTBNDRPG PGM(RSELAB10/PAYROLL) SRCFILE(RSELAB10/QRPGLESRCP) SRCMBR(PAYROLL) OPTION(\*EVENTF) DBGVIEW(\*ALL)

OK Restore defaults Cancel

図 50: CRTBNDRPG のプロンプト

1. 「デバッグ・ビュー」パラメーターを「\*ALL」に変更します。  
その他の選択可能なパラメーターを調べたい場合は、「拡張」をクリックしてください。

2. 完了したら、「OK」をクリックします。

ウィンドウ・ボックスに、コンパイルが開始されたことが示されます。

ワークベンチ上の進行状況表示バー (右下隅) は、コンパイルが実行されていることを示します。エラー・リストが表示されますが、エラーはなく、情報メッセージが表示されます。

表示するメッセージのタイプとして、「情報メッセージ」が選択されていることを確認してください。前の練習問題のトピックでこの選択項目のチェックマークを外した可能性があります。

コンパイルが成功したかどうかは確実にない場合は、次のようにします。

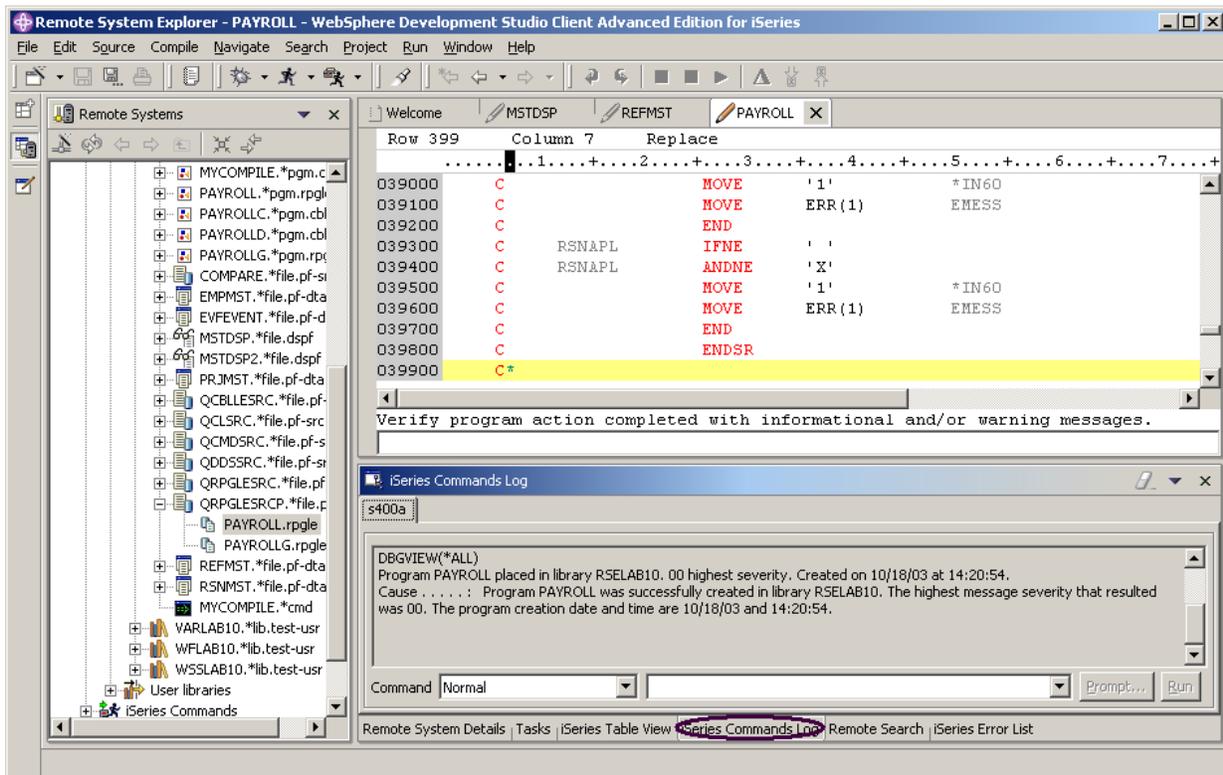


図 51: iSeries コマンド・ログ

3. ワークベンチの下部にある「iSeries コマンド・ログ」タブをクリックします。

このログには、リモート・システムで実行されたすべてのコマンドのリストと、コマンドごとに戻されたメッセージが表示されます。

## iSeries テーブル・ビューでの iSeries コマンドの実行依頼

リモート・システム・エクスプローラー内で iSeries テーブル・ビューを使用すれば、コマンドを iSeries に実行依頼できます。iSeries テーブル・ビューの下にある「コマンド」フィールドでコマンドを実行し、「メッセージ」フィールドにメッセージを表示できます。テーブルを取り込んだ後に、コマンドを入力し、「プロンプト」を選択してパラメーターを指定してから「実行」を選択するか、または単に「実行」を選択することができます(後者の場合は、プロンプトが出され、通常どおりコマンドが実行されます)。コマンドが実行されると、そのコマンドからのメッセージが「メッセージ」ドロップダウン・フィールドに取り込まれます。メッセージを選択すると、「詳細」ボタンが使用可能になります。このボタンを押すと、メッセージとそのヘルプが表示されます。

また、iSeries テーブル・ビューまたは「リモート・システム」ビューを使用すれば、コマンドおよびプログラムを実行できます。iSeries テーブル・ビューでは、すべての項目のプロパティを同時に参照することができます。プロパティはテーブルの行として表示されます。「リモート・システム」ビューでは、ナビゲーションが極めて容易です。iSeries オブジェクト・サブシステムのライブラリー・リストから作業することができます。また、実行したい項目を選択する前に多くの項目の内容を参照することもできます。

ライブラリー・リストを変更するには、以下のようにします。

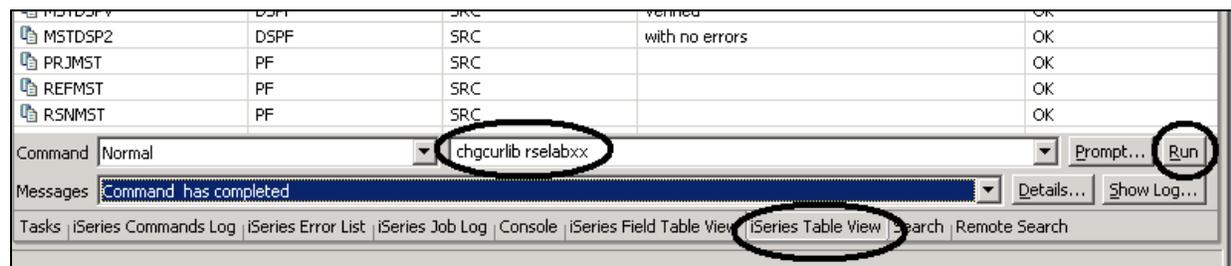


図 52: iSeries テーブル・ビューとコマンド入力

1. ワークベンチの下部にあるビューから「iSeries テーブル・ビュー」タブを選択します。
2. 「コマンド」フィールドに、たとえば「CHGCURLIB RSELABxx」と入力します。**注:** iSeries システムにあるライブラリーを使用してください。
3. 「実行」をクリックします。

iSeries テーブル・ビューを使用してこのビューに iSeries オブジェクトを表示したことがない場合は、テーブル・ビューがアクティブな接続にリンクされていないため、次のようなエラー・メッセージが表示されます。

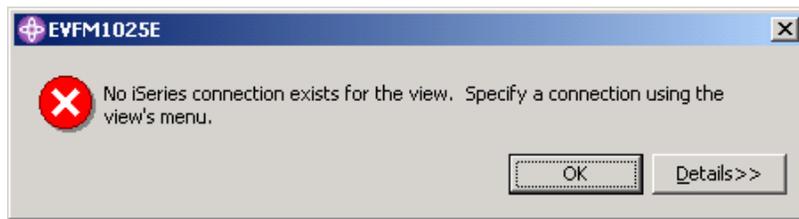


図 53: アクティブな接続がない iSeries テーブル・ビューを使用したときのエラー・メッセージ

このメッセージが表示されたら、「OK」をクリックして、「リモート・システム」ビューに移動します。

4. 「QRPGLESRC」を右クリックします。

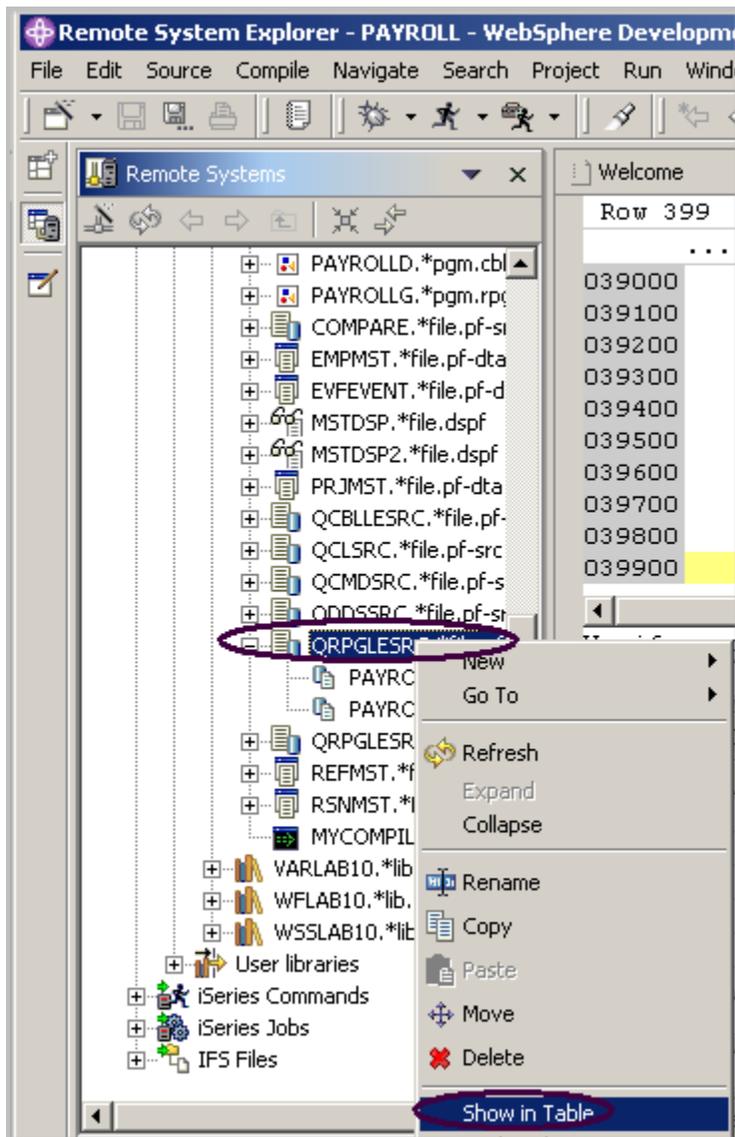


図 54: iSeries に接続するテーブル・ビューの選択

5. ポップアップ・メニューから「テーブルへの表示」オプションを選択します。  
テーブル・ビューに、選択したソース・ファイル内のメンバーが取り込まれます。
6. **CHGCURLIB** コマンドを再実行します。  
コマンドが **iSeries** で実行され、完了すると **iSeries** テーブル・ビューの下部に完了メッセージが表示されます。

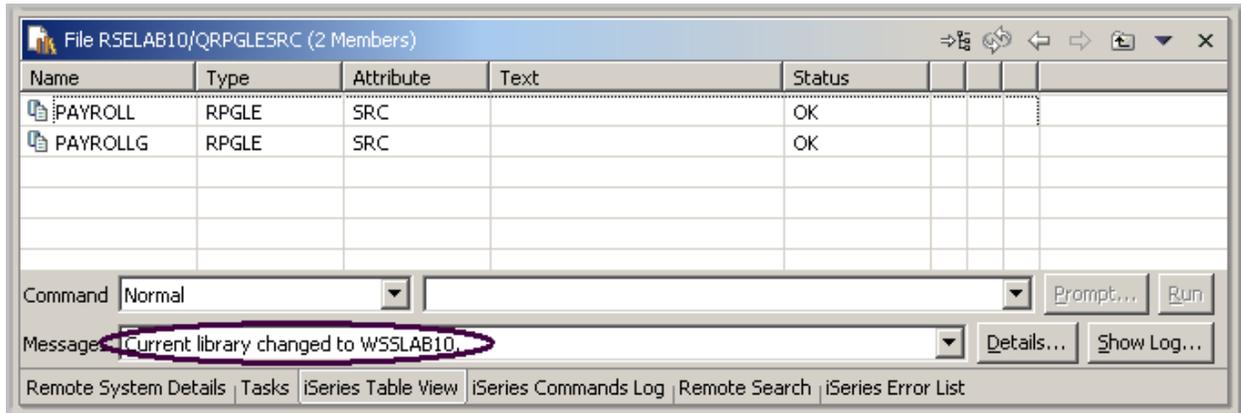


図 55: コマンドの完了メッセージ

**ヒント:** リモート・システム・エクスプローラーおよび起動コマンドを **iSeries** 以外のシステム (ローカル・システム、**LINUX** など) に使用して、**iSeries** 以外のシステムに接続することもできます。

## コマンドとプログラムの実行

以下の 3 つの方法で「リモート・システム」ビューまたは **iSeries** テーブル・ビューからプログラムおよびコマンドを実行できます。

1. リモート・システム・エクスプローラーの通信サーバー・ジョブこれは、現在使用している方法です。
2. バッチ・ジョブ
3. 対話式ジョブ (5250 アプリケーションのテスト用)

最初のオプションを使用した場合、プログラムを通信サーバーと同様のジョブで実行できます。バッチ・ジョブおよび対話式ジョブでは、その状況は容易にはモニターできませんが、通信サーバーとは関連付けず、プログラム・コマンドが終了すると通知されます。バッチ・ジョブは期待通りに作動し、いかなる初期セットアップも必要としません。対話式プログラムには **5250** エミュレーターが必要です。そのため、このエミュレーターをリモート・システム・エクスプローラー通信サーバーの特定の接続と関連付けるには、最初に **STRRSESVR <connectionName>** コマンドを実行する必要があります。

## 対話式接続の開始

対話式接続を開始するには、以下のようにします。

1. **5250** エミュレーション・セッションを開始します。

2. ユーザー ID とパスワードを指定して、iSeries にサインオンします。

**注:** 5250 エミュレーション・セッションでは、**Enter** キーの代わりに **Ctrl** キーを使用しなければならない場合があります。

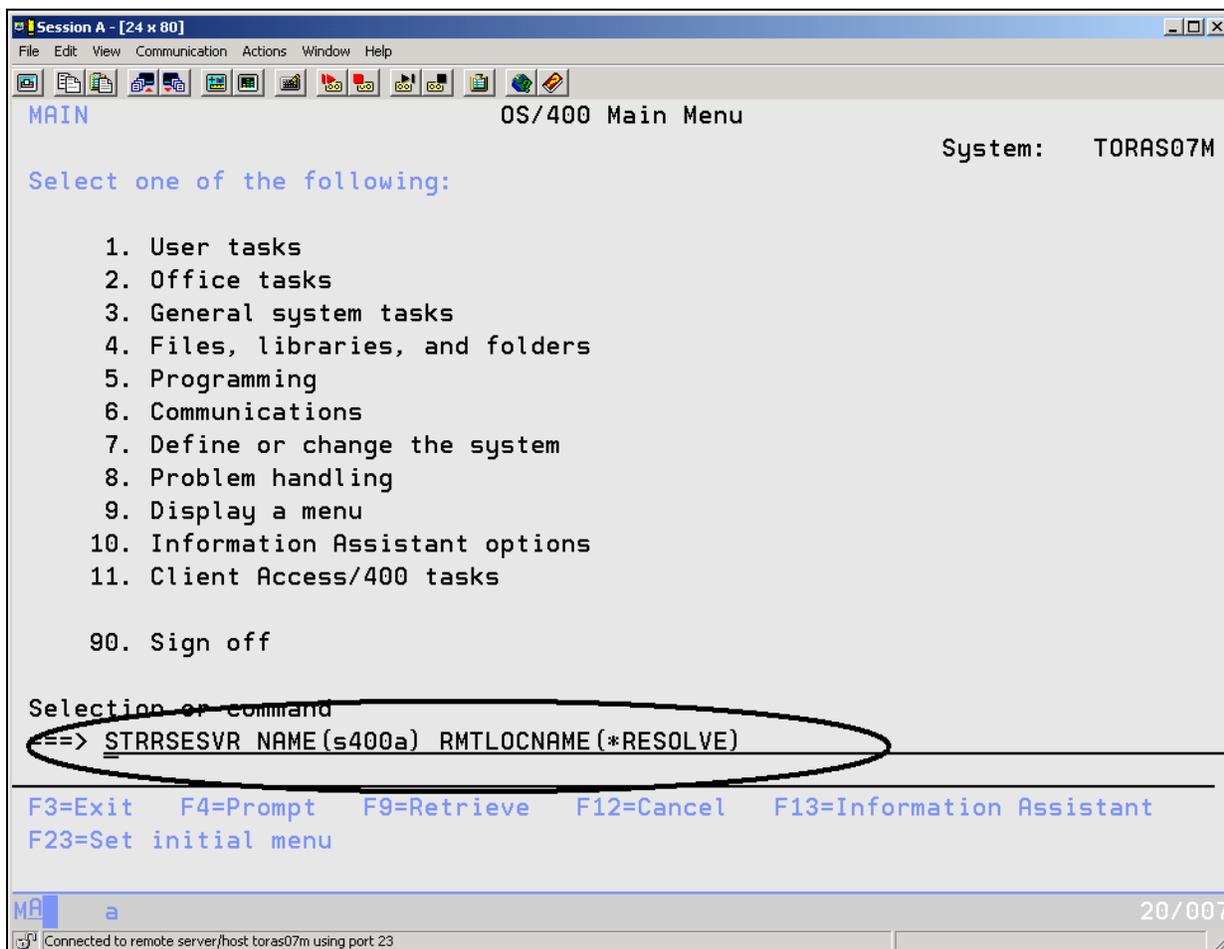


図 56: リモート・システム・エクスプローラーの対話式接続の開始

3. エミュレーターで次のコマンドを入力します。

```
STRRSESVR <connectionName> RMTLOCNAME(*RESOLVE)
```

パラメーター `connectionName` は、「リモート・システム」ビューで定義されている接続の名前です。これにより、対話式ジョブがリモート・システム・エクスプローラー通信サーバーに関連付けられます。**\*RESOLVE** キーワードを指定すると、ワークステーションの IP アドレスを取得できます。リモート・システム・エクスプローラー通信サーバーはこの情報を使用し

て、ワークステーションで実行されるリモート・システム・エクスプローラー・デーモンと通信します。

**注:** 接続名では、大文字小文字が区別されます。また、リモート・システム・エクスプローラーですでに定義されていなければなりません。

**図 57:** リモート・システム・エクスプローラーの接続画面に示されるような画面が表示されます。

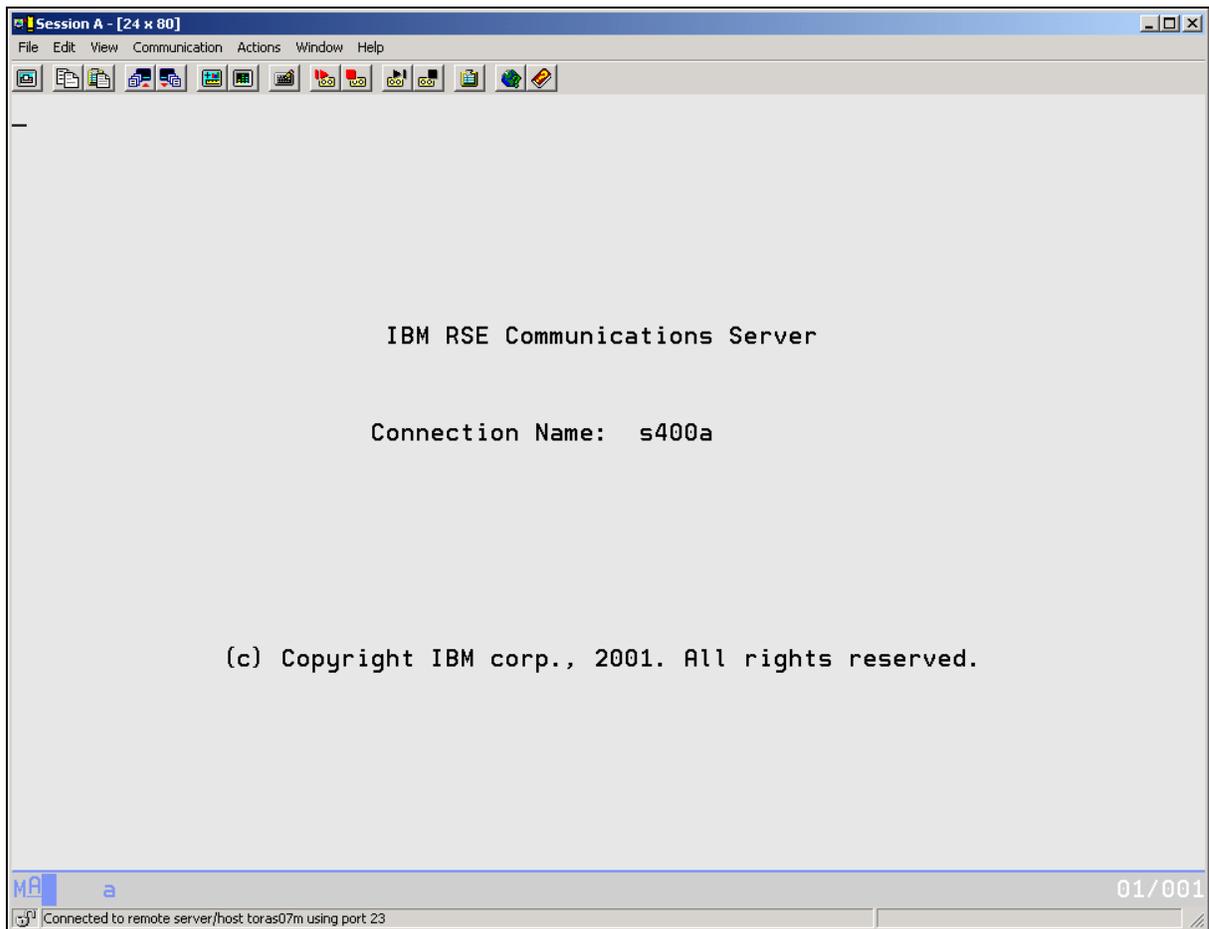


図 57: リモート・システム・エクスプローラーの接続画面

**注:** この画面はこの状態のままです。画面の終了を待つ必要はありません。このセッションは、リモート・システム・エクスプローラーから開始された対話式コマンドの対話式セッションです。

## PAYROLL プログラムの実行

**PAYROLL** プログラムをコンパイルしたので、実行する準備が整いました。

ワークベンチに戻り、「リモート・システム」ビューで、以下のようにします。

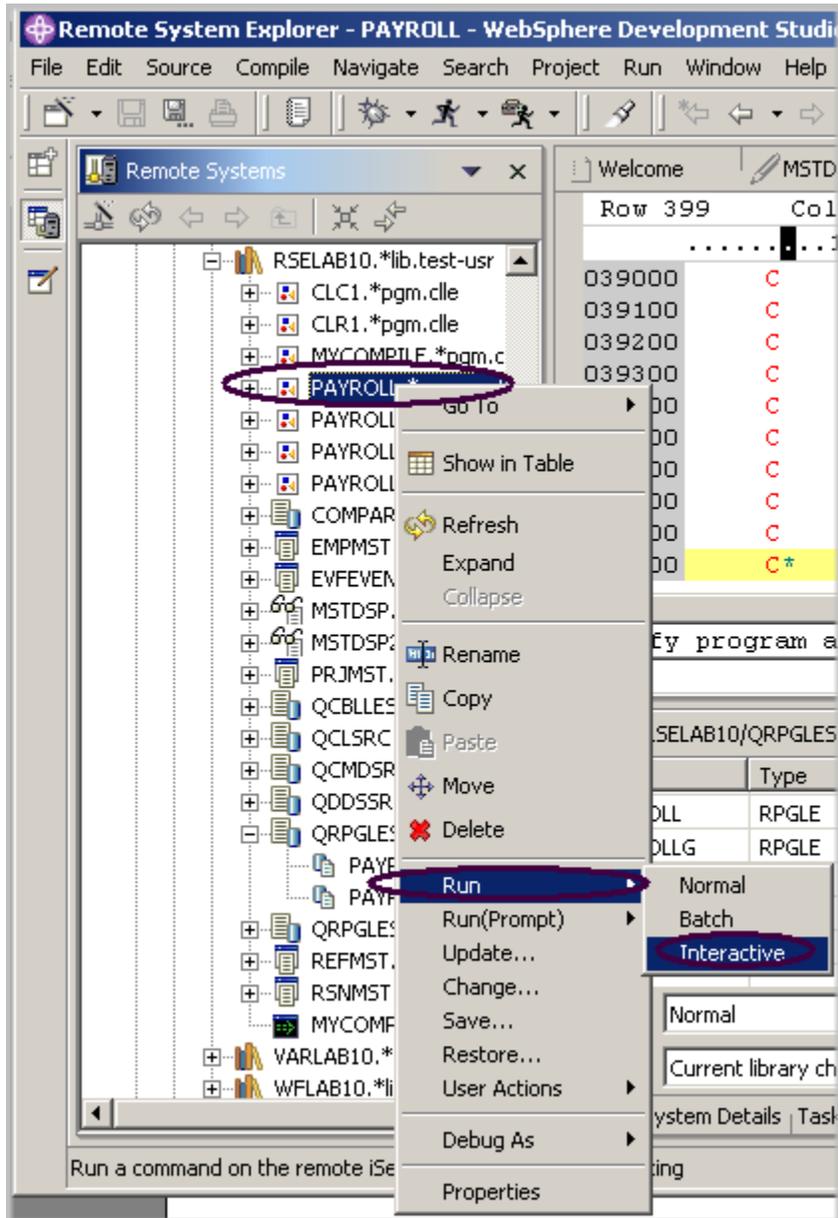


図 58: プログラムの対話式実行

1. 作成した **PAYROLL** プログラムを見つけます。
2. 「リモート・システム」ビューで「**PAYROLL**」プログラムを右クリックします。
3. ポップアップ・メニューから「実行」>「対話式」を選択します。
4. 5250 エミュレーション・セッションに切り替えます。

PAYROLL プログラムの開始メニューが表示されます。

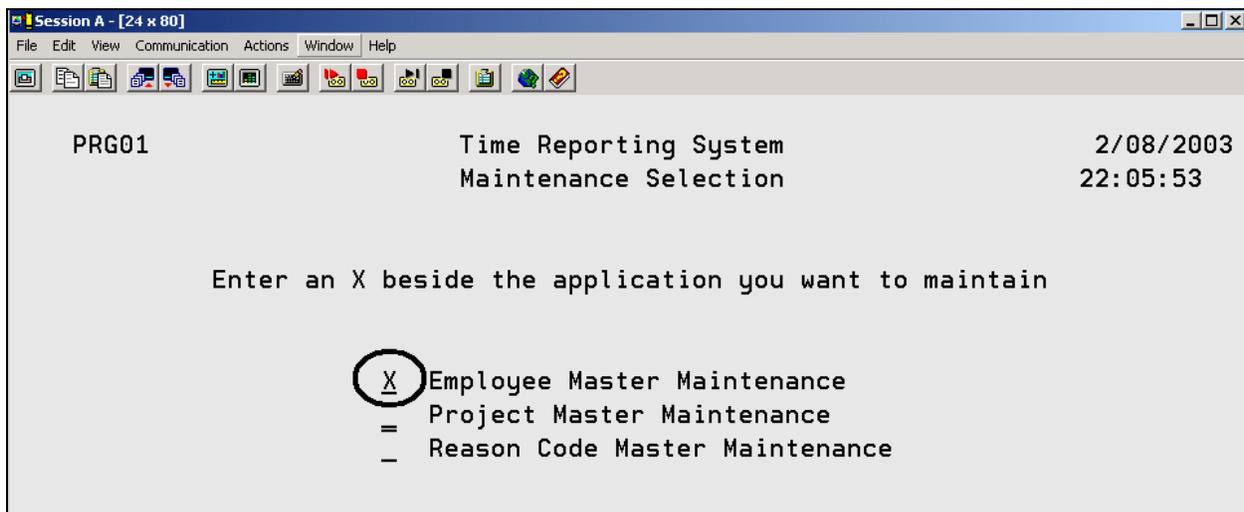


図 59: PAYROLL 開始メニュー

5. 「従業員マスター保守」の横に「x」を入れます。
6. **Enter** を押します。
7. 「従業員番号」に「123」と入力します。
8. 「アクション・コード」に「A」と入力して、従業員 123 を追加します。
9. **Enter** を押します。
10. その従業員について登録したい情報があれば入力します。
11. **Enter** を押します。
12. アプリケーションをいろいろと操作します。
13. **F3** を押して、PAYROLL プログラムを終了します。

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

### チェックポイント

1. \_\_\_\_\_ ツールはワークステーションで意味 (コンパイル) エラーを検査し、ユーザーが iSeries でクリーン・コンパイルを確実に行うことができます。
  - A. コンパイル
  - B. LPEX エディター
  - C. プログラム生成プログラム

- D. プログラム検査プログラム
2. \_\_\_\_\_ ビューは、それぞれのエラーをアイコンで識別します。アイコンは、エラーの重大度レベル、エラーの ID、メッセージ、重大度、ソース・メンバー内でエラーの原因となった行、エラーが発生したソース・メンバーの位置、および接続名を示します。
- A. リモート・システム
  - B. アウトライン
  - C. ナビゲーター
  - D. iSeries エラー・リスト
3. 「iSeries エラー・リスト」ビューでエントリーをソートできる基準:
- A. ID
  - B. メッセージ
  - C. 重大度
  - D. 行
  - E. 位置
  - F. 接続
  - G. 上記すべて
4. ローカル・プログラム検査プログラムを使用した場合は、ホストのコンパイルが正常に行われます (無駄な iSeries サイクルがありません)。 (T, F)
5. コンパイル・コマンドは、リモート・システム・エクスプローラーから iSeries サーバーで実行し、コンパイルからエラー・フィードバックを検索することができます。 (T, F)
6. リモート・システム・エクスプローラー内のそれぞれのプロファイルには、1 セットのソース・メンバー・タイプがあり、それぞれのソース・タイプには、それらと関連付けられた 1 セットのコンパイル名があります。 (T, F)
7. コンパイルの方法はいくつかあります。以下から選択しなさい。
- A. プロンプトあり
  - B. プロンプトなし
  - C. 対話式
  - D. バッチ
  - E. リモート・システム・エクスプローラーの通信サーバー
  - F. 上記すべて
8. 項目をソートし、多くのアクションを実行する機能があるため、iSeries テーブル・ビューには \_\_\_\_\_ ビューと同じ情報が表示されます。
- A. リモート・システム
  - B. アウトライン
  - C. ナビゲーター
  - D. iSeries エラー・リスト
9. iSeries テーブル・ビューからできること:
- A. ライブラリー、オブジェクト、およびメンバーをリストおよびソートする。
  - B. ポップアップ・メニューからビュー内の任意の項目をコピー、名前変更、削除、編集、コンパイル、および実行する。
  - C. システム間でファイルを転送する。
  - D. 上記すべて

10. 「リモート・システム」ビューまたは iSeries テーブル・ビューからプログラムおよびコマンドを実行できるジョブ:
  - A. リモート・システム・エクスプローラーの通信サーバー・ジョブ
  - B. バッチ・ジョブ
  - C. 対話式ジョブ
  - D. 上記すべて
11. \_\_\_\_\_ ビューは、それぞれのエラーをアイコンで識別します。アイコンは、エラーの重大度レベル、エラーの ID、メッセージ、重大度、ソース・メンバー内でエラーの原因となった行、エラーが発生したソース・メンバーの位置、および接続名を示します。
  - A. リモート・システム
  - B. アウトライン
  - C. ナビゲーター
  - D. iSeries エラー・リスト
12. 対話式プログラムには **5250** エミュレーターが必要です。そのため、このエミュレーターをリモート・システム・エクスプローラー通信サーバーの特定の接続と関連付けるには、最初に **STRRSESVR <connectionName>** コマンドを実行する必要があります。(T, F)

## 実習

この練習問題ではプログラム検査プログラムとコンパイル・コマンドの使用方法を学びました。現在、自分の iSeries システムに自分の RPG ソースがあります。そこで、以下の新規タスクを試してみます。コンパイル・コマンドでコンパイル (プロンプト) 作業を確認します。ソースを検査するときにソースでエラーが発生したと仮定して (エラーがない場合はソースにエラーを追加します)、エラー・リストからエディターにすべてのエラー・メッセージを挿入するように選択します。タスクの途中で困ったときは、**Development Studio Client for iSeries** のオンライン・ヘルプを使用してください。

## ここで学んだ内容

この練習問題では、「リモート・システム」ビューからプログラム検査プログラムを起動することによって、ソースを検査しました。次に、エラー・リストを検討し、いくつかのエラーを修正して、ソースを保管しました。さらに、「リモート・システム」ビューからリモート・コンパイルを実行しました。iSeries テーブル・ビューを使用して、コマンドを入力しました。最後に、iSeries への対話式接続を開始し、「リモート・システム」ビューからプログラムをすべて実行しました。

次の練習問題では、**CODE** 設計機能ツールを使用し、**DDS** ソースを視覚的に操作することで **RPG PAYROLL** プログラムの画面を設計します。



## 練習問題 6: 画面およびレポートの設計

この練習問題では、画面を追加するための表示装置ファイルを変更しながら **CODE** 設計機能のさまざまな面を理解します。

この練習問題を終了すると、以下のことができるようになります。

- **CODE** 設計機能のフィーチャーを説明する。
- **CODE** 設計機能を開始する。
- 表示装置ファイルをオープン、変更、および保管する。
- グループを追加する。
- レコードを追加する。
- さまざまな **DDS** フィールドを追加、削除、および変更する。
- **DDS** を検査する。
- **DDS** をコンパイルする。

エディターを使用して表示装置ファイルおよび印刷装置ファイルの **DDS** ソースを作成および保守する場合、タスクははかどらず、困難になる可能性があります。重要となるのは、画面やレポートを視覚的に設計し、**DDS** ソースを生成するためのグラフィック設計ツールです。それこそ、**CODE** 設計機能で実行できる作業です。

**CODE** 設計機能により、**DDS** 初心者プログラマーは **DDS** 言語の詳細について考えることなく、画面、レポート、およびデータベースを素早く容易に作成することができる一方で、**DDS** 熟練プログラマーは言語のフィーチャーと能力すべてにアクセスすることができます。これから、インターフェースの各部分をステップスルーして、**DDS** の一部も更新しましょう。

ワークベンチのリモート・システム・エクスプローラー・パースペクティブでは、前の練習問題で使用した接続を使用します。

### 「リモート・システム」ビューでの **DDS** メンバーのオープン

「リモート・システム」ビューで **DDS** ファイル・メンバーをオープンするには、以下のようになります。

1. 「ライブラリー・リスト」フィルターを拡張するか、すでに拡張されている場合はそのまま使用します。
2. 「**RSELABxx**」ライブラリーの「**QDDSSRC**」ファイルを拡張します。
3. 「**MSTDSP**」メンバーを右クリックします。
4. 「アプリケーションからオープン」>「**CODE** 設計機能」を選択します。

メンバー **MSTDSP** がワークステーションにダウンロードされ、CODE 設計機能にロードされます。

## DDS ツリーの表示

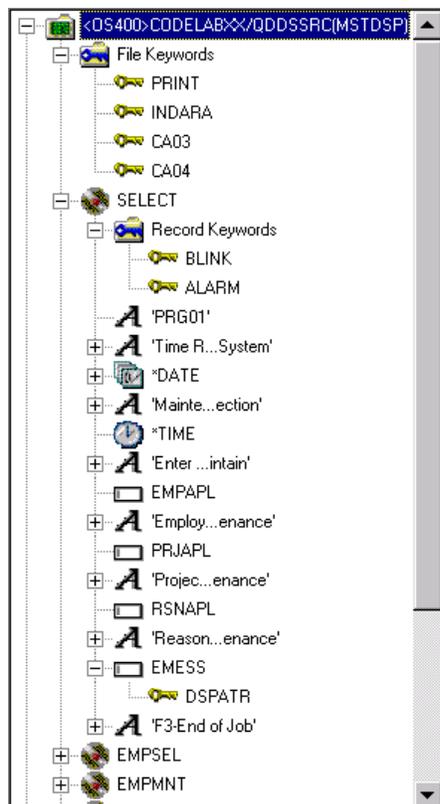
基本的に、この時点では DDS のエクスプローラー・ビューが表示されています。設計機能の左側にある DDS ツリー・ビューには、DDS ソースがファイル、レコード、フィールド、およびキーワードの階層で表示されます。ありふれた直感的な方法で、DDS ソースの全体の構造を参照し、素早くナビゲートできます。必要事項についてはすべて説明しますので、DDS のエキスパートでないユーザーでも心配する必要はありません。

DDS ツリーは CODE 設計機能ウィンドウの左側にあります。このビューには、ロードされた DDS ソースがツリー構造で表示されます。選択された DDS オブジェクトごとに、ファイル、レコード、フィールド、ヘルプ仕様、キー、およびキーワードの階層が示されます。DDS ツリーには、レコードのグループが表示されます。このグループは、ツリー階層内でファイルの仲間として、設計している画面またはレポートを表します。

このビューでは、グループの作成、およびキー、キーワード、フィールド、レコードのコピーまたは移動を行うことができます。DDS オブジェクトにエラーがある場合、DDS オブジェクトを表すアイコンに赤い X 印が表示されます。

DDS ツリーでの選択をオープンしてファイル・レベル・キーワードおよびレコード **SELECT** を参照するには、以下のようにします。

1. フォルダー「<Servername>RSELABxx/QDDSSRC(MSTDSP)」の横のプラス符号 (+) をクリックします。
2. フォルダー「ファイル・キーワード」の横のプラス符号 (+) をクリックします。
3. レコード「**SELECT**」の横のプラス符号 (+) をクリックします。
4. フォルダー「レコード・キーワード」の横のプラス符号 (+) をクリックします。
5. フィールド「**EMESS**」の横のプラス符号 (+) をクリックします。



DDS ツリーにファイル・レベル・キーワードおよびレコード「SELECT」の要約が表示されます。

## DDS オブジェクトの選択

設計機能の右上には、複数の異なるタブ付きページを持つワークブックがあります。ワークブックとは、表示装置ファイルおよび印刷装置ファイルを設計する CODE 設計機能の領域のことです。このノートブックは CODE 設計機能ウィンドウの右上に表示できます。先頭ページ「詳細」には、DDS ツリーによって選択された DDS オブジェクトの詳細なビューが表示されます。このページは、詳細モードとリスト・モードのどちらでも表示できます。

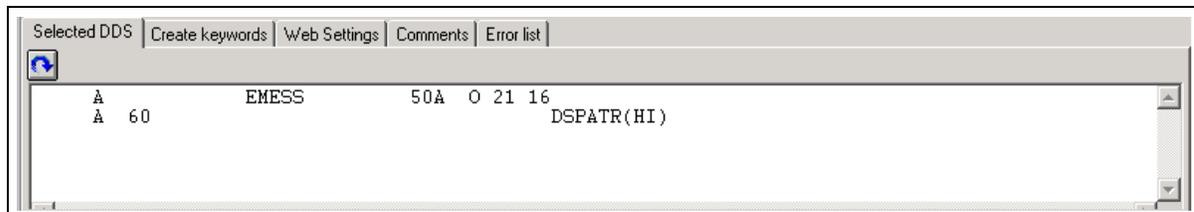
*詳細モード*では、DDS オブジェクトごとの情報が列に表示されます。DDS オブジェクト (レコード、ヘルプ仕様、フィールド、キーワード、またはキー) ごとに詳細を知りたい場合は、このモードを使用します。

*リスト・モード*では、DDS オブジェクトが名前付きアイコンの列として表示されます。ページ内でより多くの DDS オブジェクトを参照したい場合、このモードを使用します。

設計機能の右下にはユーティリティー・ノートブックがあります。これを使用すれば、エラーや選択オブジェクトなどの DDS ソースの特定エレメントを表示できます。このノートブックには、



3. DDS ツリーでフィールド「**EMESS**」をクリックします。「詳細」ページに、フィールド・レ



ベル・キーワードが表示されます。「選択された DDS」ページに、**EMESS** フィールドの DDS が表示されます。

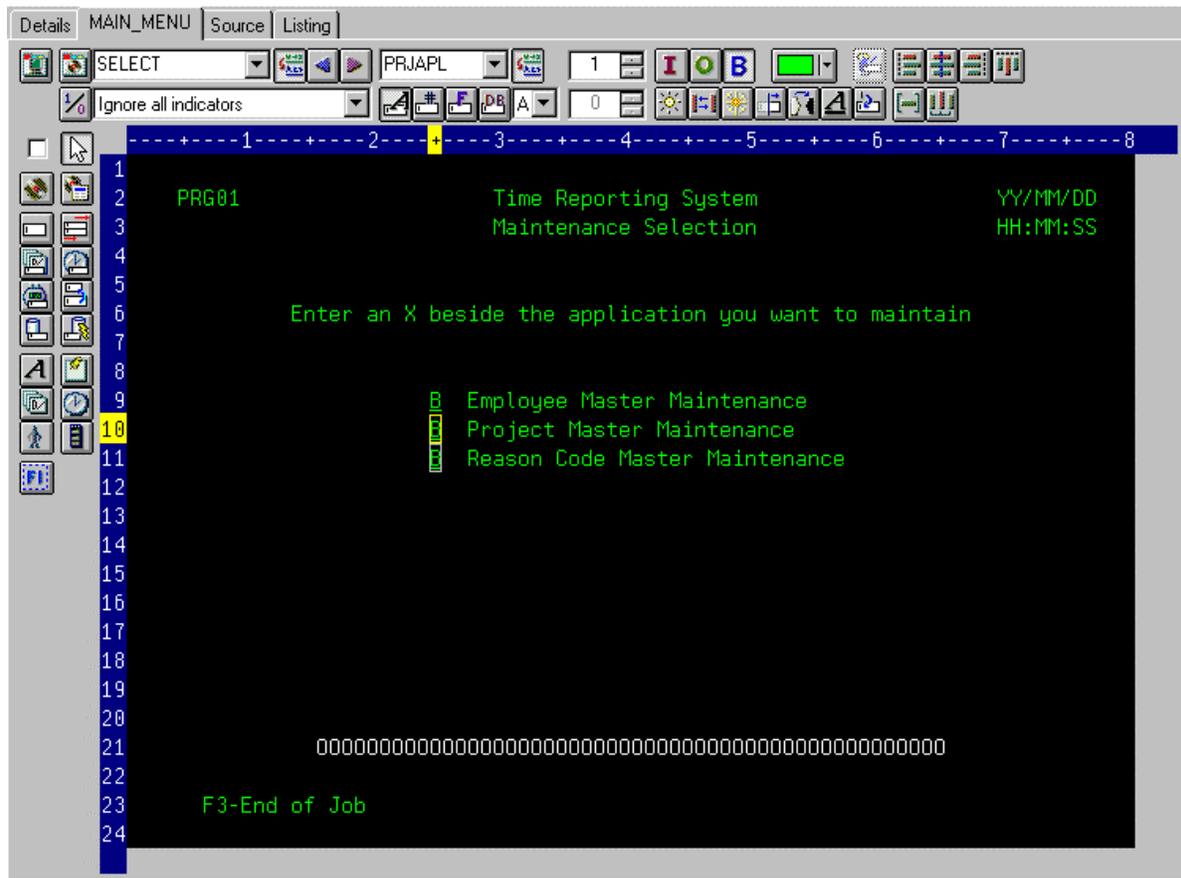
この比較的小さくて単純な DDS ソース・メンバーであっても、設計機能を使用すると簡単に DDS ソースをナビゲートできることがわかります。設計機能は、構文が直感的でグラフィカルな方法で説明されるため、DDS を学ぶには理想的なツールです。ただし、生産性を飛躍的に向上させるには、表示通りに画面やレポートを操作する必要があります。その場合でも、その操作に必要な DDS の存在を少しでも意識させてはなりません。それには、設計ページが必要です。

## DDS 画面の設計

ここでは、多くの時間を使用して、設計ページで DDS 画面とレポートを作成、更新、および設計します。設計ページでは、直感的なグラフィカル・ユーザー・インターフェースを使用して、画面とレポートを視覚的に設計することができます。設計ページには、DDS ソースが、画面 (表示装置ファイル用) または印刷ページ (印刷装置ファイル用) に表示されるように表示されます。これにより、グラフィカル・ユーザー・インターフェースでレコードやフィールドをレイアウトすることによって、アプリケーションの画面またはレポートを設計できます。

設計ページでは、グラフィカルな DDS オブジェクトを容易に作成、編集、サイズ変更、および移動できます。設計域の左にあるパレット・プッシュ・ボタンを使用して、またはポップアップ・メニューから、設計ページ上で新規のレコード、フィールド、および固定情報を直接作成します。設計域の上にあるツールバーを使用すれば、現在選択されているオブジェクトに関する情報だけでなく、多くの編集フィーチャーにも素早くアクセスできます。

ワークブックでタブ「**MAIN\_MENU**」をクリックします。



「MAIN\_MENU」が何によって構成されているのかを理解するために、グループの概念を説明する必要があります。グループとは、画面またはレポートが実行時にどのように組み立てられるのかを表す1つ以上のDDSレコードの単なる集まりです。グループを使用することにより、実行時に画面とレポートがプログラムによって組み立てられるときの表示通りに開発時に画面とレポートを操作することができます。CODE設計機能でグループを操作するには、画面を構成するレコード・フォーマットをCODE設計機能で指定する必要があります。今回は、設計ページに表示される画面についてこれを行いました。グループMAIN\_Menuが作成され、CODE設計機能はこの情報をDDSソースのコメント行に保管しました。作成したグループは、以降のCODE設計機能セッションで再利用できるようにDDSにコメント行として保持されます。

作成したグループは、ツリー・ビューにも、設計ページとしてワークブックにも、定義されたグループごとに表示されます。これにより、レコードの各グループに素早くアクセスできます。

## 既存レコードからのグループの作成

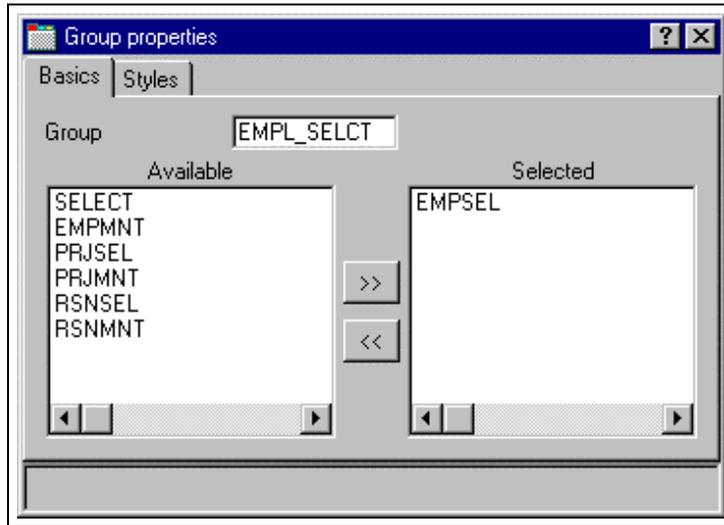
既存の DDS を操作している場合、レコードが使用されている方法に対応するグループを作成する必要があります。この例では、次の画面用のグループを作成します。ユーザーは、**payroll** データベースに維持する従業員をその画面で選択します。画面はレコード・フォーマット **EMPSEL** で構成されます。

新規グループを作成するには、以下のようになります。

1. **DDS ツリー**の下部へスクロールし、「**MAIN\_MENU**」グループの横のプラス符号 (+) をクリックします。このグループの唯一のレコードとして **SELECT** レコードが表示されます。



2. 「**MAIN\_MENU**」グループを右クリックします。
3. ポップアップ・メニューから「**グループの挿入**」を選択します。「グループ特性」ノートブックが表示され、グループ **SCREEN1** のブランク設計ページも表示されます。  
プロパティ・ノートブックでは、現在選択されている DDS オブジェクトのプロパティを表示および更新できます。このノートブックは、**CODE** 設計機能の任意のビュー、ポップアップ・メニュー、またはメニュー・バーからオープンできます。プロパティ・ノートブックには、モードはありません。オブジェクトのプロパティを変更すると、選択されたオブジェクトはすぐに変更されます。
4. 「グループ特性」ノートブックの「**使用可能**」リスト・ボックスでレコード「」をクリックし、 ボタンを押します。わかりやすくするために、ここではこのレコードのみを追加します。設計ページに、レコード「」の内容が表示されます。
5. 「**SCREEN1**」を「**EMPL\_SELCT**」で上書き入力して、グループに名前を付けます。



6. 右上隅の「X」をクリックして、「グループ特性」ノートブックをクローズします。

グループの作成が終了しました。ここでは、このグループに含まれるレコード・フォーマットを設計ページで操作することができました。次に、新規レコード・フォーマットを作成します。

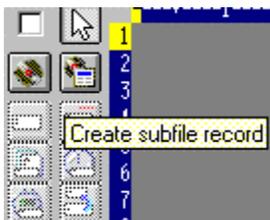
これは、不便なアプリケーションの1つと考えられています。理由は、ユーザーがデータベースの内容を参照できないため、事前に従業員番号を知らなければならないためです。必要なのはサブファイルです。しかし、コーディングするのが難しいとお考えですか? CODE 設計機能を使用すれば簡単です。

## 新規画面の作成

設計ページで新規レコード画面を作成するには、操作可能な空のページを作成するグループを作る必要があります。新規グループを作成するには、以下のようになります。

1. DDS ツリー内の新規の「EMPL\_SELCT」グループを右クリックします。
2. ポップアップ・メニューから「グループの挿入」を選択します。「グループ特性」ノートブックが表示され、グループ **SCREEN1** のブランク設計ページも表示されます。
3. グループ名を「EMPL\_LIST」に変更し、「グループ特性」ノートブックをクローズします。
4. 左側のパレットから適切なツールを選択し、項目を作成したい設計ページの場所をクリックすることにより、設計ページで項目を作成できます。現在、フィールドを作成するレコードがないため、パレット内の多くの項目が使用不可になっています。「標準レコードの作成」および「サブファイル・レコードの作成」の2つのツールのみが使用可能です。マウスをボタンの上

にしばらく置くと、そのボタンを説明する吹き出しヘルプが表示されます。



5. 「サブファイル・レコードの作成」ボタンをクリックし、濃い灰色の領域内をクリックします。サブファイルおよびサブファイル制御レコードのペアが作成されます。

### サブファイル・レコードへのフィールドの追加

ここでは、設計ページを使用していくつかの桁をサブファイルに追加します。サブファイルは 8 行目に位置します。マウス・カーソルを使用して、画面に置きたいパーツ (この場合、サブファイル) の位置を指定することができます。

1. ここでは、「名前付きフィールドの作成」ボタン  をクリックして、**8 行目**の任意の場所をクリックします。6 つのフィールドが垂直の桁で表示されます。これは、作成されたサブファイルが現在 6 の SFLPAGE (表示可能なリスト・サイズ) を指定したためです。
2. 先頭のフィールドをクリックして、マウス・ボタンを押したまま **8 行目の 5 桁目に移動しま**



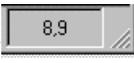
す。現在の行と桁は、移動したフィールドの真上に表示されます。

- マウスをフィールドの**右端**に移動します。マウスが両方向矢印に変わります。  
マウス・ボタンを押したまま**左に移動**します。フィールドのサイズが小さくなります。現在のサイズはフィールドの真上に表示されます。サイズが**3**のときにマウスを**放します**。
- 設計ページの上部にあるツールバーは、現在選択されているパーツのモニターおよび操作に非常に便利です。



- ドロップダウン・リスト・ボックスのテキストを上書きして、レコード名を「RECORD1」から「EMPLSTSFL」に、フィールド名を「FIELD1」から「OPCODE」に変更します。



- Color palette** ボタンをクリックし、ピンクを選択してフィールドの色を変更します。
- I** ボタンをクリックし、フィールドの使用法を入力に変更します。  
ここでは追加の桁をサブファイル内に作成します。
- カーソルを**8行目の9桁目**に置きます。設計機能フレームの右下に、現行カーソル位置が表示されます 。画面上にカーソル位置のフィールドが表示されない場合、右上隅にある**最大化**ボタンをクリックします。カーソルを移動するには、カーソル・キーまたはマウスを使用します。
- 決まった長さを持つ長いフィールドを作成する場合、**SDA** 構文を使用するとより簡単に作成できます。次のように入力します。

+0(30)

次に、**左矢印 (バックスペースではありません)** を押して、作成したテキストを選択します。

「選択された DDS」 ページで「+O(30)」を含むテキスト固定情報を作成したことに注意してください。

10. ツールバー上の「フィールドへのストリングの変換」ボタン  をクリックするか、または **F11** を押して、SDA 構文を長さ 30 の文字出力フィールドに変換します。
11. ツールバーを使用して新規フィールド名を「ENAME」に変更します。これにより、従業員の名前が表示されます。
12. カーソルを **8 行目の 41 桁目** に置きます。
13. 従業員の給与のフィールドを追加します。ここで、設計機能に数値の外観を指定し、隠れた EDTCDE をすべて設計機能に生成させることができると便利ではないでしょうか。タイプ  
\$666,666.66  
次に、左矢印を押します。
14. **F11** を押して、このフィールドを、コンマ区切り文字、小数点以下 2 桁、通貨記号、符号なしの出力数値フィールドに変換します。「選択された DDS」ページで、生成された内容を確認します。いかがですか。
15. ツールバーを使用して、フィールド名を「SALARY」に変更し、フィールドの色を黄色にします。
16. サブファイルは、左に詰められているように見えます。横にスペースを広げたほうがよいでしょう。フィールドを選択し、ツールバーの右端にある  をクリックします。その他の配置ボタンはフィールドを左、右、中央、および上に位置合わせされます。
17. パレットの下に 3 つのスピン・ボタンがあります。1 番上 (サブファイル・サイズ) は、アプリケーションによって入力されるリスト内のエントリーの総数を指定します。2 番目 (サブファイル・ページ・サイズ) は、画面に表示されるエントリーの数です。サブファイル・サイズを (上書きで) 300 に設定し、サブファイル・ページ・サイズを 9 に設定します。それに合わせて設計ページが更新されます。

## 複数レコードの切り替え

ここで、サブファイル制御レコードを修正しましょう。作成したグループには 2 つのレコードが含まれています。これは、ツールバーのレコード・ドロップダウン・リスト・ボックスをドロップダ

ウンすることで確認できます。



1. ドロップダウン・リスト・ボックスから「RECORD1CTL」を選択して現行レコードを変更するか、 をクリックするか、または **Alt+End** を押します。列見出しを並べることができるようにサブファイルのフィールドが表示されていますが、現行レコードのフィールドと区別できるように半分の輝度で表示されています。
2. ツールバーを使用してレコード名を「EMPLSTCTL」に変更します。

サブファイル制御ヘッダーに「position to」エントリーを指定しましょう。

3. カーソルを **4 行目の 9 桁目** に置き、次のように入力します。  
Position to:
4. ここで、従業員名フィールドが必要となります。サブファイルで行ったように名前つきフィールドを適切な特性で作成したり、パレットの  ボタンを使用してソース参照を作成したり、 ボタンのいずれかを使用して元のデータベース・フィールドを参照したりすることができました。しかし、さらに簡単な方法があります。コピーおよび貼り付けを使用します。DDS ツリーで「EMPMENT」レコードを拡張します (プラス符号 (+) をクリックします)。
5. 「ENAME」フィールドをクリックして、**Ctrl+C** を押します。(ポップアップ・メニューまたは「編集」プルダウンにも、「コピー」メニュー項目があります)。
6. カーソルを **4 行目の 23 桁目** に置き、**Ctrl+V** を押します。このように、簡単に操作することができます。
7. フィールド名を「POS\_TO」に変更します。

## フィールド・エラー処理の追加

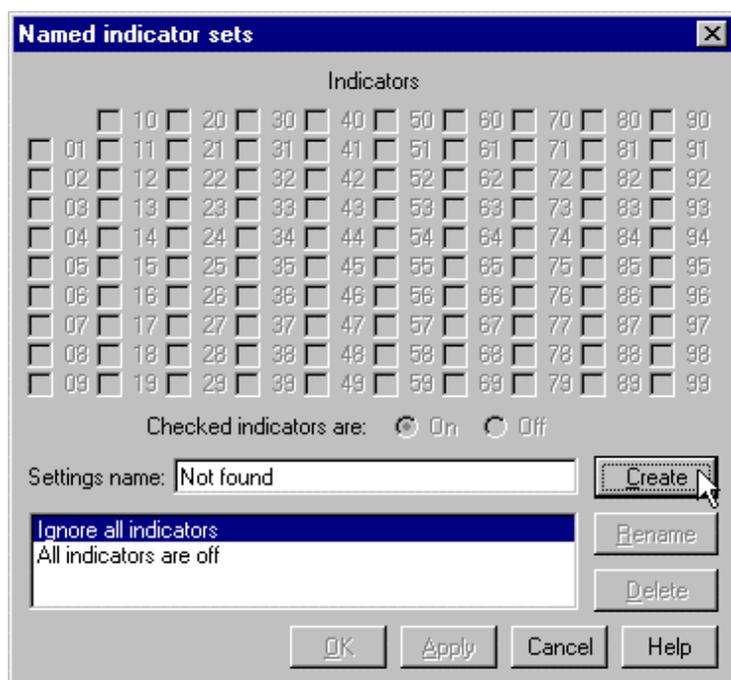
「position to employee name」フィールドのエラー処理を行いましょう。データベースに従業員名が見つからない場合、プログラムは標識 60 をオンにします。標識とは、処理中にイベントが発生したかどうかをシグナル通知する 2 桁の変数 (01, 02, 03, ... 98, 99) です。標識を使用することによ

り、プログラム・イベントに画面表示を制御させることができます。この標識はプログラムのある部分で「オン」または「オフ」に設定されていて、その状況はプログラムの別の部分で参照されます。標識を「オン」または「オフ」に設定することにより、フィールドを標識に関連付けて、画面にフィールドを表示させるかどうかを制御することができます。

画面では、フィールドが赤色になり、反転表示され、カーソルがそのフィールドに置かれます。ここで、より高レベルで、1 から 99 までの任意の数値を覚えるより簡単な作業を行うことができる便利ではないでしょうか。

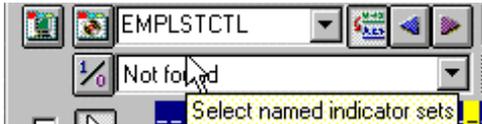
標識を設定するには、以下のようにします。

1. 設計ページのツールバーにある  ボタンをクリックします (または **F7** を押します)。「名前付き標識セット」ウィンドウが表示されます。
2. 「設定名」フィールドに「Not Found」と入力します。
3. 「作成」をクリックします。



4. 「60」の横のチェック・ボックスにチェックマークを入れて、「OK」をクリックします。  
「Not found」標識セットが有効になります。

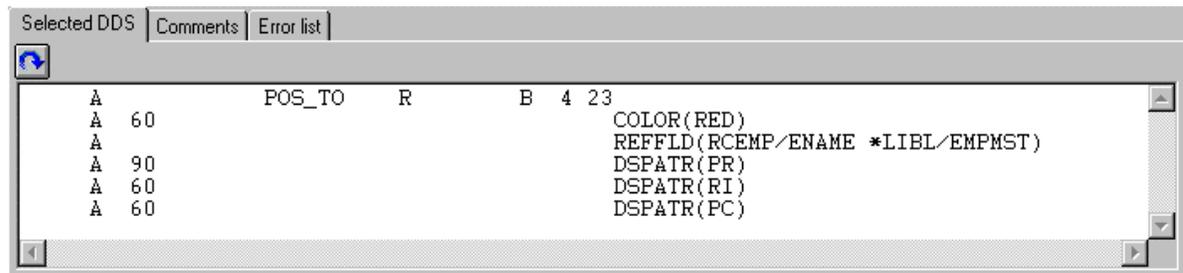
設計域は、標識 60 がオンであり、他のすべての標識がオフであるかのように表示されます。設計ページ・ツールバーの左下のドロップダウン・リスト・ボックスには、現在の標識セットが表示されます。



5. ここで、「**POS\_TO**」フィールドを選択します。
6. ツールバーで**赤色**を選択し、表示属性に**反転表示**および**カーソル配置**を選択します。  
(現在の表示属性を表すツールバー・ボタンのセットは、色ボタンの下にあります)。ツールバーは次のようになります。



7. 「**選択された DDS**」 ページで生成された DDS を検査します。

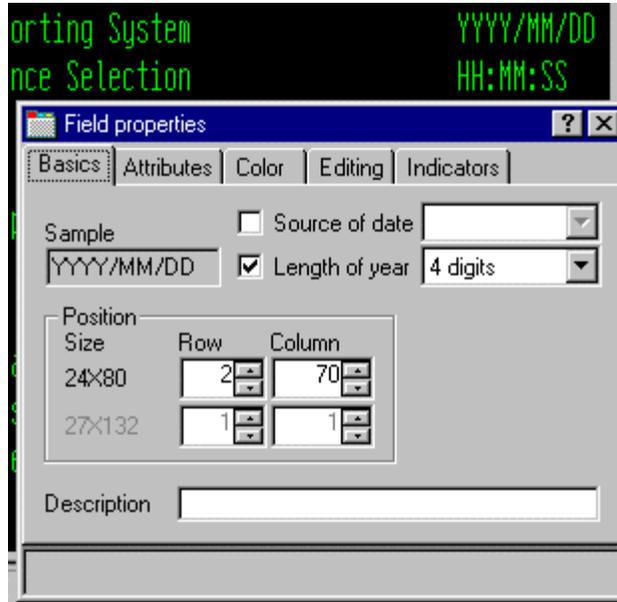


すべての新規キーワードが条件 60 で作成されたことに注意してください。(当初、このフィールドを使用して DSPATR(PR) が貼り付けられました)。

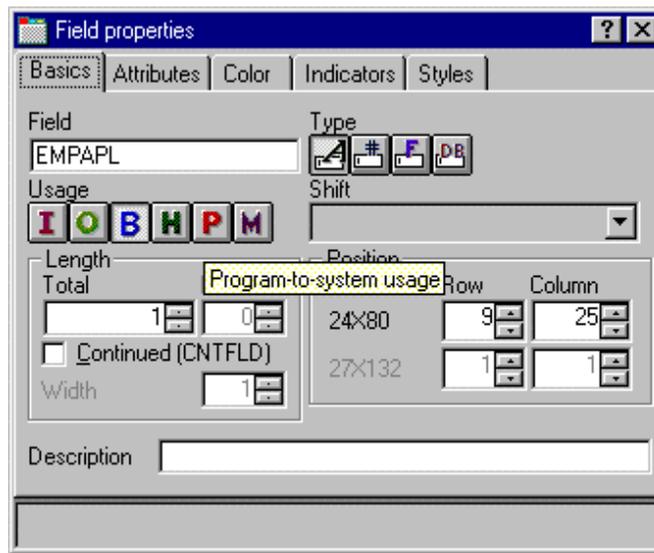
8. では、実際に試してみましよう。「名前付き標識セットの選択」ドロップダウン・リスト・ボックスから、「**すべての標識がオフ**」を選択します。



4. ドロップダウン・リスト・ボックスから「4桁」を選択します。プロパティ・ノートブックでサンプルがどのように更新されるかに注意してください。



5. 設計ページをテストするために、ワークブック内の「MAIN\_MENU」タブをクリックして、画面の右上隅を参照します。日付が4桁に変更されています。
6. ここで、SELECT レコード内の「EMPAPL」フィールドをクリックします。「フィールド・プロパティ」ノートブックで「基本」タブをクリックします。このページでは、フィールドの名前、使用法、長さ、タイプ、画面位置を変更できます。その他のページからは、このフィールドの他のプロパティに素早くアクセスすることができます。



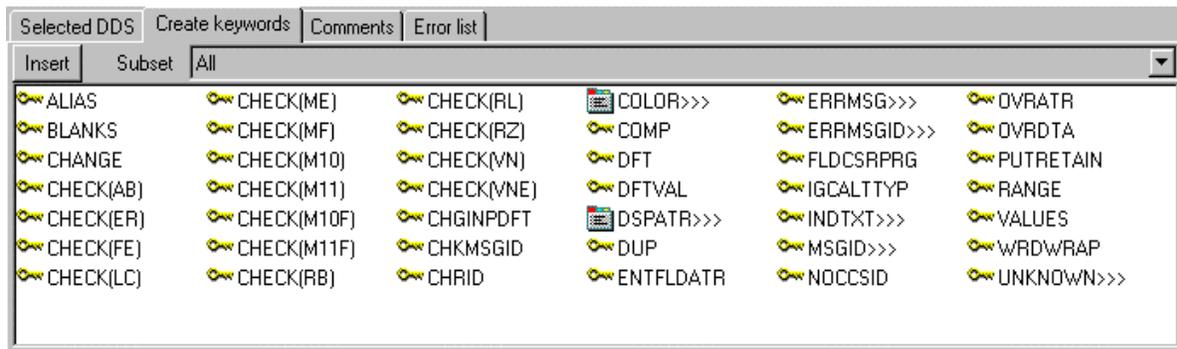
## 新規キーワードの追加

これまでに、ディスプレイやレポートなどの視覚的な面を管理する場合に **CODE** 設計機能が役立つ場面がわかりました。しかし、本当に行うべき作業があります。**DDS** のすべての機能へのアクセスが必要です。キーワードにアクセスする必要があります。

キーワードを追加するには、以下のようにします。

1. **DDS** ツリーで「**EMPAPL**」フィールドをクリックします。
2. **F5** を押すか、ポップアップ・メニューから「**キーワードの挿入**」を選択します。

「**EMPAPL**」フィールドの「詳細」ページが表示され、ユーティリティー・ノートブックで「**キーワードの作成**」タブが選択されます。このページでは、選択されたファイル、レコード、またはフィールドに許可されたキーワードのサブセットが表示され、フィールドのタイプ、使用法、シフト、およびそのフィールドが属するレコードが考慮されます。オプションの内容を正確に知ることは非常に役立ちます。この情報は解説書で素早く確認できるものではありません。



3. 「基本」ページの **EMPAPL** プロパティ・ノートブックを使用し、 ボタンをクリックしてフィールドを数値型に変更します。「キーワードの作成」ページのキーワード・リストが変更されました。
4.  ボタンをクリックして、フィールドを英数字に戻します。「キーワードの作成」ページのキーワード・リストが変更されました。
5. 「**ALIAS**」キーワードをクリックして、**F1** を押します。  
「**ALIAS**」キーワードの **DDS** 参照ヘルプが表示されます。**CODE** 設計機能には多くのオンライン・ヘルプがあります。項目、アイコン、またはノートブックのヘルプを参照したい場所で **F1** を押してください。現在試みている操作に関するヘルプが表示されます。「ヘルプ」メニューからは、他の有用な情報ソースだけでなく、**DDS** 言語解説書にも素早くアクセスできます。
6. 「ヘルプ」ウィンドウを最小化します。
7. 「**INDTXT**」キーワードをダブルクリックします。(キーワードを見つけるには、右にスクロールしなければならない場合があります)。変更可能なデフォルト値を持つキーワードが作成されます。
8. もう一度「**INDTXT**」キーワードをダブルクリックします。競合する同じデフォルト値を持つキーワードが作成されます。

## ソースの検査

**DDS** ソースに 1 つの新規レコードといくつかの新規フィールドを追加しました。**CODE** 設計機能が **DDS** ソースに追加するものはすべて、必ず正しい構文を持ちます。ここでは、意味エラーがないことを確認する必要があります。前のセクションで、同じ標識を記述する 2 つの **INDTXT** キーワ

ードを作成し、エラーを発生させました。

ソースを検査するには、以下のようにします。

1. 「ツール」メニューから「ファイルの検査」を選択します (またはメイン・ツールバーの  ボタンをクリックします)。CODE エディターまたは LPEX エディターが使用する検査プログラムと同じ検査プログラムを使用して、DDS ソースが検査されます。設計機能の下部にある状況表示行にメッセージが表示され、検査処理がエラーを発生させて完了したことが示されます。

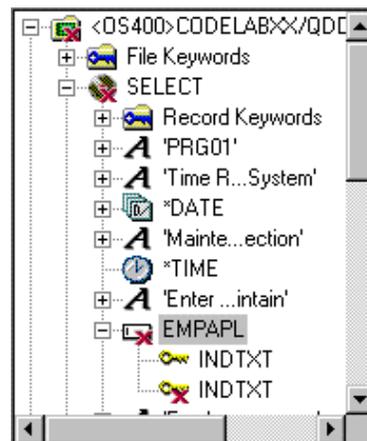
2. DDS ツリーには、問題に通じる赤い x 印の証跡が表示されます。ファイル・アイコン、「SELECT」レコード、

「EMPAPL」フィールド、最後に 2 番目の「INDTXT」キーワードに赤い x 印が付いています。

3. ワークブックの「MAIN\_MENU」タブをクリックします。

「EMPAPL」フィールドが赤く強調表示されています。

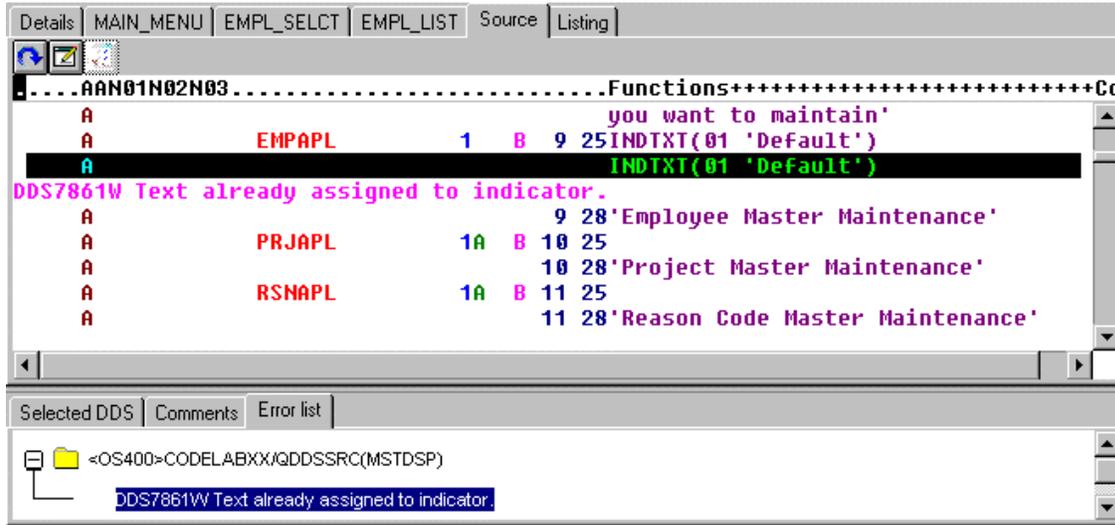
4. ワークブックの「リスト」タブをクリックします。このページには、最も新しいプログラム検査によって生成されたリストが表示されます。警告メッセージはリストのどこかに埋め込まれていますが、容易には見つけることができません。



5. 問題がある場合、問題はユーティリティー・ノートブックの「エラー・リスト」ページに表示されます。このページは、CODE エディターまたは LPEX エディターのエラー・リストとまったく同じように機能します。「エラー・リスト」タブをクリックします。

6. 「エラー・リスト」内の警告「DDS7861」をダブルクリックします。

(F1 を押して、メッセージに関する詳細ヘルプを参照します)。「ソース」ページが表示され、ソース内でエラーがある場所にカーソルが置かれます。「ソース」ページとは、DDS ソースの現在の状況を示すトークン化された読み取り専用ビューです。読み取り専用のため、編集はできません。しかし、ここでエラーを消去できると便利ではないでしょうか。一部の作業はエディターで行う方が明らかに速く実行できますが、多くの作業はビジュアル環境で行う方が速く実行できます。ボタンを押すことで 2 つのモードを切り替えることができるのは、非常に便利です。実際にやってみましょう。



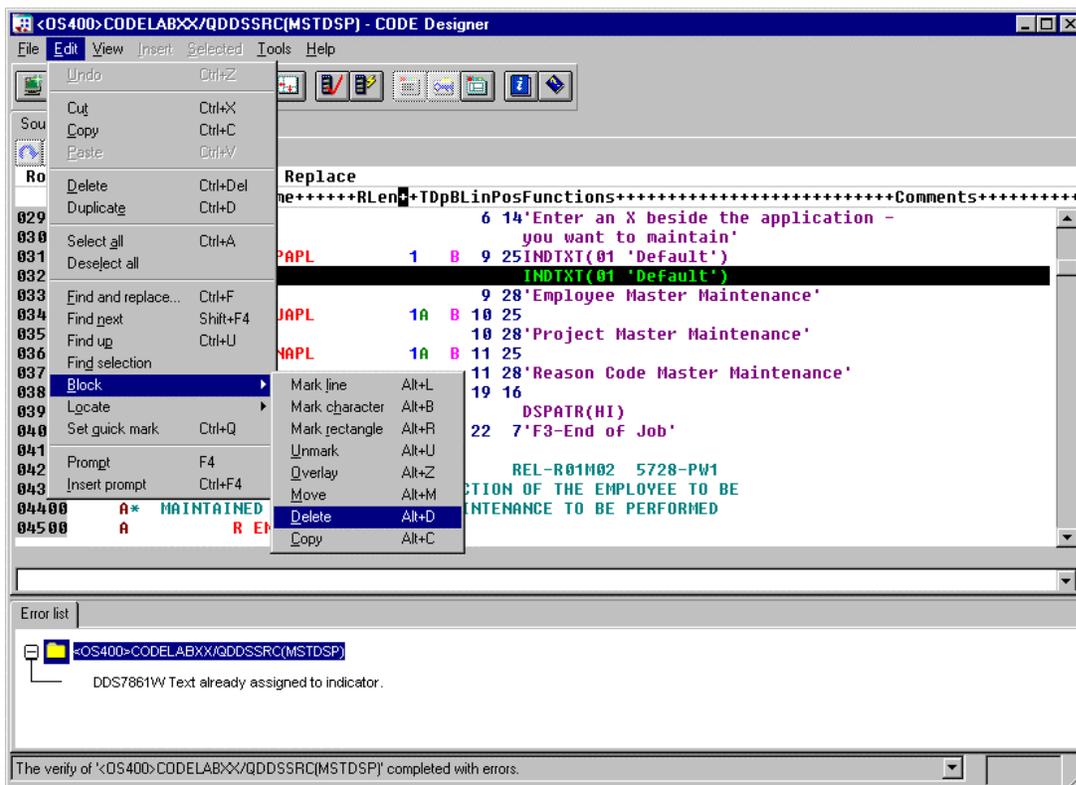
## 画面設計と画面編集の切り替え

設計モードと編集モードを切り替えるには、以下のようにします。

-  ボタンをクリックするか、または「編集」メニューから「ファイル」->「DOS ソースの編集」を選択します。ここでは、エディターのすべての機能にアクセスできます。
- 「編集」および「表示」プルダウンを探索します。
- Ctrl-F** を押して「検索および置換」ウィンドウを表示します。
- 「INDTXT」と入力し、「検索」をクリックします。



- Ctrl-N** を押して次のオカレンスを検索します。
- 2 番目の INDTXT 行を削除します。



## ソースのコンパイル

ここでは、リモート・システム LPEX エディターで行ったように、iSeries 上でソースをコンパイルします。

1. 「ファイル」メニューから「保管」を選択し、ソースを iSeries に保管します。
2. 「ツール」メニューから「コンパイル」を選択して、「プロンプトなし」を選びます (またはメイン・ツールバーの  ボタンをクリックします。)
3. コンパイルが完了すると、メッセージが表示されます。「メッセージ」ウィンドウで「OK」をクリックします。

PAYROLL プログラムを実行すると、プログラムの開始画面に行った変更の 4 桁の年が表示されます。

## CODE 設計機能のクローズ

「ファイル」メニューから「終了」を選択します。

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

- 画面やレポートを視覚的に設計し、DDS ソースを生成するためのグラフィック設計ツールがあります。このツールの名前は何ですか。
  - Page Designer
  - CODE 設計機能
  - 画面設計機能
  - 上記のいずれでもない
- DDS ツリー・ビューに表示されるもの:
  - ツリー構造としてロードされた DDS ソース
  - 選択された DDS オブジェクトごとのファイル、レコード、フィールド、ヘルプ仕様、キー、およびキーワードの階層
  - レコードのグループ (ツリー階層内でファイルの仲間として、設計している画面またはレポートを表す)
  - 上記すべて
- 以下の項目とその正しい記述を結び付けなさい。
  - ワークブック
  - 「詳細」ページ
  - ユーティリティー・ノートブック
  - 設計ページ
  - プロパティ・ノートブック
  - グループ
  - 標識
  - エラーや選択オブジェクトなどの DDS ソースの特定エレメントを表示する。
  - DDS ツリーで現在選択されている項目の内容をリストする。
  - 複数の異なるタブ付きページを含む。
  - 画面やレポートの視覚的設計用
  - 単一の画面またはレポートを構成する 1 つ以上の DDS レコードの集まり
  - フィールド、レコード、またはファイル全体のプロパティにアクセスする。
  - 処理中にイベントが発生したかどうかをシグナル通知する 2 桁の変数
- ユーティリティー・ノートブックに含まれるもの:
  - 「選択された DDS」ページ
  - 「エラー・リスト」ページ

- C. 「キーワードの作成」 ページ
  - D. 「コメント」 ページ
  - E. 上記すべて
5. 詳細ページには詳細モードとリスト・モードの 2 つのモードがあります。詳細モードでは DDS オブジェクトごとの詳細が表示され、リスト・モードではページ内に多くの DDS オブジェクトが表示されます。(T, F)
  6. 設計ページで簡単にできること:
    - A. DDS オブジェクトを作成する。
    - B. DDS オブジェクトを編集する。
    - C. DDS オブジェクトをサイズ変更する。
    - D. DDS オブジェクトを移動する。
    - E. レコードを作成する。
    - F. フィールドを作成する。
    - G. 固定表示を作成する。
    - H. 上記すべて
  7. レコードをグループ化することにより、バックグラウンドで関連レコードを参照しながら、レコードを処理できます。(T, F)
  8. プロパティ・ノートブックにはモードがありません。オブジェクトのプロパティを変更すると、選択されたオブジェクトはすぐに変更されます。(T, F)
  9. DDS ソースは、LPEX エディターまたは CODE エディターが使用する検査プログラムと同じ検査プログラムを使用して検査されます。(T, F)
  10. CODE 設計機能では、設計モードと編集モードを切り替えることができます。(T, F)
  11. LPEX エディターで行ったように、iSeries 上でソースをコンパイルできます。(T, F)

## 実習

この練習問題では、「リモート・システム」ビューから CODE 設計機能を使用する方法について学びました。そこで、プリンター・レポートを作成してみてください。ライブラリー RSELABxx 内のメンバー QDDSSRC の物理ファイル仕様 REFMST を使用できます。少し時間を掛けて、この物理ファイルのフィールドや情報を調べます。レポートを作成するとき、この情報を再び参照しなければならない場合があります。ファイル REFMST を詳しく理解していれば、印刷装置ファイルを作成できます。タスクの途中で困ったときは、Development Studio Client for iSeries のオンライン・ヘルプを使用してください。

## ここで学んだ内容

この練習問題で学んだこと: CODE 設計機能で表示装置ファイルを変更し、画面を追加しました。グループ、レコード、およびフィールドの追加方法について学びました。自分の DDS を保管し、そのコンパイル・エラーを検査しました。コンパイル・エラーがないので、DDS をコンパイルしました。

次の練習問題では、iSeries システム上で実行されるアプリケーションをデバッグおよびテストする方法について学びます。

## 練習問題 7: ILE RPG プログラムのデバッグ

この練習問題の目標は、デバッガー・フィーチャーを検討することです。デバッガーの開始、ブレークポイントの設定、変数のモニター、プログラムの実行およびステップイントゥ、「デバッグ」ビューでの呼び出しスタックの表示、ブレークポイントの除去、ストレージ・モニターの追加、監視ブレークポイントの設定、およびデバッグ・パースペクティブのすべての設定を行います。

この練習問題を終了すると、以下のことができるようになります。

- デバッガーのフィーチャーのいくつかを説明する。
- デバッグ・セッションを開始する。
- 行ブレークポイントを追加および削除する。
- 変数を表示する。
- 変数を変更する。
- 呼び出しスタックを表示する。
- プログラムを実行、ステップ、およびステップイントゥする。
- ソース・ビューとリスト・ビューを切り替える。
- ストレージ・モニターを追加する。
- 監視ブレークポイントを追加する。

## デバッグ・パースペクティブからの iSeries プログラムのデバッグ

統合デバッガーは、iSeries システム上で実行されているアプリケーションをデバッグおよびテストできるようにするソース・レベルのデバッガーです。デバッガーには、以下の操作が可能となる機能豊富な対話式のグラフィカル・インターフェースが用意されています。

- プログラムが iSeries システムで実行されているときに、ソース・コードまたはコンパイラー・リストを表示する。
- アプリケーション・プログラムで行ブレークポイントを設定、変更、および削除する、使用可能および使用不可にする。「ブレークポイント」ビューを使用して、すべてのブレークポイントを容易に管理できます。
- 監視ブレークポイントを設定し、指定の変数に変更されたときには必ずプログラムを停止させる。
- プログラムの呼び出しスタックを「デバッグ」ビューに表示する。デバッグすると、呼び出しスタックは動的に更新されます。呼び出しスタック・エントリをクリックすれば、デバッグ・プログラムのソースを表示できます。
- 一度に 1 行のコードをステップスルーする。

- プログラム呼び出しおよび ILE プロシージャ呼び出しをステップイントまたはステップオーバーする。
- 「モニター」ビューに変数とその値を表示する。この値は、プログラムの実行に与える影響を確認するために容易に変更できます。
- 「モジュール/プログラム」ビューを使用して、大きなプログラムでプロシージャ呼び出しを素早く容易に見つける。
- 個々のスレッドを使用可能および使用不可にする機能によりスレッドごとに個別のスタックを維持したまま、マルチスレッド化アプリケーションをデバッグする。
- iSeries の代わりにワークステーションからソースをロードする。これは、実動マシンでこのソース・コードを使用したくない場合に有効です。
- クライアント/サーバー・アプリケーションおよび分散アプリケーションをデバッグする。

デバッガーは、RPG/400 と ILE RPG、COBOL と ILE COBOL、C、C++、および CL をサポートします。

次の練習問題では、デバッガーの基本フィーチャーのいくつかについて学びます。この練習問題の目的として、ILE RPG/400 プログラムをデバッグします。RPG を知らなくても、心配には及びません。

## 統合デバッガーの開始

ここでは ILE RPG プログラム **PAYROLLG** を取り上げます。

**注:** PAYROLLG は PAYROLL と同じ RPG プログラムですが、コンパイル・エラーがありません。

この練習問題では、エディター練習問題を完了せずにこの練習問題までスキップすることにした人に配慮し、PAYROLL の代わりに PAYROLLG を使用します。

デバッガーは複数の方法で開始できます。「リモート・システム」ビューのプログラムまたはサービスのポップアップ・メニューから直接開始する方法、または「起動構成」ウィンドウから開始する方法です。「リモート・システム」ビューから直接開始する場合、プログラムに渡されるパラメーターを指定することはできません。「起動構成」ウィンドウでは、プログラムの起動方法を変更し、パラメーターを指定することができます。練習問題を面白くするために、CL プログラム **CLR1** を使用して **PAYROLLG** を呼び出し、1つのパラメーターを **CLR1** に渡します。したがって、「起動構成」ウィンドウを使用します。

リモート・システム・エクスプローラー・パースペクティブで、「起動構成のデバッグ」ウィンドウをオープンしてください。

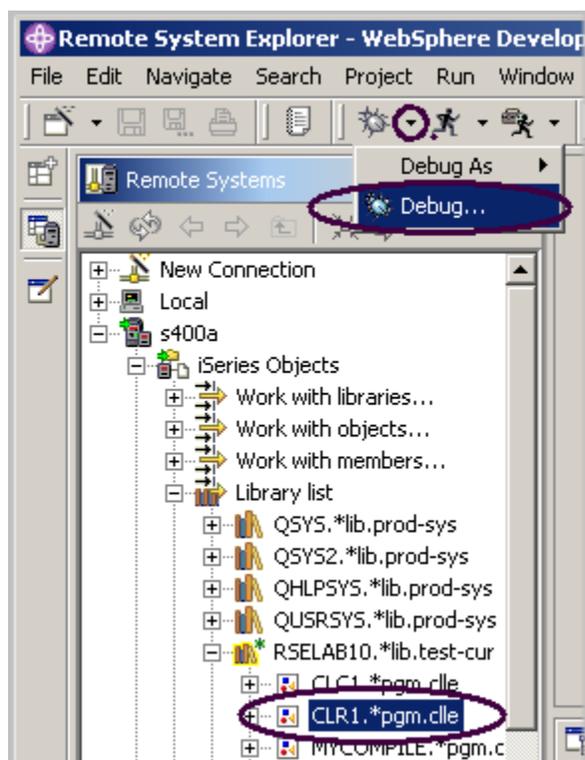


図 60: 「起動構成」 ウィンドウの開始

「リモート・システム」ビューで、以下のようにします。

1. 「ライブラリー・リスト」フィルターを拡張します (まだ拡張されていない場合)。
2. ライブラリー「RSELABxx」を拡張します (まだ拡張されていない場合)。
3. ライブラリー「RSELABxx」でプログラム「CLR1」を選択します。

ワークベンチ・ツールバーで、以下のようにします。

4. 図 60: 「起動構成」 ウィンドウの開始 に示されるように、デバッグ・アイコン  の横にあるメニュー・ドロップ・ダウン矢印をクリックします。
  5. ドロップ・ダウン・メニューから「デバッグ」を選択します。
- 「起動構成」ウィンドウが表示されます。

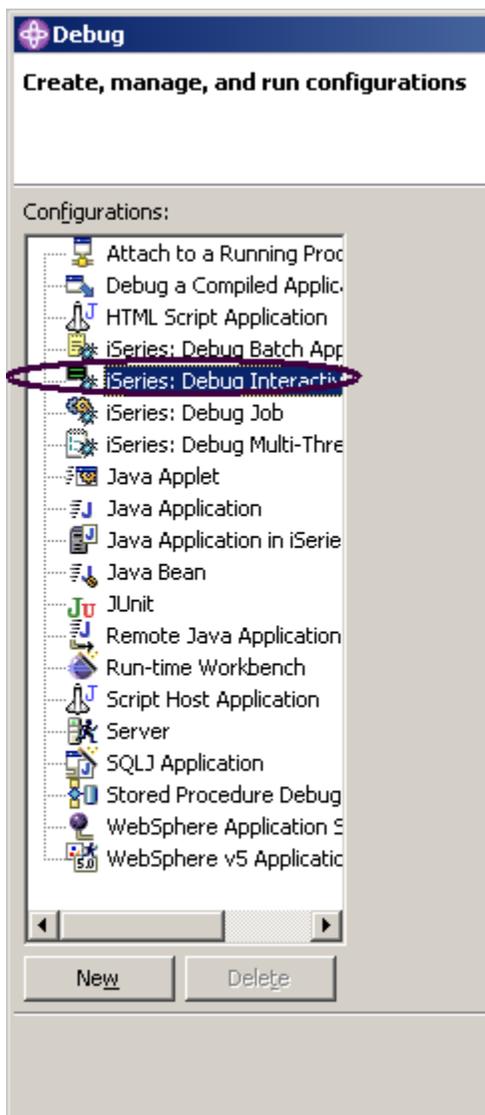


図 61: 「起動構成」 ウィンドウ

ツリー・ビューで、以下のようにします。

1. 「**iSeries: 対話式アプリケーションのデバッグ**」を選択します。
2. 「**新規**」をクリックします。

対話式アプリケーションのデバッグのペインが、ウィンドウの右側に表示されます。

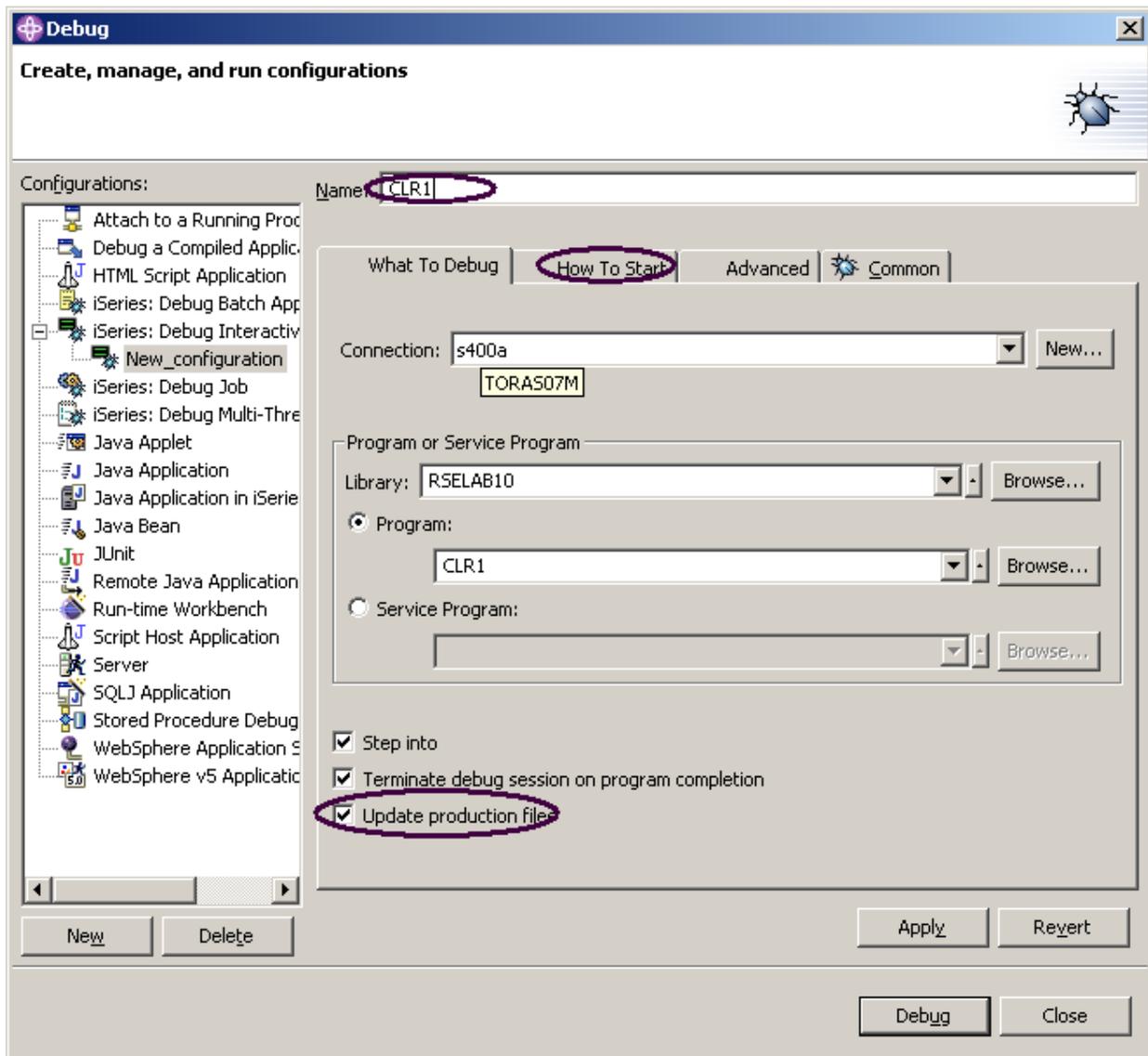


図 62: 対話式アプリケーションのデバッグ

3. プログラム名「CLR1」を「名前」フィールドに入力します。
4. 「実動ファイルを更新する」チェック・ボックスを選択します。
5. 「開始方法」タブをクリックします。

「開始方法」ページで、次のようにします。

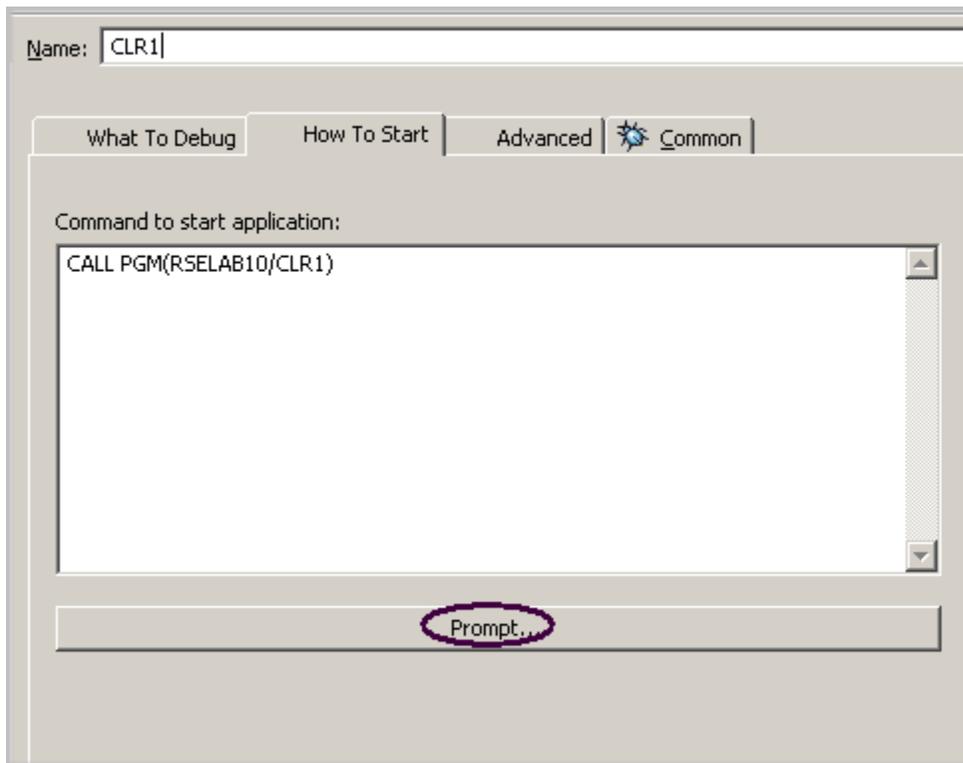


図 63: 「開始方法」 ページ

6. 「プロンプト」をクリックします。  
プロンプト・ウィンドウがオープンします。

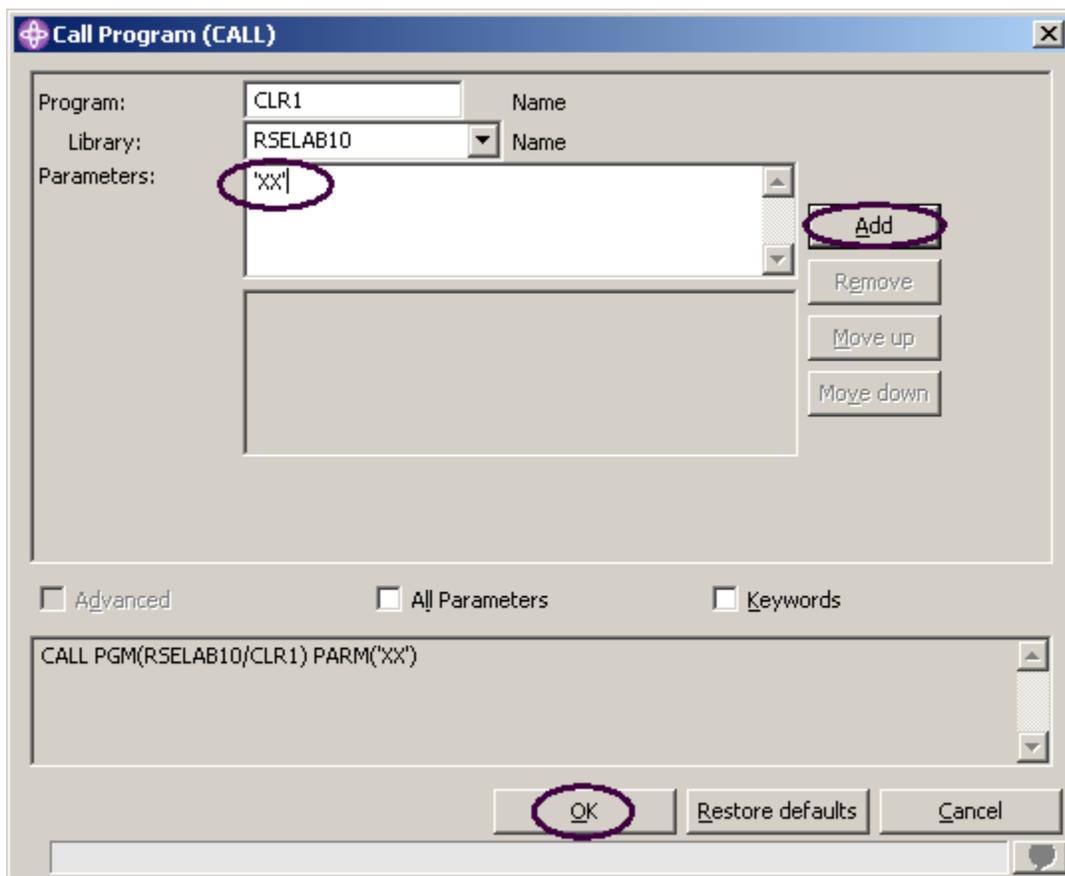


図 64: プロンプト・ウィンドウ

7. 「パラメーター」フィールドに「xx」と入力します。
8. 「追加」をクリックします。

下のリスト・ボックスにパラメーター値が表示されます。

9. 「OK」をクリックします。

プログラムの完全な開始コマンドが、ウィンドウに表示されます。

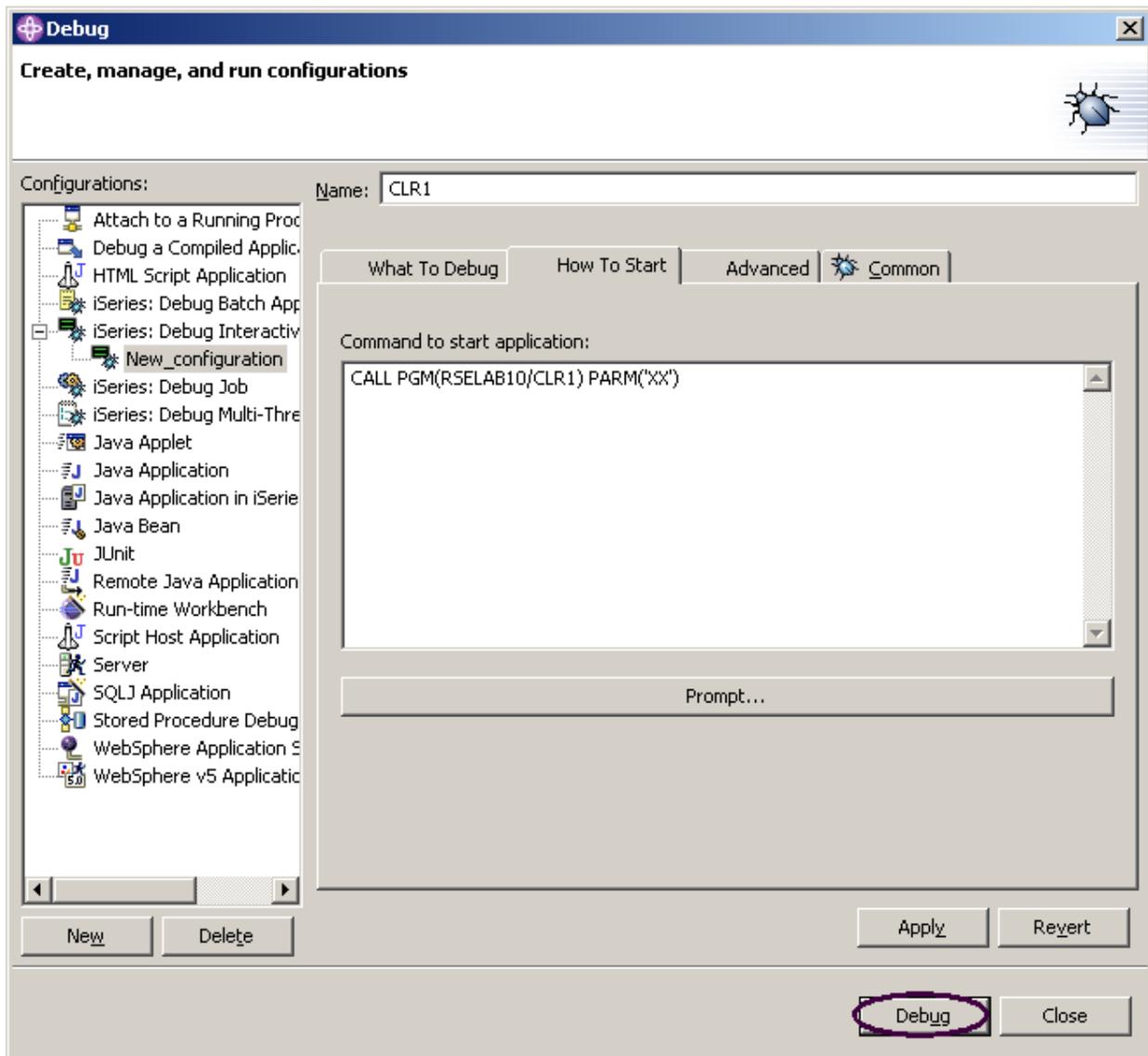


図 65: デバッグの準備完了

10. 「デバッグ」をクリックします。

次のようなエラー・メッセージが表示されることがあります。

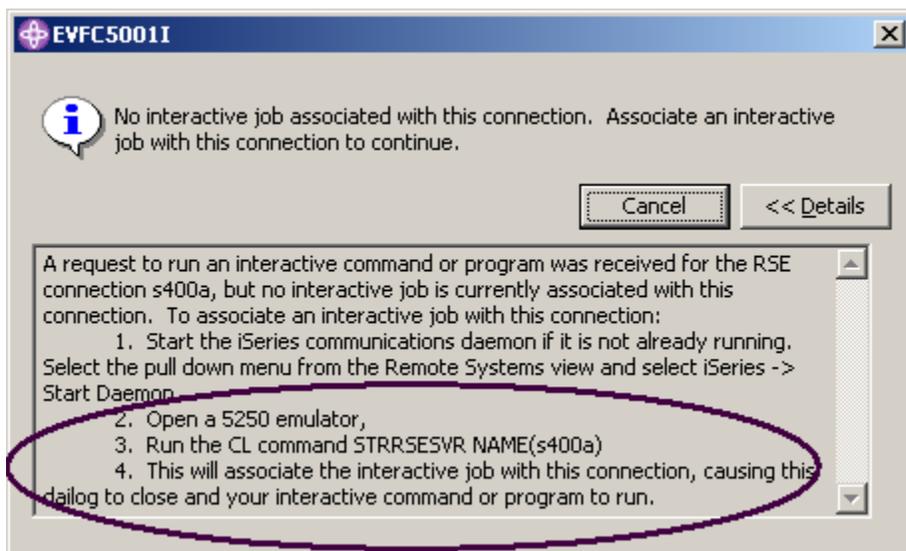


図 66: 対話式セッションが存在しない場合のエラー・メッセージ

しばらくすると、リモート・システム・エクスプローラー通信サーバーがシャットダウンされます。5250 エミュレーターに移動し、メッセージにある手順に従って、リモート・システム・エクスプローラー通信サーバーを再始動します。

このメッセージをキャンセルする必要はありません。リモート・システム・エクスプローラー通信サーバーと対話セッションの間の接続が確立されるとすぐに、このメッセージは除去されます。

これで、デバッグ・パースペクティブがワークステーションにロードされます。

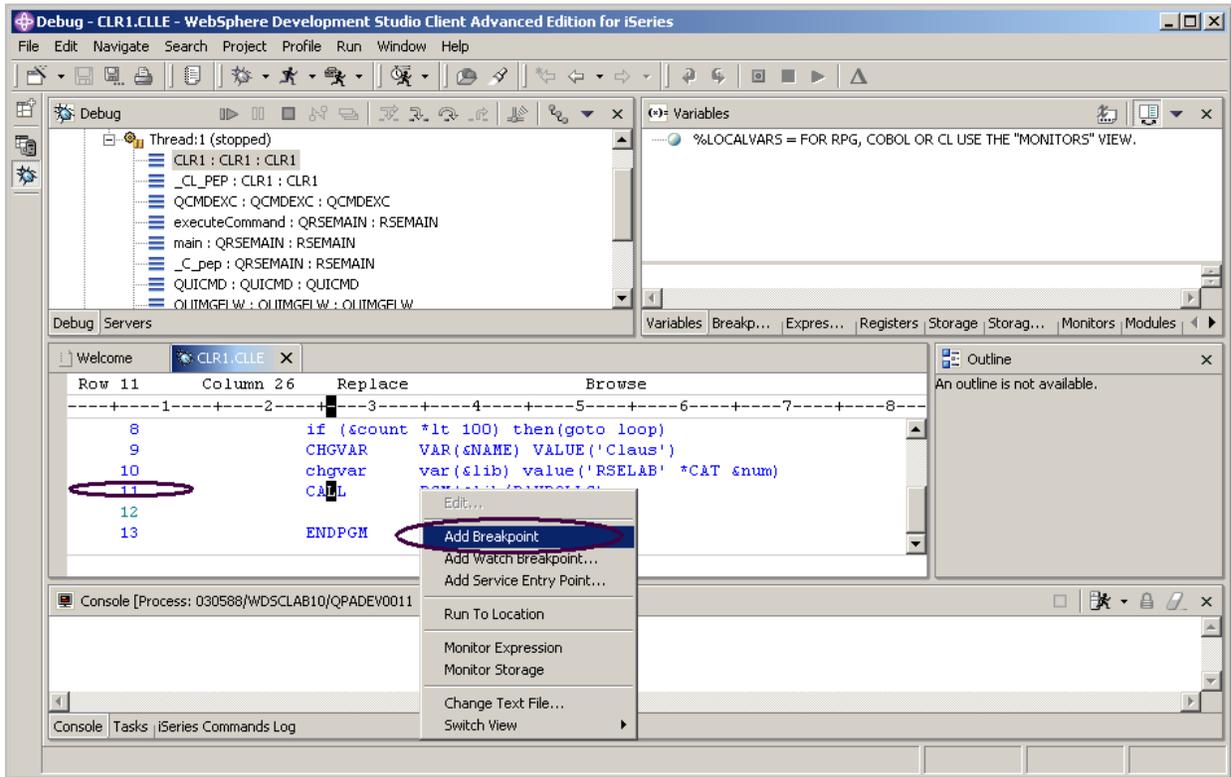


図 67: デバッグ・パースペクティブ(ブレークポイントの追加)

プログラムが iSeries 上でアクティブになり、最初の実行可能ステートメントで停止したとき、デバッガはソースを表示します。

## ブレークポイントの設定

実行可能行でのみブレークポイントを設定できます。実行可能行はすべて、青で表示されます。

ブレークポイントを設定するための最も簡単な方法は、ソース・ビューで行を右クリックする方法です。

1. 11 行目の任意の場所で右クリックします。
2. ポップアップ・メニューから「**ブレークポイントの追加**」を選択します。

接頭部にチェックマークを持つ点は、その行にブレークポイントが設定されていることを示します。接頭部域とは、ソース行の左にある灰色の細いマージンのことです。

ループが 99 回繰り返されたときにループに条件付きブレークポイントを追加するには、以下のようになります。

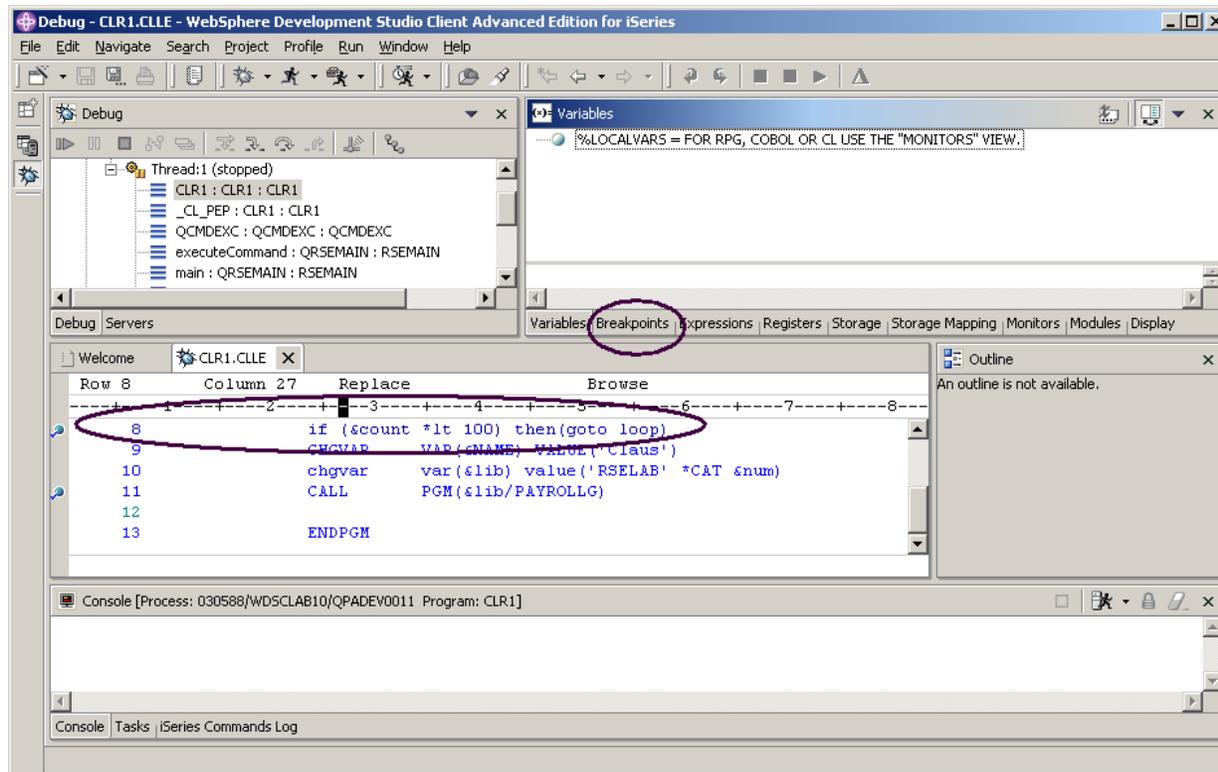


図 68: ブレークポイントの追加とブレークポイント・ビューの選択

3. 8 行目を右クリックします。
4. ポップアップ・メニューから「ブレークポイントの追加」を選択します。
5. 図 68: ブレークポイントの追加とブレークポイント・ビューの選択 に示されるように、ワークベンチの右上のペインで「ブレークポイント」タブを選択します。  
「ブレークポイント」ビューで、以下のようになります。

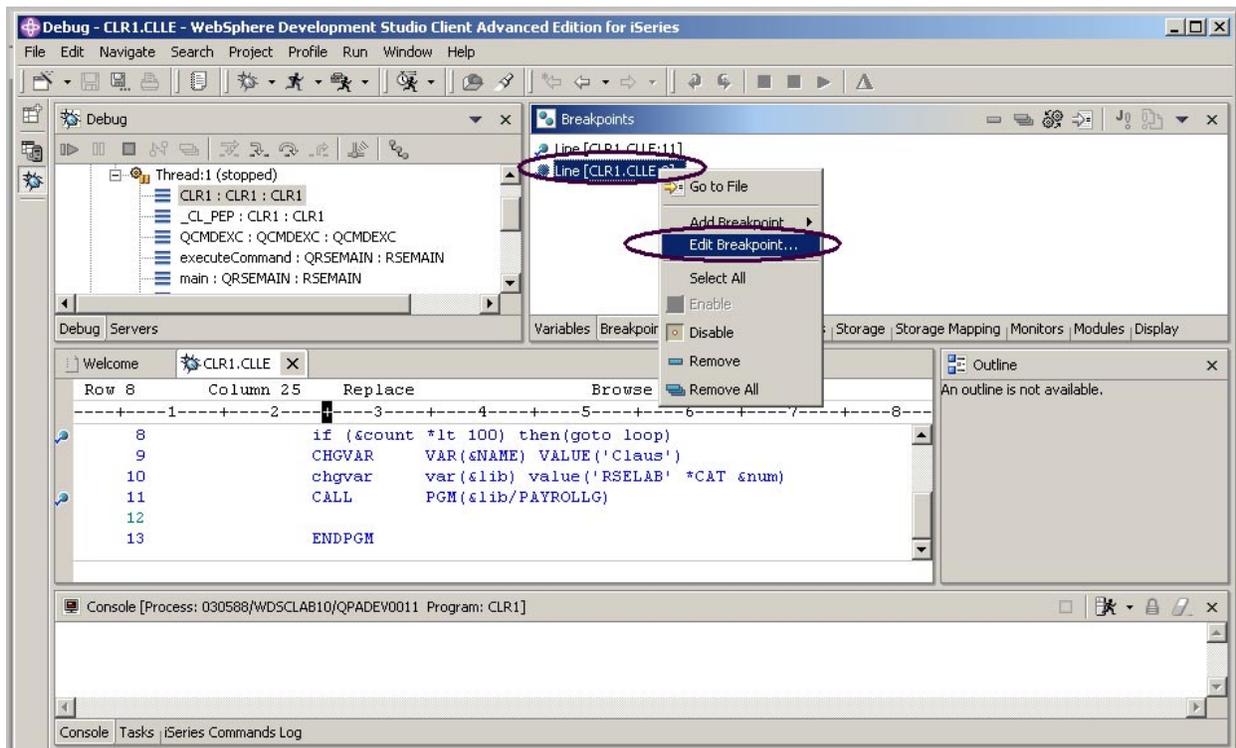
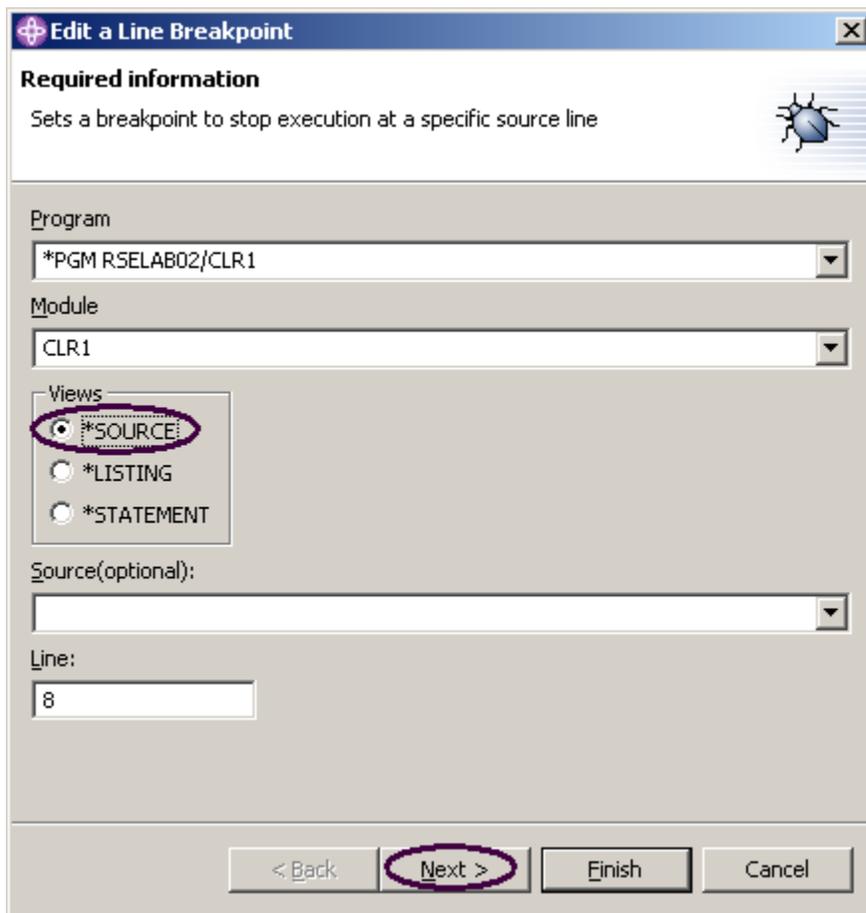


図 69: 「ブレイクポイント」ビューでのブレイクポイントの編集

6. 8 行目のブレイクポイントを右クリックします。
7. ポップアップ・メニューから「ブレイクポイントの編集」を選択します。

8. ビューのリストから「\*SOURCE」を選択します。



**Edit a Line Breakpoint**

**Required information**  
Sets a breakpoint to stop execution at a specific source line

Program  
\*PGM RSELAB02/CLR1

Module  
CLR1

Views

- \*SOURCE:
- \*LISTING
- \*STATEMENT

Source(optional):

Line:  
8

< Back   **Next >**   Finish   Cancel

9. 「行ブレークポイントの編集」ウィンドウで「次へ」をクリックします。  
ループが 99 回以上実行されたときのみ、停止したいとします。それには、「頻度」グループの「始め」フィールドを「99」に設定します。

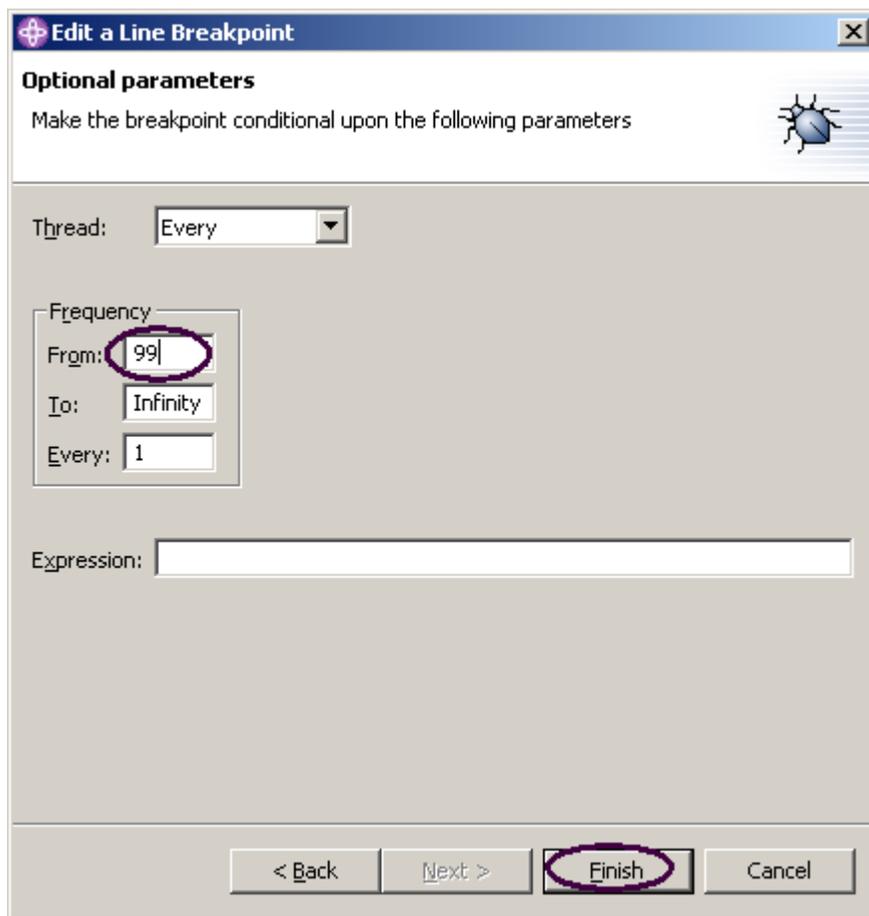


図 70: 中断の条件の指定

10. 「始め」入力フィールドに「99」と入力します。
11. 「終了」をクリックします。

### 変数のモニター

変数は「モニター」ビューでモニターできます。ここでは、変数 **&COUNT** をモニターします。

ソース・ビューで、以下のようにします。

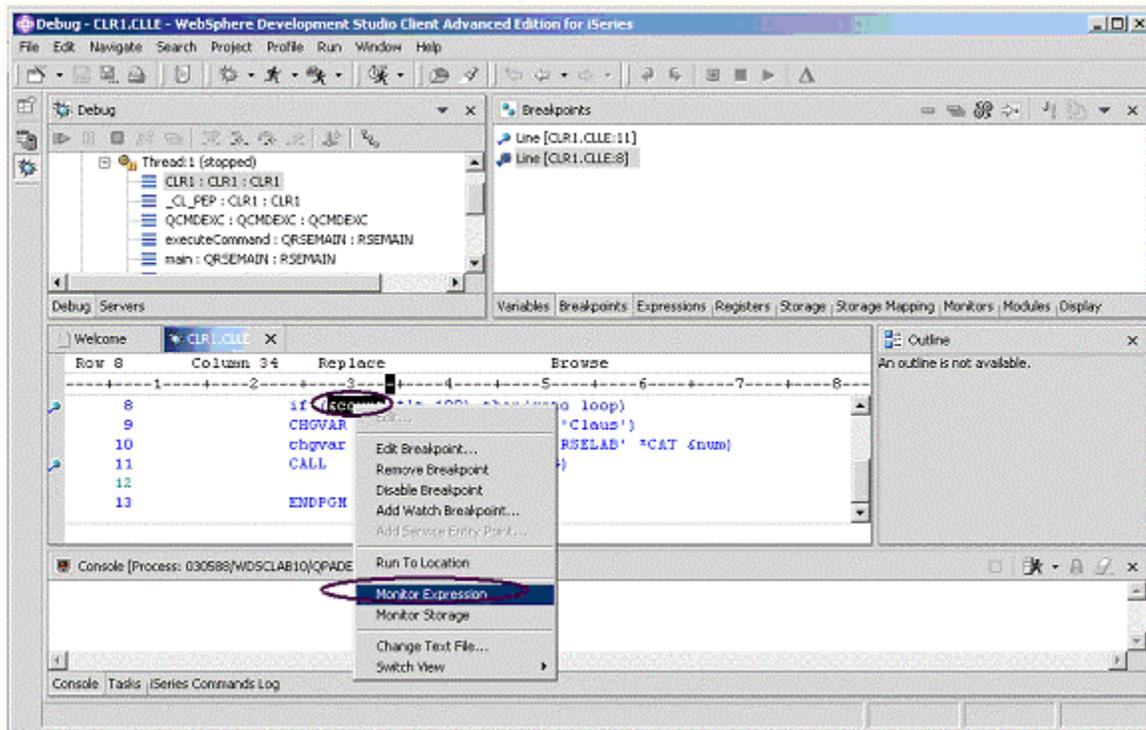


図 71: モニターする変数の選択

1. 変数「&COUNT」をダブルクリックします。
2. 「&COUNT」を右クリックします。
3. ポップアップ・メニューから「式のモニター」を選択します。

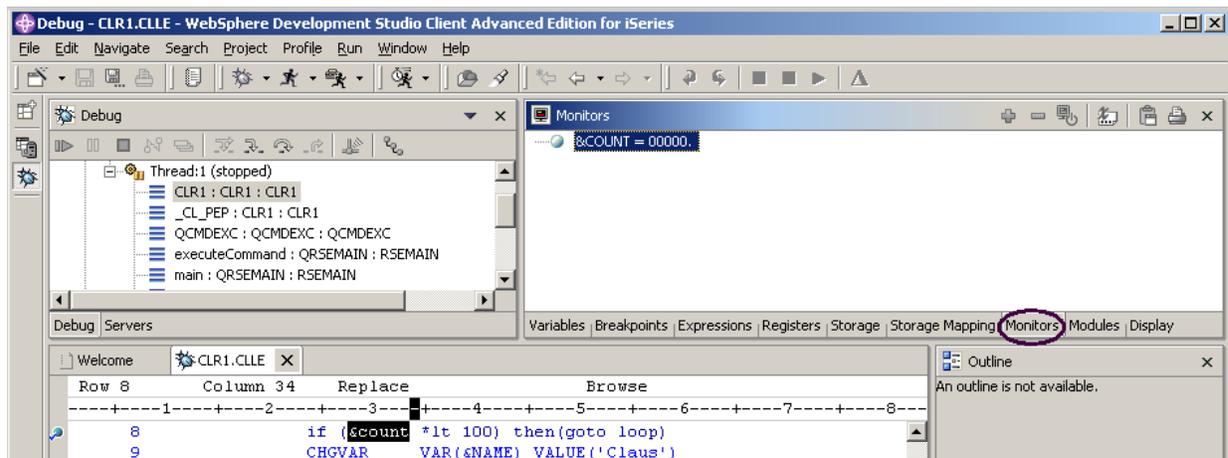


図 72: 「モニター」ビューと変数 &COUNT

変数が「モニター」ビューに表示されます。  
現在の値はゼロです。

ヒント: 変数の値を「モニター」に追加せずに素早く参照したい場合は、1 秒ほどマウス・ポインターを変数の上に置けば、ポップアップ・ウィンドウに値が表示されます。

これで、いくつかのブレークポイントが設定されたため、アプリケーションの実行を開始できます。



図 73: 「デバッグ」ツールバーの再開アイコン

4. 「デバッグ」ツールバーの「再開」  をクリックします。

プログラムが実行を開始し、8 行目のブレークポイントで停止します。(しばらくお待ちください。デバッガーは 98 回停止しなければなりません。ただし、条件により 99 回目までは続けて実行されます。)

「モニター」ビューで、&COUNT の値が現在 99 であることに注意してください。

5. 「再開」をもう一度クリックします。

プログラムは再び 8 行目のブレークポイントで停止し、&COUNT の値が 100 になります。

6. 11行目のブレークポイントまでプログラムが実行されるように、「再開」をもう一度クリックします。

## プログラムのステップイントゥ

デバッガーにより、プログラム呼び出しをステップオーバーまたはステップイントゥできます。プログラム呼び出しをステップオーバーすると、呼び出されたプログラムが実行され、呼び出し側プログラムにある次の実行可能ステートメントでデバッガーが停止します。これから、Payroll プログラムをステップイントゥします。



図74: 「デバッグ」ツールバーとステップ・デバッグ・アイコン

1. 「デバッグ」ツールバーで「ステップイントゥ」 をクリックします。PAYROLLG のソースが表示されます。

プログラムのコンパイルに使用したオプション (\*SRCDBG/\*LSTDBG for RPG、\*SOURCE、\*LIST、または \*ALL for ILE RPG) に応じて、このウィンドウにはソース・ビューまたはリスト・ビューのいずれかが表示されます。

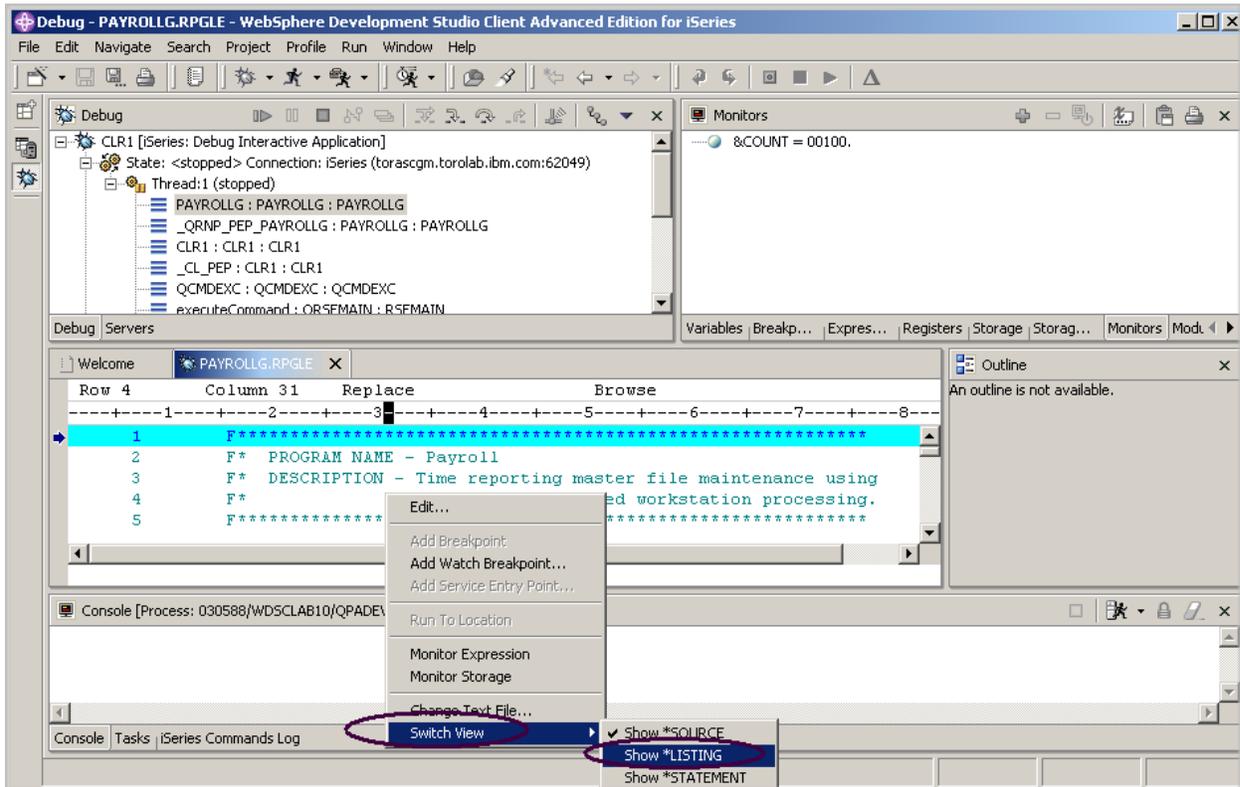


図 75: RPG ソース・ビュー

2. ソース・ビューの任意の場所で右クリックします。
3. ポップアップ・メニューから「ビューの切り替え」>「\*リストの表示」を選択します。

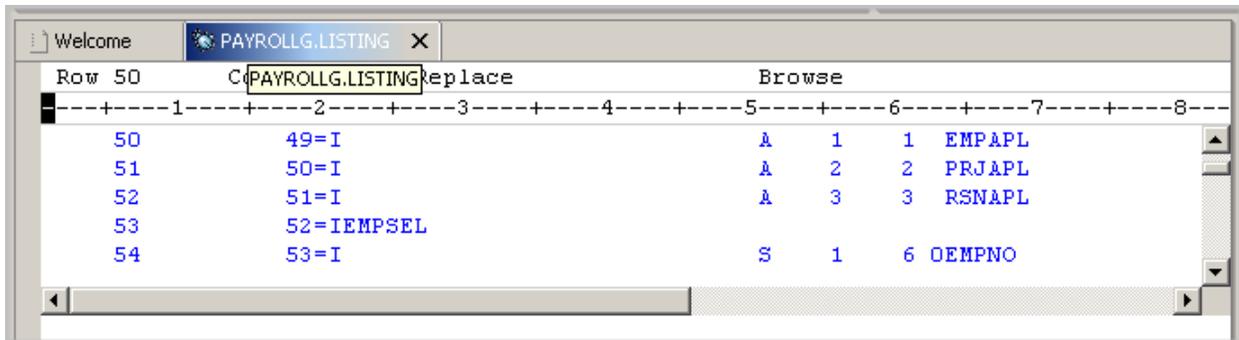


図 76: 記述ファイルの拡張入力仕様

4. ソースでページ・ダウンし、拡張ファイル記述を調べます。

PAYROLL プログラムには /Copy メンバーがありませんが、リスト・ビューには表示されません。再びソース・ビューに切り替えます。

5. ソース・ビューの任意の場所で右クリックします。
6. 「ビューの切り替え」 > 「\*ソースの表示」を選択します。

### 呼び出しスタック・エントリーのリスト

左上のペインにある「デバッグ」ビューに、すべての呼び出しスタック・エントリーがリストされます。このビューには、スレッドごとのツリー・ビューが含まれます。スレッドを拡張すれば、現在の実行ポイントのスタックにある各プログラム、モジュール、プロシージャ、およびメソッドを表示できます。スタック・エントリーをダブルクリックすると、使用可能な場合に限り、対応するソースが表示されます。それ以外の場合は、「ソース」ビューに「使用可能なデバッグ・データがありません」というメッセージが表示されます。「デバッグ」ビューで、スタック・エントリー「Thread1」がまだ拡張されていない場合は、その前にあるプラス符号 (+) をクリックして拡張します。

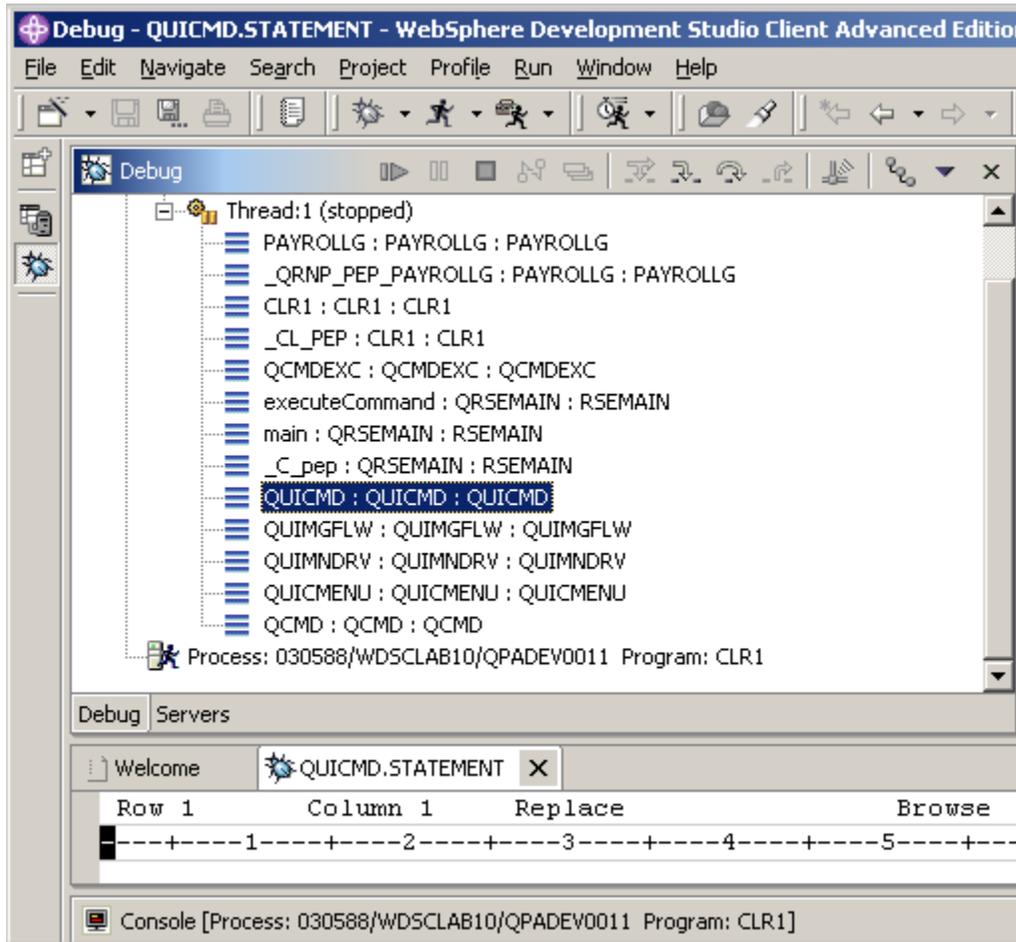


図 77: 「デバッグ」ビューでの呼び出しスタック

これにより、異なるプログラムおよび/または ILE モジュールを使用し、切り替えることができます。

## PAYROLLG でのブレークポイントの設定

ここでは、PAYROLLG でいくつかのブレークポイントを追加します。

ブレークポイントを追加するには、以下のようにします。

1. 「Thread1」内の「PAYROLLG」を選択します。
2. **57** 行目までスクロールします。
3. **57** 行目の接頭部域をダブルクリックします。

この行の接頭部域にブレークポイント・アイコンが追加され、ブレークポイントが設定されたことが示されます。

4. **58** 行目に上記のステップを繰り返します。

5. **87** 行目に上記のステップを繰り返します。

すべてのブレークポイントを表示するには、左上のペインから「**ブレークポイント**」タブを選択します。このビューには、デバッグ・セッションで現在設定されているすべてのブレークポイントが表示されます。

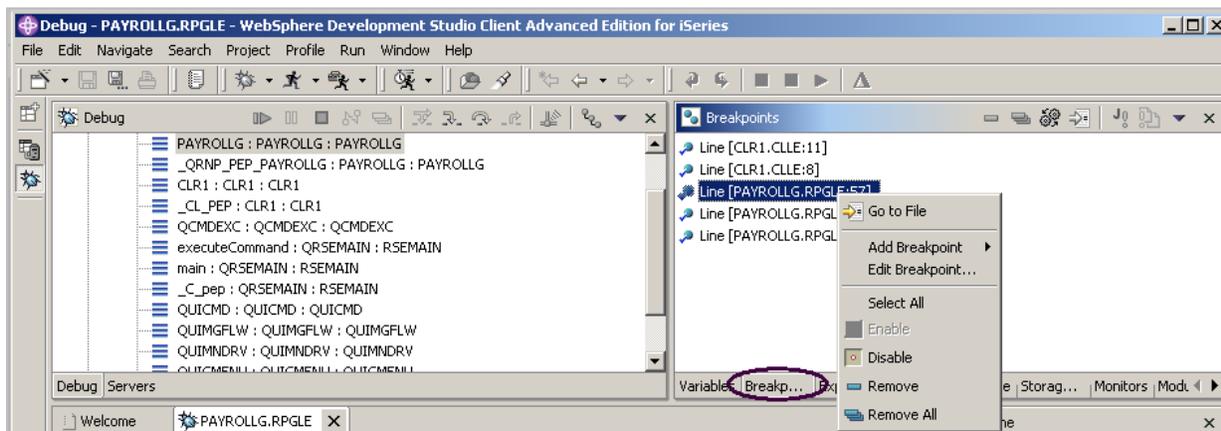


図 78: 「ブレークポイント」ビュー

これは、ブレークポイントを操作するために便利な場所です。ブレークポイントは除去したり、使用不可/使用可能にしたり、追加したり、編集したりすることができます。これらのタスクは、ビュー域で右マウス・クリックしたときのポップアップ・メニューから使用できます。任意のエントリをダブルクリックして、ブレークポイントが設定されているソースを表示します。

## ブレークポイントの除去

ブレークポイントは、ソース・ビューから容易に除去できます。

1. **58** 行目を右クリックします。

2. 「**ブレークポイントの除去**」を選択します。

アイコンが接頭部域から除去され、その行にブレークポイントが設定されていないことが示されます。

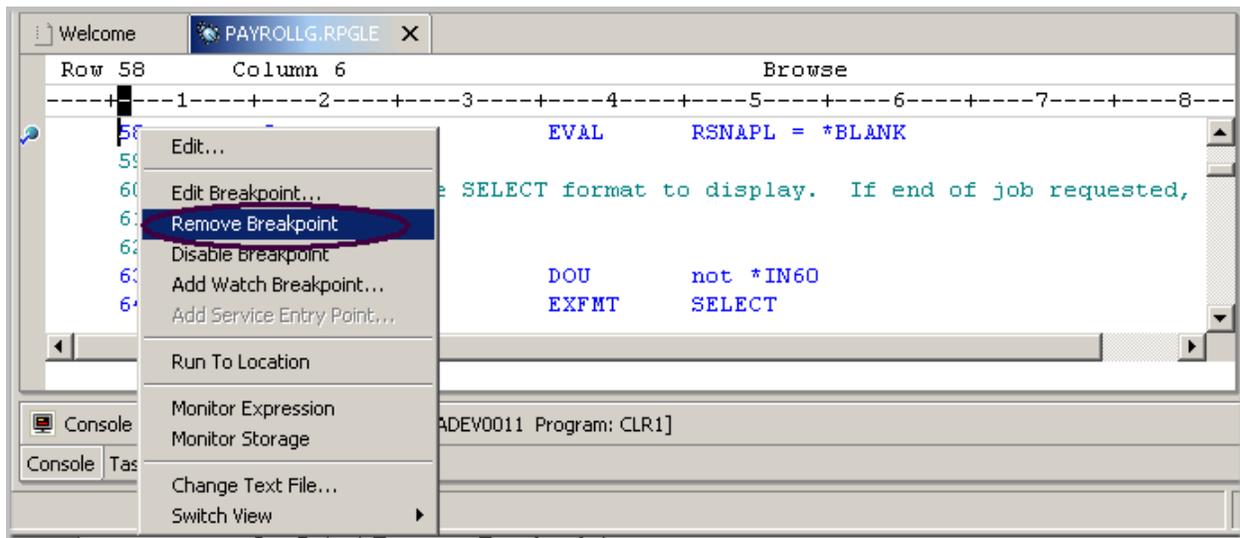
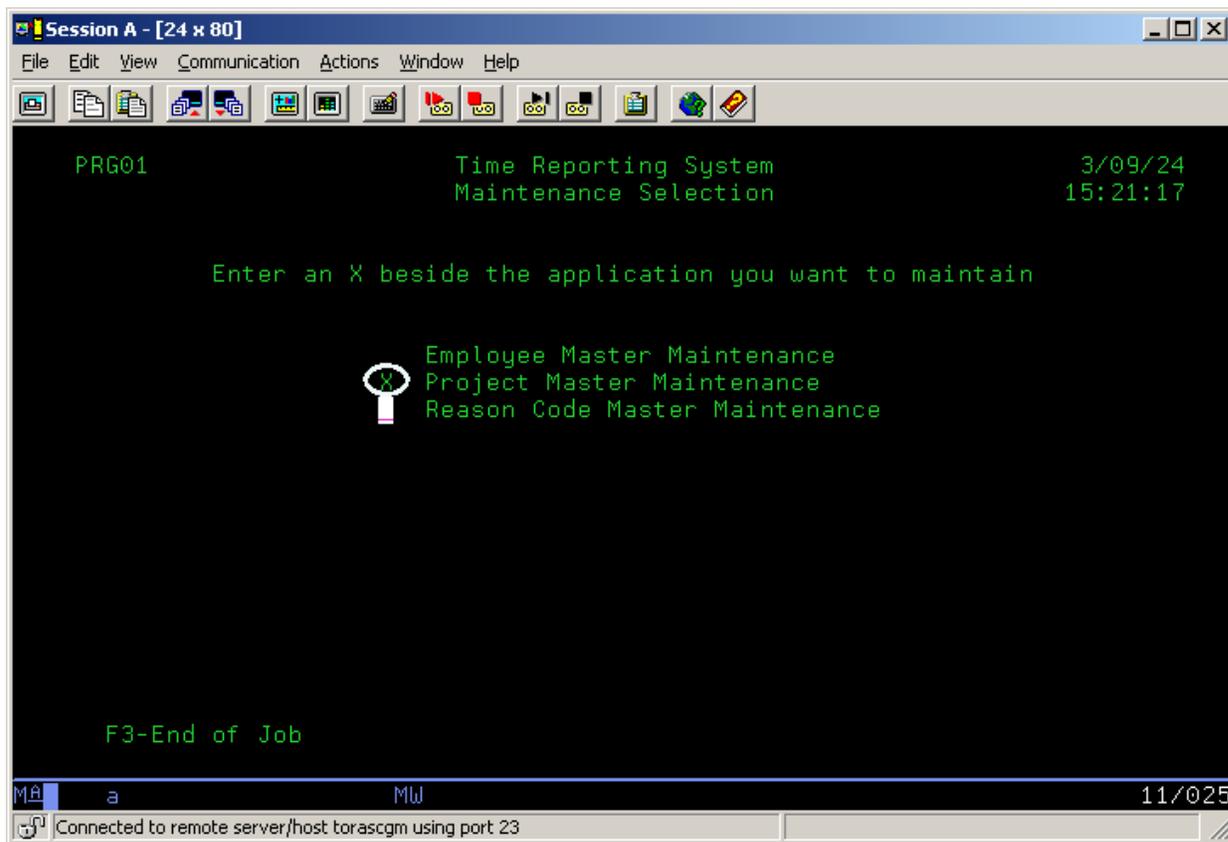


図 79: ブレークポイントの除去

これで、PAYROLLG プログラムを実行する準備が整いました。

3. 「デバッグ」ツールバーの「再開」  をクリックします。
4. プログラムが開始され、57 行目のブレークポイントまで実行されます。
5. 「再開」をもう一度クリックします。

プログラムは、5250 エミュレーション・セッションからの入力を待ちます。



6. 「プロジェクト・マスター保守」オプションの横に「X」と入力します。

7. エミュレーション・セッションで **Enter** を押します。

プログラムが、87 行目のブレークポイントまで実行されます。

### PAYROLLG での変数のモニター

ここでは、PAYROLLG で変数をモニターし、変更します。

ソース・ビューで、以下のようにします。

1. 88 行目にある変数 EMPAPL をダブルクリックします。
2. 変数を右クリックします。
3. 「式のモニター」を選択します。

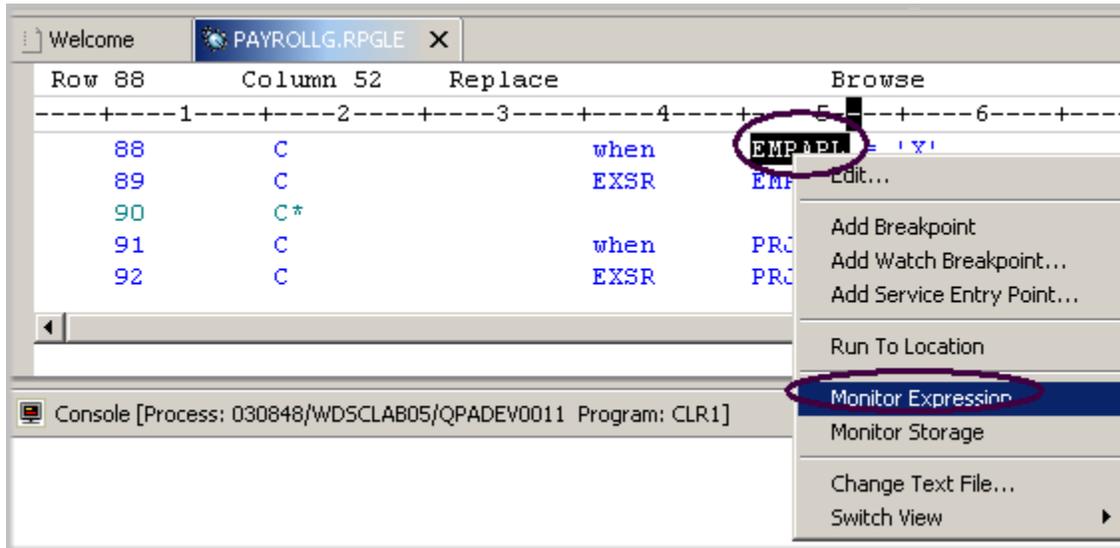


図 80: モニターする変数の追加

4. 右上のペインにある「モニター」タブをクリックします。  
変数が「モニター」ビューに表示されます。「従業員マスター保守」オプションを選択していないため、値はブランクです。
5. 同じようにして、モニターの 91 行目に変数 PRJAPL を追加し、113 行目に変数 RSDSC を追加します。

「プロジェクト・マスター保守」オプションを選択したため、変数 PRJAPL は「X」と等しくなっています。

「モニター」ビューで、以下のようにします。

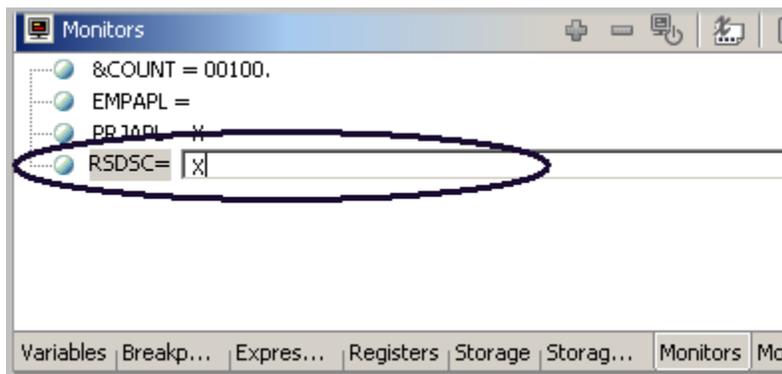


図 81: 変数の内容の変更

6. 変数「RSCDE」をダブルクリックします。  
値が入力フィールドに変わります。
7. 変数について新しい値「x」を入力します。
8. Enter を押します。

変数が正常に変更されました。

### ストレージ・モニターの追加

変数についてストレージ・モニターを追加すると、変数の位置するアドレスから始まるストレージを表示できます。ストレージは、16進数とテキストのフォーマットで表示されます。

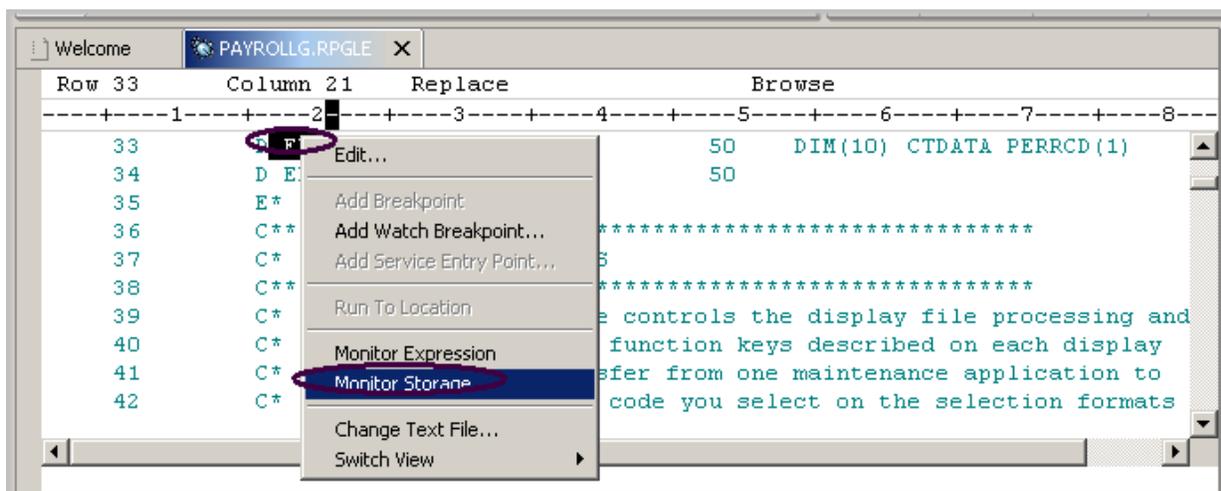


図82: ストレージ・モニターの追加

ソース・ビューで、以下のようにします。

1. **33** 行目にある変数「ERR」をダブルクリックします。
2. 右クリックで表示されるポップアップ・メニューから「**ストレージのモニター**」を選択します。

新規ページが「ストレージ」ビューに追加されます。タブに変数の名前が表示されます。

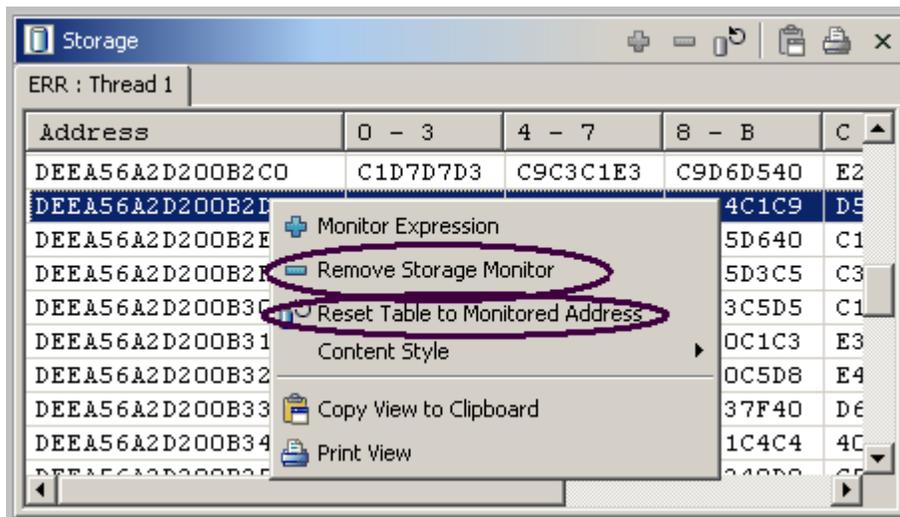


図83: ストレージ・モニターの変更

- 「ストレージ」ビューの右側にあるスクロール・バーを使用して、スクロールダウンします。メモリーの現在の内容を参照できます。
- ビュー域で右クリックします。
- 「モニターされるアドレスへのテーブルのリセット」を選択して、開始アドレスに戻ります。
- ビュー域で右クリックします。
- 「ストレージ・モニターの除去」を選択して、ストレージ・モニターを除去します。

### 監視ブレイクポイントの設定

監視ブレイクポイントは、変数に変更されたときにユーザーに通知します。アクションが実行されるまで、プログラムの実行は中断されます。

ソース・ビューで、**116** 行目に移動します。

- ソース・ビュー内の任意の場所をクリックして、**Ctrl+L** を押します。ソース域の下部に入力フィールドが追加されます。この入力フィールドに「116」と入力して、**116** 行目に移動します。
- 変数「\*IN60」をダブルクリックして強調表示します。
- 右クリックで表示されるポップアップ・メニューから「監視ブレイクポイントの追加」を選択します。

「監視ブレークポイント」ウィンドウが表示されます。「式」フィールドには、強調表示された変数「\*IN60」が事前に入力されています。デフォルトで「監視するバイト数」はゼロに設定されます。つまり、変数は、定義された長さで監視されます。

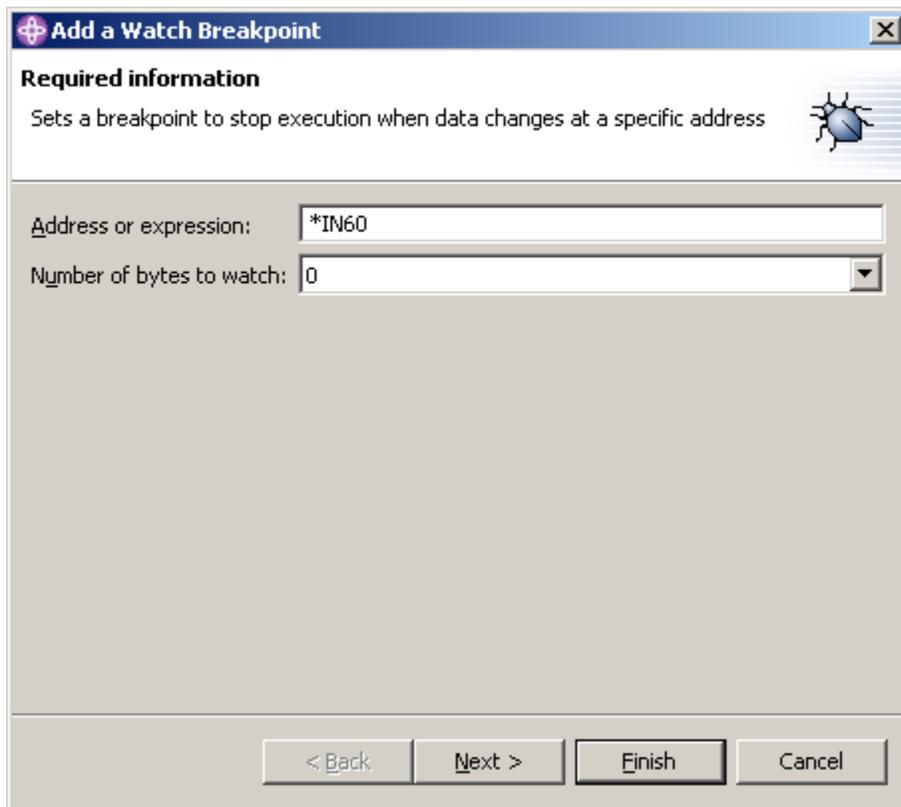
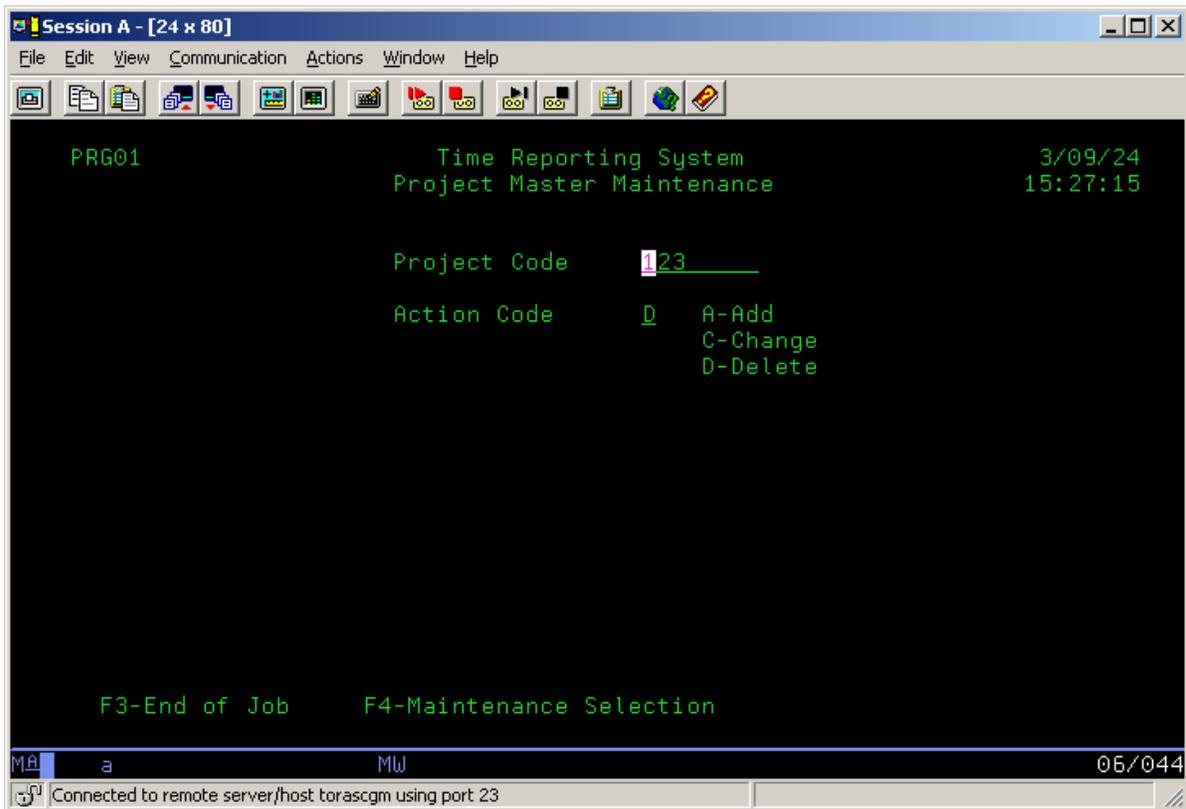


図84: 監視ブレークポイントの追加

4. 「終了」をクリックします。  
これで監視ブレークポイントが設定されました。
5. 「デバッグ」ツールバーの「再開」をクリックします。

アプリケーションは、5250 エミュレーション・セッションからの入力を待ちます。



5250 エミュレーション・セッションで、以下のようにします。

6. 「プロジェクト・コード」に「123」を入力し、「アクション・コード」フィールドに「D」（削除）を入力します。
7. **Enter** を押します。

変数 \*IN60 が変更されたことを示すメッセージが表示されます。

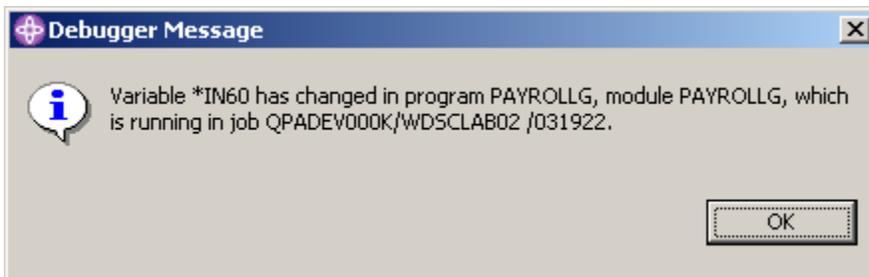


図 85: 監視ブレークポイントのメッセージ

8. 「OK」をクリックします。

プログラムは **465** 行目で停止します。この行は、変数 \*IN60 が変更される原因となったステートメントのすぐ後ろにあります。

### デバッグ・セッションのクローズ

1. 「デバッグ」ツールバーの「再開」をクリックします。

アプリケーションは、**5250** エミュレーション・セッションからの入力を待ちます。

2. **5250** エミュレーション・セッションに切り替えます。

3. **F3** を押して、ジョブを終了します。

「プログラムが終了しました」というメッセージが表示されます。



4. 「OK」をクリックします。

以下のステップを実行して、デバッグ・パースペクティブをクローズします。

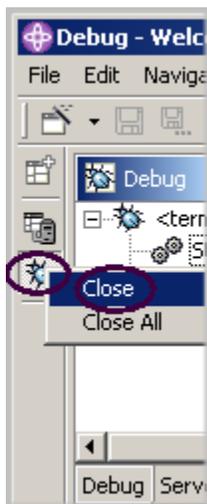


図 86: デバッグ・パースペクティブのクローズ

5. ワークベンチの左側フレームにあるデバッグ・アイコン  を右クリックします。
6. ポップアップ・メニューから「クローズ」を選択します。

次の練習問題に進む準備ができたかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

1. ここでデバッガーを開始することができます。
  - A. 「リモート・システム」ビュー
  - B. 「起動構成」ウィンドウ
  - C. 両方
2. 実行可能行でのみブレークポイントを設定できます。(T, F)
3. ブレークポイントを設定するための最も簡単な方法:
  - A. その行を右クリックする。
  - B. その行の後を右クリックする。
  - C. その行の前を右クリックする。
  - D. 上記すべて
4. 変数および標識を変更できる場所:
  - A. 「リモート・システム」ビュー
  - B. 「デバッグ」ビュー
  - C. 「モニター」ビュー
  - D. ストレージ・モニター
  - E. 上記すべて
5. デバッガーでできること:
  - A. プログラム呼び出しをステップオーバーする。
  - B. プログラム呼び出しをステップイントゥする。
  - C. 両方
6. 「デバッグ」ビューでは、すべての呼び出しスタック・エントリーがリストされます。このビューには、スレッドごとのツリー・ビューが含まれます。(T, F)
7. ブレークポイントで実行できるアクション:
  - A. 削除
  - B. 追加
  - C. 使用不可にする
  - D. 使用可能にする
  - E. 編集
  - F. 上記すべて

8. 変数についてストレージ・モニターを追加すると、変数の位置するアドレスから始まるストレージを表示できます。(T, F)
9. ストレージ・モニターがサポートする表示フォーマット:
  - A. 16 進数および文字
  - B. 文字のみ
  - C. 10 進数
  - D. 上記すべて
  - E. A と B
10. \_\_\_\_\_ ブレークポイントは、変数に変更されたときにユーザーに通知します。アクションが実行されるまで、プログラムの実行は中断されます。ここでデバッガーを開始することができます。
  - A. 監視
  - B. サポート
  - C. Java 例外
  - D. タイプ

## 実習

この練習問題では、デバッガー・フィーチャーの操作を学びました。そこで、自分のソースでデバッグしながら、行ブレークポイントを設定、変更、および削除し、行ブレークポイントを使用可能および使用不可にし、監視ブレークポイントを設定し、変数を表示および変更し、呼び出しスタックを表示してみます。タスクの途中で困ったときは、**Development Studio Client for iSeries** のヘルプを使用してください。

## ここで学んだ内容

この練習問題では、デバッガー・フィーチャーについて学びました。デバッガーの開始、ブレークポイントの設定、変数のモニター、プログラムの実行およびステップイントゥ、「デバッグ」ビューでの呼び出しスタックの表示、ブレークポイントの除去、ストレージ・モニターの追加、監視ブレークポイントの設定、およびデバッグ・パースペクティブのすべての設定を行いました。

次の練習問題では、フィルターおよびアクションの作成し、**iSeries** オブジェクトをすべてリモート・システム・エクスプローラーから管理する方法について学びます。つまり、リモート・システム・エクスプローラーはどのようにしてユーザーの作業を編成および統合し、どのようにしてその作業をより容易にするのかを確認します。



## 練習問題 8: リモート・システム・エクスプローラーの探査

この練習問題では、リモート・システム・エクスプローラー・パースペクティブを使用して、前の練習問題で使用した **iSeries** オブジェクトを操作します。フィルターの定義、アクションの実行、独自のアクションの定義がいかに簡単であるかも分かります。つまり、リモート・システム・エクスプローラーはどのようにしてユーザーの作業を編成および統合し、どのようにしてその作業をより容易にするのかを確認します。

この練習問題を終了すると、以下のことができるようになります。

- リモート・システム・エクスプローラーを説明する。
- リモート・システム・エクスプローラー、フィルター、ユーザー・アクション、および実行中のコマンドを説明する。
- リモート・システム・エクスプローラーをオープンする。
- フィルター(ライブラリー、オブジェクト)を作成する。
- ライブラリー・フィルターを変更する。
- ユーザー定義のアクションを作成する。
- プロパティーを表示する。
- **iSeries** テーブル・ビューからコマンドを実行する。

### リモート・システム・エクスプローラーの紹介

CODE プロジェクト・オーガナイザーの多くの機能は、**WebSphere Studio** の機能によって、ADM パーツにアクセスするという例外に置き換えられました。

リモート・システム・エクスプローラーは、ワークステーションの **PDM** (プログラミング開発マネージャー) を置き換えます。現時点では、**PDM** の全機能を備えているわけではありませんが、今後、完全に **PDM** から置き換わる予定です。

リモート・システム・エクスプローラーにより、以下を行うことができます。

- **iSeries** ライブラリー、オブジェクト、メンバー、**IFS** ファイル、**UNIX** ファイル、およびローカル・ファイルのリストに素早くアクセスできるようにして、作業を簡素化する。
- これらのリストにあるコンテキスト依存ポップアップ・メニューを使用して、リモート・システム **LPEX** エディター、**CODE** 設計機能、または統合デバッガーなどのアクションや、他の共通 **iSeries** アクションを実行する。
- 「ユーザー・アクションの操作」オプションを使用して、独自のユーザー定義のアクションを作成および管理し、それをポップアップ・メニューに表示させる。

- コマンド・サポートを使用して、エミュレーター・セッションに切り替えずに、iSeries コマンドまたはローカル・コマンドを入力し、繰り返すことによって、生産性を向上させる。

## ライブラリー・フィルターの作成

リモート・システム・エクスプローラー・パースペクティブで、特定の iSeries オブジェクトを操作します。

これまでの練習問題で、ライブラリー・リストを扱いました。ここでは、独自のライブラリー・フィルターを作成します。ライブラリー・フィルターにより、「リモート・システム」ビューに iSeries システムからライブラリーのセットがリストされます。その前にまず、フィルターとは何かを理解しましょう。

ユーザーは、システム間の統合に使用できる柔軟なユーザー・インターフェースを必要とします。開発者は、多くの場合、ライブラリー、ソース・ファイル、およびソース・メンバーを作成する必要があります。

「リモート・システム」ビューには、サブシステム、フィルター、およびフィルターごとに指定された項目が表示されます。このビューでは、フィルターに掛けられた情報を、分かりやすいツリー・ビューで編成できます。フィルターは、ライブラリー、オブジェクト、およびソース・メンバーのコレクション用に作成できます。

リモート・システム・エクスプローラーは iSeries ネイティブ・ファイル・システム (QSYS) サポートを提供します。iSeries ネイティブ・ファイル・システムでは、iSeries ホスト上にあるオブジェクトを照会し、そのオブジェクトに対してアクションを実行できます。フィルターを使用すれば、システム内のエレメントを容易に編成できます。フィルター機能は、iSeries ネイティブ・ファイル・システム・オブジェクト (ライブラリー、オブジェクト、メンバーなど) をリストするために使用します。

デフォルトで、初期ライブラリー・リスト・フィルターも表示されます。ここには、iSeries ホスト上のユーザー・プロファイルに定義されている初期ライブラリーが表示されます。このリストを操作して、リスト内でライブラリーを追加または移動することができます。

リモート・システム・エクスプローラーは、ネイティブ・ファイル・システム (QSYS) をサポートし、iSeries ホスト上にあるオブジェクトを照会できるようにします。フィルターを使用すれば、システム内のオブジェクトの表示方法を容易に編成できます。フィルターを拡張することでリストにアクセスするとき、ネイティブ・ファイル・システムのオブジェクト (ライブラリーやファイルなど) が表示され、そのリモート・オブジェクトに対してアクションを実行できます。

しばらくの間、リモート・システム・エクスプローラーを使用していた場合、ワークスペースに含まれるフィルターが多すぎて、容易にナビゲートできない場合があります。

この場合、フィルターはいくつかのプールに区分され、1つのプールに多くの異なるエレメントのフィルターが含まれた可能性があります。たとえば、あるフィルター・プールに給与計算プログラムのフィルターが含まれ、別のフィルター・プールに売掛管理プログラムのフィルターが含まれる場合があります。

リモート・システム・エクスプローラーは、ネイティブ・ファイル・システム (QSYS) をサポートし、iSeries システム上にあるオブジェクトを照会できるようにします。フィルターを使用すれば、システム内のオブジェクトの表示方法を容易に編成できます。フィルターを拡張することでリストにアクセスするとき、ネイティブ・ファイル・システムのオブジェクト (ライブラリーやファイルなど) が表示され、そのリモート・オブジェクトに対してアクションを実行できます。

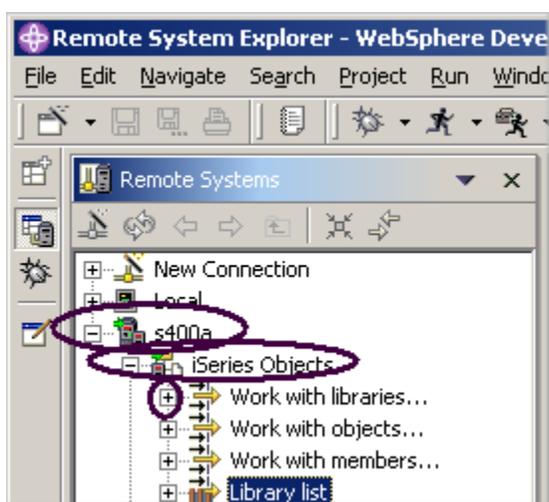


図87: 「ライブラリーの操作」の拡張

まず、直接操作するライブラリーを指定する必要があります。

1. 「リモート・システム」ビューで、iSeries システムへの接続を、その接続の横にあるプラス符号 (+) をクリックして拡張します。
2. 「iSeries オブジェクト」を拡張します。  
新規ライブラリー・フィルターを作成するには、以下のようにします。
3. 「ライブラリーの操作」を拡張します。(また、「iSeries オブジェクト」を右クリックし、「新規」>「ライブラリー・フィルター」を選択することもできます。)

「ライブラリーの操作」の拡張は WRKLIBPDM コマンドに相当し、これにより「リモート・システム」ビューにフィルターが作成および拡張されます。次のウィンドウがオープンします。

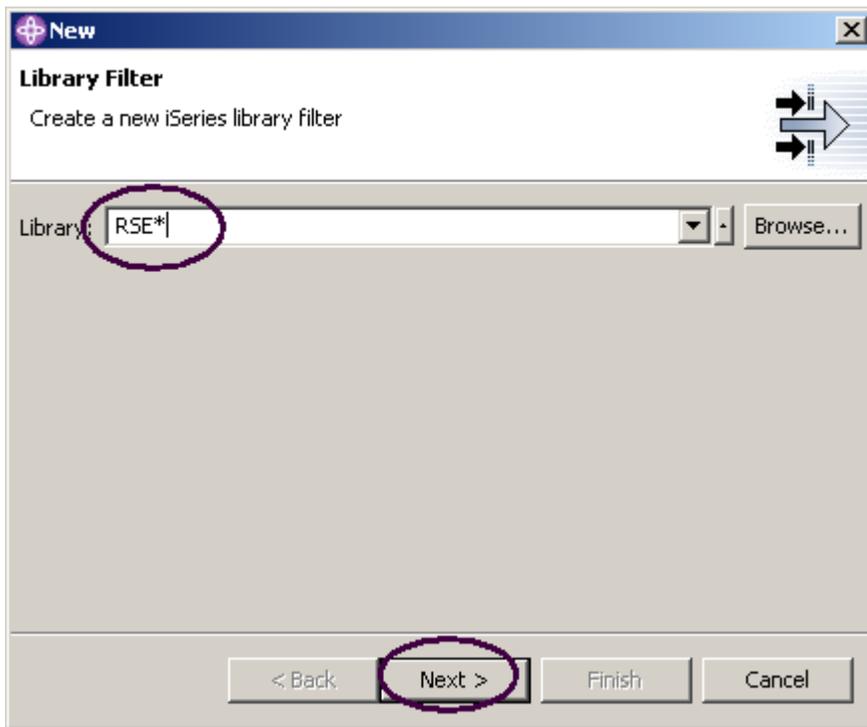


図 88: フィルター・ストリングの指定

フィルターを作成し、操作するライブラリーを指定して、iSeries オブジェクトが表示されるようにします。iSeries にあるすべてのライブラリーを表示するフィルターを、**RSExxxxxx** および **VARxxxxxxx** という名前で作成します。xxx は任意の文字です。

**注:** 上記の名前のライブラリーが存在しない場合、システム上に表示される別のライブラリーを選択する必要があります。

リモート・システム・エクスプローラーで始まるライブラリーを選択するフィルター・ストリングを指定します。

1. 「\*」ワイルド・カード文字を使用して、「**ライブラリー**」フィールドに「RSE\*」と入力します。
2. 「次へ」をクリックします。  
次のウィンドウが表示されます。

**New**

**Library Filter**

Name the new filter

Filters are saved for easy re-use. Specify a unique name for this filter. This name will appear in the Remote Systems view, and will be expandable.

Filter name: All RSE and VARPG libraries

< Back   Next >   Finish   Cancel

このフィルターに名前を指定します。

3. 「フィルター名」フィールドに、「すべての RSE および VARPG ライブラリー」と入力します。

リモート・システム・エクスプローラーはフィルターを後で使用するために保管するので、フィルターに名前を付けてください。これとは反対に、PDM はフィルターを保管しません。

4. 「終了」をクリックします。

「リモート・システム」ビューに戻ると、「iSeries オブジェクト」の下に新しいフィルターが拡張され、すべての RSE\* ライブラリーがリストされています。次に、VARPG ライブラリーを追加する必要があります。

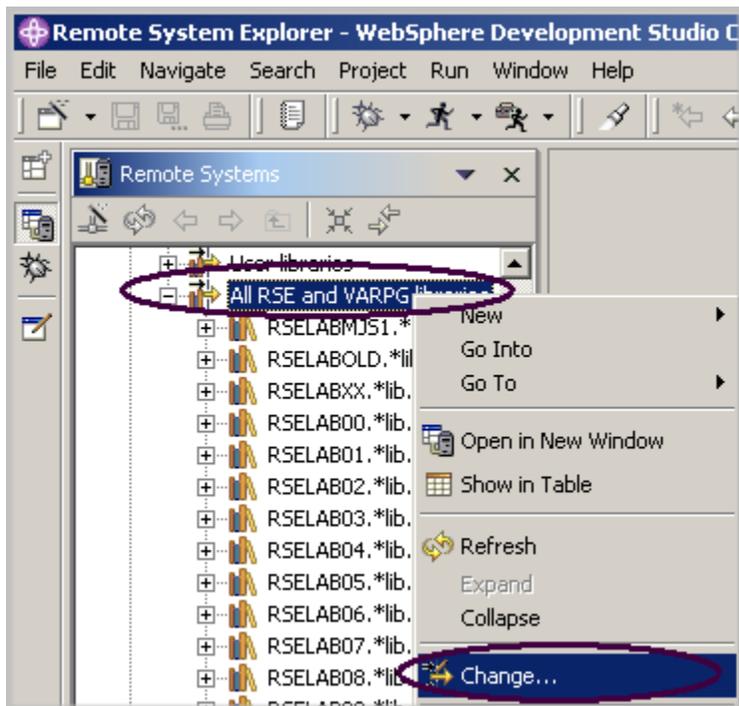


図 89: 「変更」オプションの選択

5. フィルター「すべての **RSE** および **VARPG** ライブラリー」を右クリックし、「変更」を選択します。

「ライブラリー・フィルターの変更」ウィンドウが表示されます。

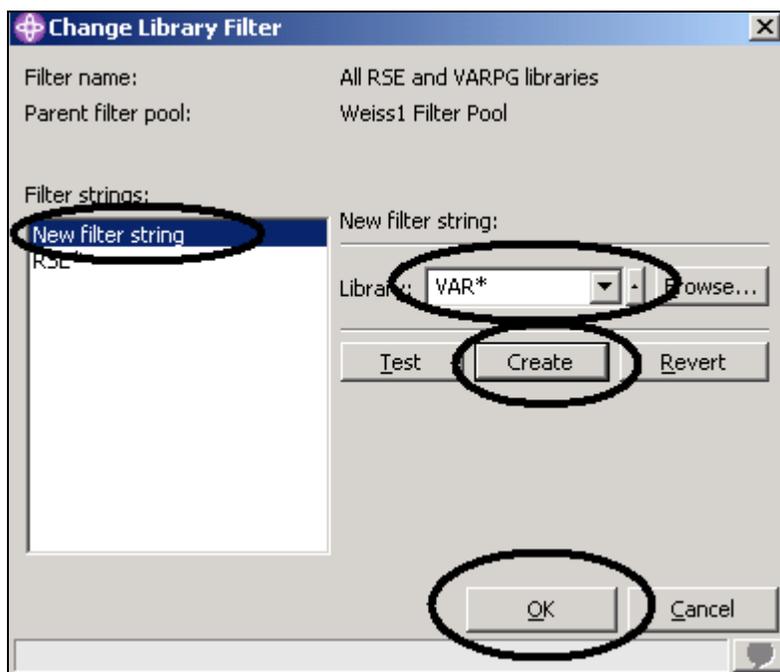


図 90: 別のフィルター・ストリングの追加

6. リスト・ボックスで「**新規フィルター・ストリング**」をクリックします。
7. 「**ライブラリー**」フィールドに「**VAR\***」と入力します。
8. 「**作成**」をクリックします。  
「**VAR\***」フィルター・ストリングがリスト・ボックスに追加されます。
9. 「**OK**」をクリックします。  
「**リモート・システム**」ビューに戻ります。拡張されたリストに、作成したフィルターが含まれています。

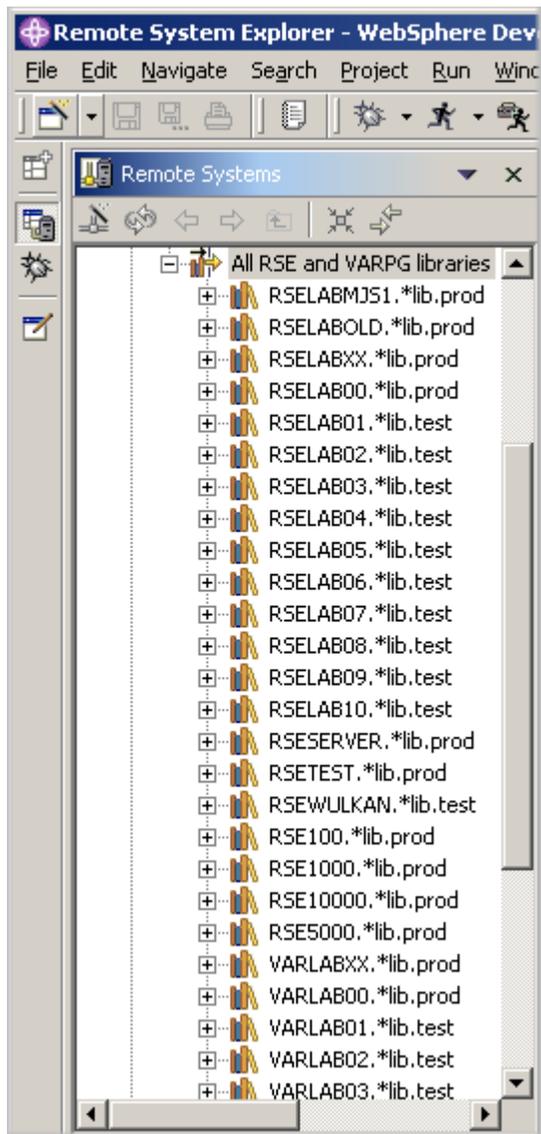


図 91: 拡張されたフィルター

この時点で、ライブラリーを直接操作したり、直接操作したいオブジェクトへドリルダウンしたりすることができます。

次に、オブジェクト・フィルターを作成します。オブジェクト・フィルターは、「リモート・システム」ビューに iSeries ホストからのオブジェクト・セットをリストします。

### オブジェクト・フィルターの作成

1. 「リモート・システム」ビューで接続を拡張し、次に「iSeries オブジェクト」を拡張します。

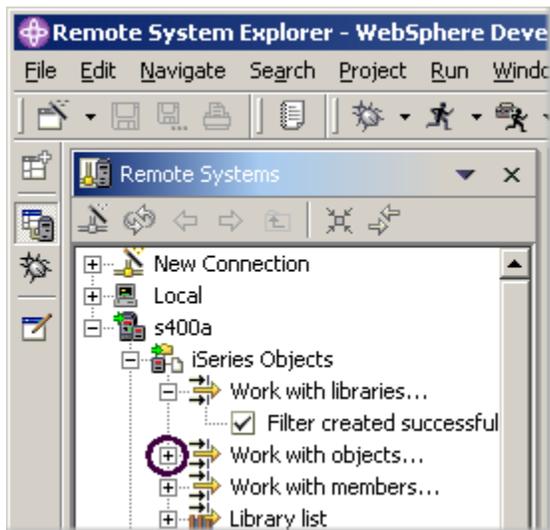


図 92: オブジェクト・フィルターの作成

2. 「オブジェクトの操作」を拡張します。「**iSeries オブジェクト**」を右クリックして、「**新規**」 > 「**オブジェクト・フィルター**」を選択することもできます。

注: 「オブジェクトの操作」の拡張は、WRKOBJPDM コマンドに相当します。

これにより、「**新規オブジェクト・フィルター**」ウィンドウが表示されます。

ここで、フィルターを作成して、**RSELABxx** ライブラリーにあるすべてのソース・ファイルを表示します。

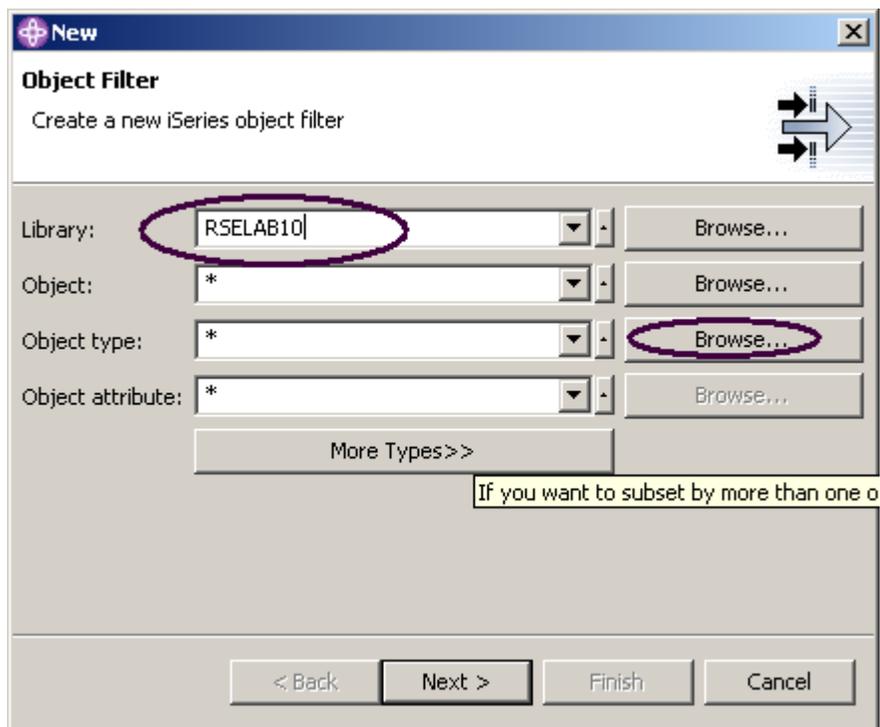


図 93: フィルター・ストリングの指定

3. 「ライブラリー」フィールドに「RSELABXX」と入力します。
4. 「オブジェクト・タイプ」フィールドの横にある「参照」をクリックします。

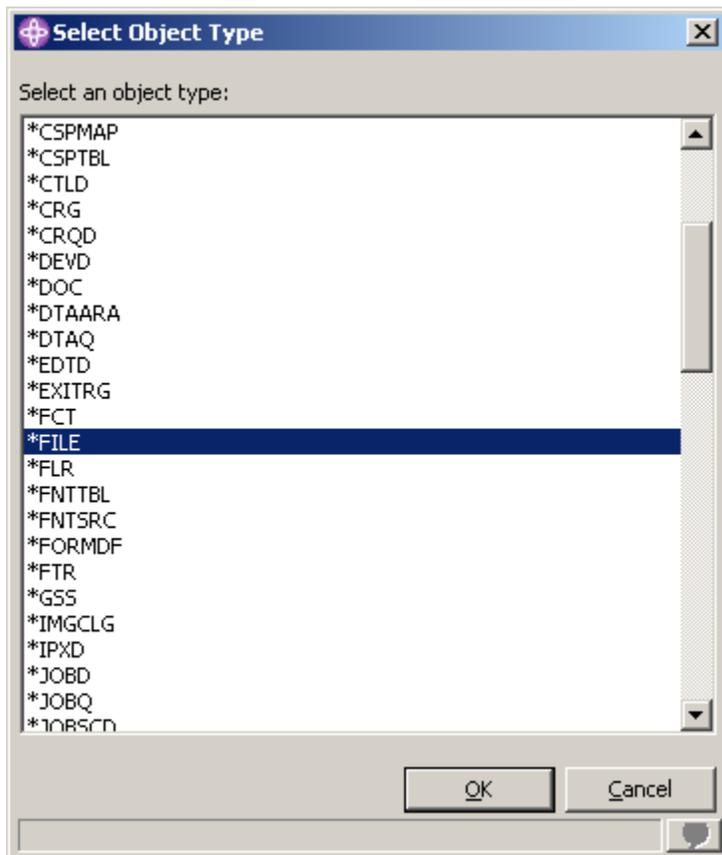


図94: オブジェクト・タイプの選択

5. リストから「**\*File**」オブジェクト・タイプを選択します。
6. 「**OK**」をクリックします。  
「オブジェクト・フィルター」ウィンドウに戻ります。

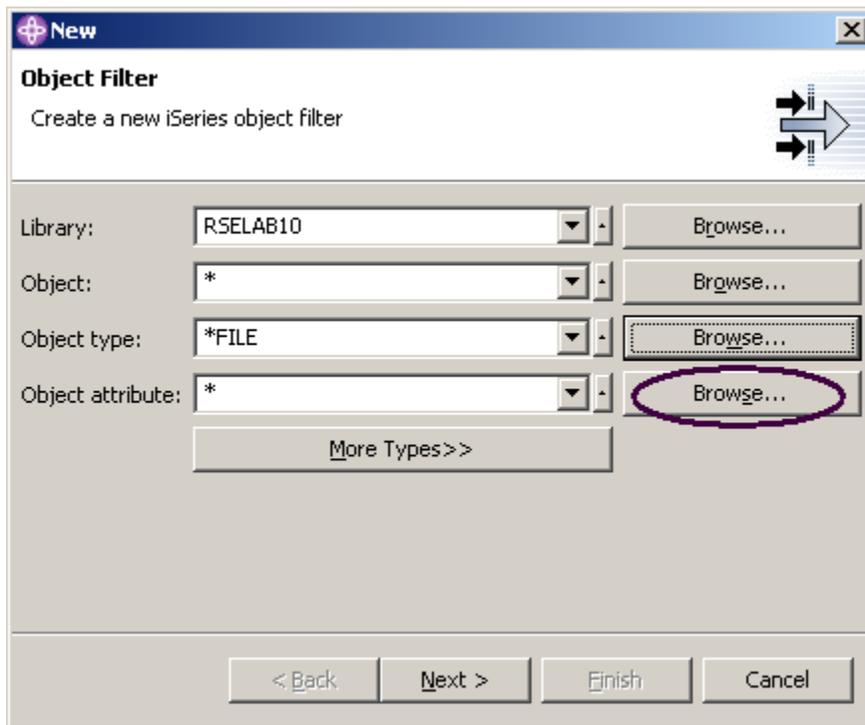


図 95: オブジェクト属性の参照

7. 「オブジェクト属性」フィールドの横にある「参照」をクリックします。

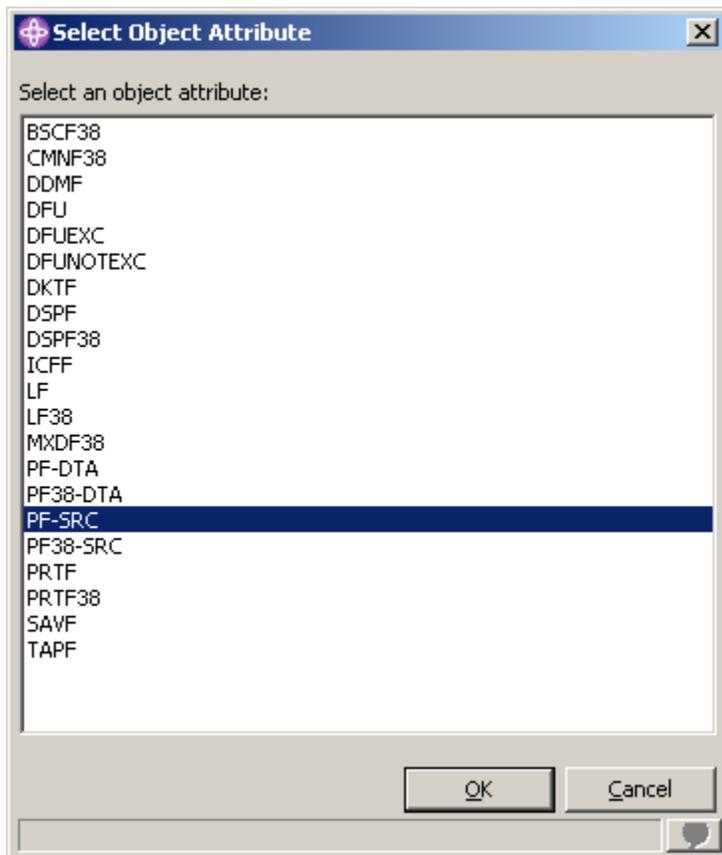


図 96: フィルター・オブジェクト属性の選択

8. オブジェクト属性リストから「**PF-SRC**」を選択します。
9. 「**OK**」をクリックします。
10. 「オブジェクト・フィルター」ウィンドウで「次へ」をクリックします。

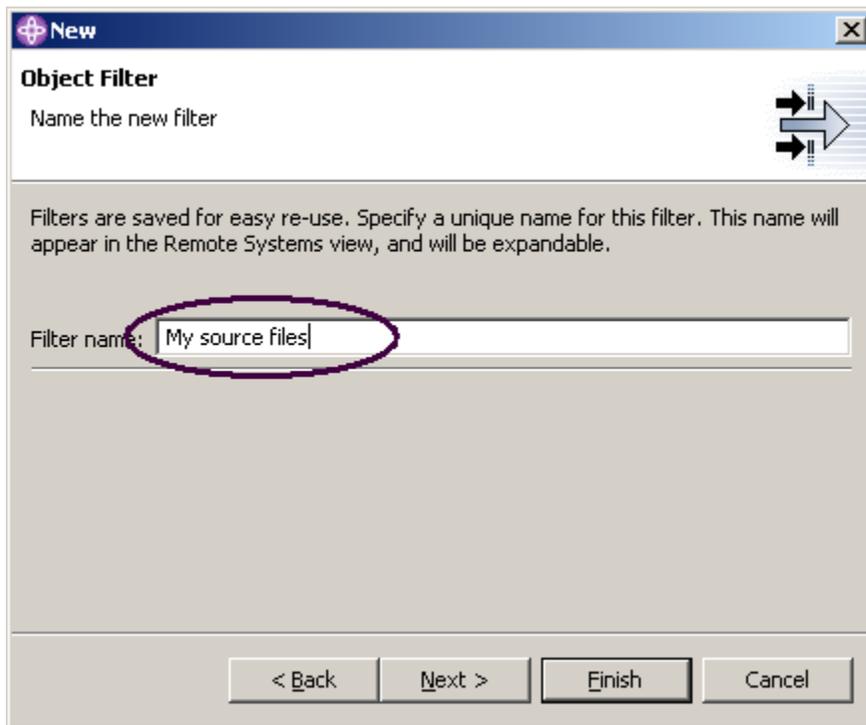


図97: フィルター名の指定

11. 「フィルター名」に「My source files」と指定します。
12. 「終了」をクリックします。

**注:** リモート・システム・エクスプローラーはフィルターを後で使用するために保管するので、フィルターに名前を付けてください。これとは反対に、PDM はフィルターを保管しません。

「リモート・システム」ビューの「iSeries オブジェクト」の下に新規オブジェクト・フィルターが表示されます。

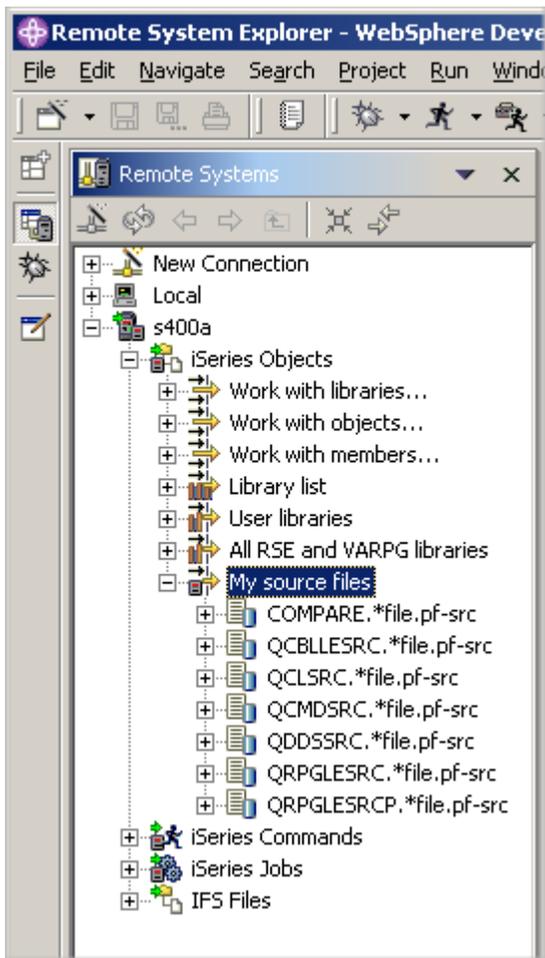


図98: オブジェクト・フィルターによるライブラリー内の全ソース・ファイルの表示

フィルターの作成方法および開発環境の調整方法について学びました。フィルターは、非 iSeries サーバーやローカル・システムにも指定できます。

**ヒント:** しばらくの間、リモート・システム・エクスプローラーを使用していた場合、ワークスペースに含まれるフィルターが多すぎて、容易にナビゲートできない場合があります。この場合、フィルターはいくつかのプールに区分され、1つのプールに多くの異なるエレメントのフィルターが含まれた可能性があります。たとえば、あるフィルター・プールに給与計算プログラムのフィルターが含まれ、別のフィルター・プールに売掛管理プログラムのフィルターが含まれる場合があります。

これで、PDM でライブラリー、オブジェクト、またはメンバーを操作したように、「リモート・システム」ビューにあるオブジェクトを操作できます。

QRPGLESRC 内のメンバー PAYROLL を編集したい場合は、以下のようにします。

1. 「**QRPGLESRC**」を拡張します。
2. メンバー「**PAYROLL**」を右クリックします。
3. 「アプリケーションからオープン」>「リモート・システム **LPEX** エディター」を選択します。

これにより、ソース・メンバーがダウンロードされ、このメンバーでエディターがオープンされます。

メンバーを編集した後、このメンバーでポップアップ・メニュー・オプションを使用して、「リモート・システム」ビューからメンバーを保管およびコンパイルできます。

また、デフォルトのアクションに加え、独自のアクションも作成できます。

### ユーザー・アクションの作成

PDM では、事前に用意されているシステム・アクションの他に、ユーザー・アクションも作成できます。リモート・システム・エクスプローラーでも、同様のことができます。アクションとは何でしょうか。アクションとは、「ユーザー・アクションの操作」ウィンドウで定義されるホスト・コマンドのことであり、**iSeries** ライブラリー、オブジェクト、ジョブ、およびメンバーに対して実行されるものです。アクションは、リモートの **UNIX** システム、**Windows** システム、**Linux** システム、ローカル・システム、または **IFS** システム内のフォルダーおよびファイルに対しても定義できます。

ユーザー・アクションを作成するには、以下のようにします。

1. **iSeries** 接続を拡張し、「**iSeries オブジェクト**」を拡張します。

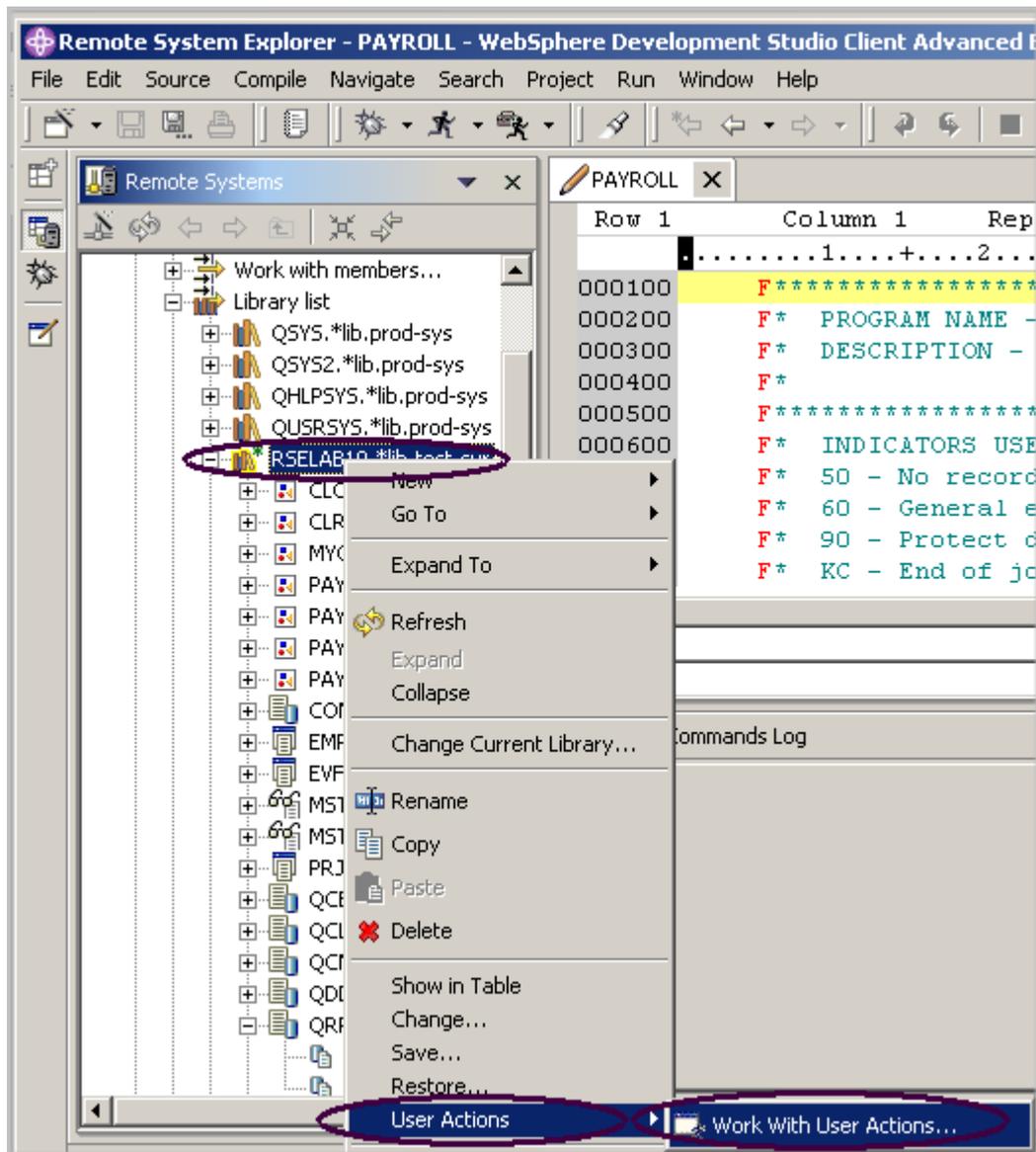


図 99: ユーザー・アクションの操作

2. 「ライブラリー・リスト」フィルターを拡張します。
3. 「RSELABxx」を右クリックします。
4. 「ユーザー・アクション」 > 「ユーザー・アクションの操作」を選択します。  
「ユーザー・アクションの操作」ウィンドウが表示されます。

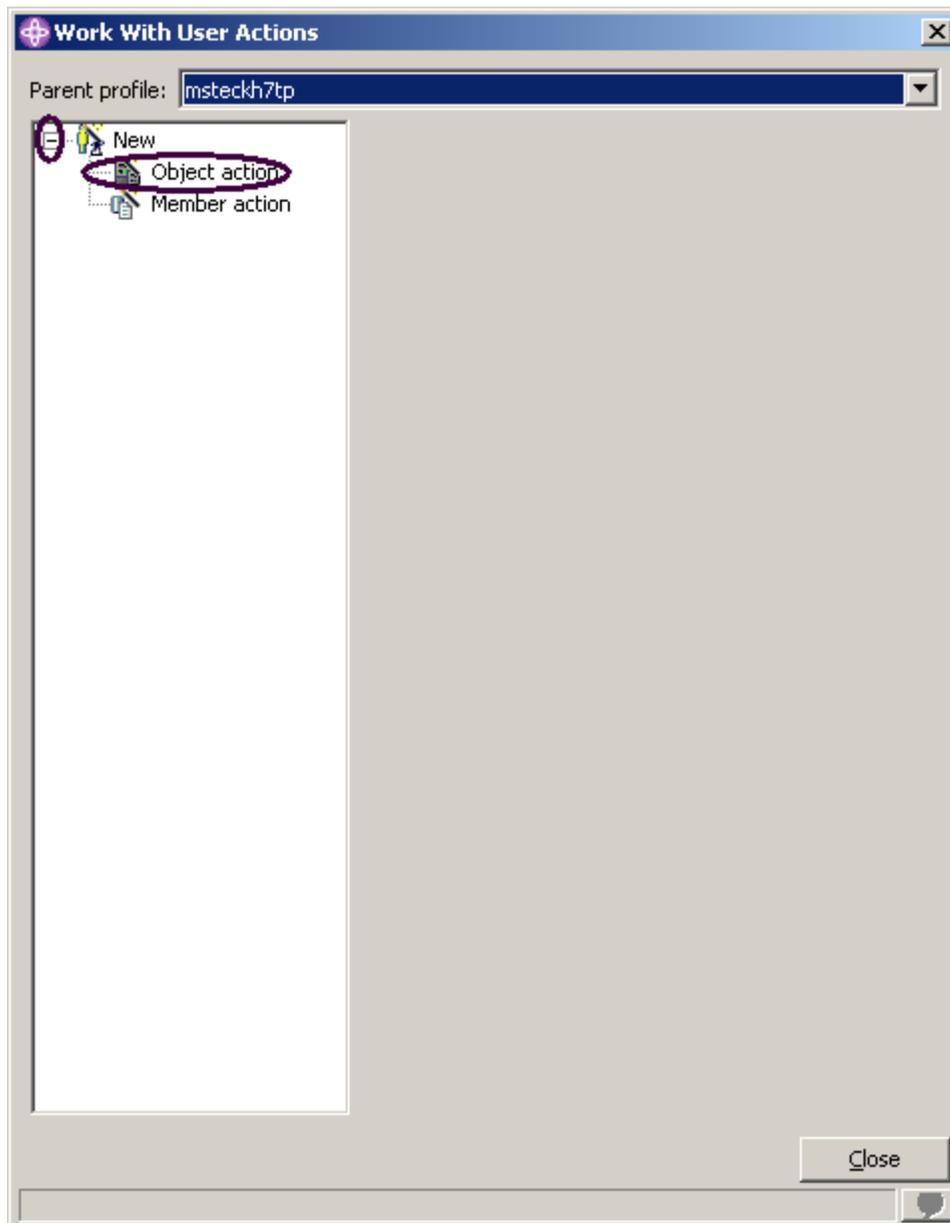


図100: 「ユーザー・アクションの操作」 ウィンドウ

5. リスト内の「新規」を拡張します (まだ拡張されていない場合)。
  6. 「オブジェクト・アクション」をクリックします。
- 「ユーザー・アクションの操作」ウィンドウが表示されます。

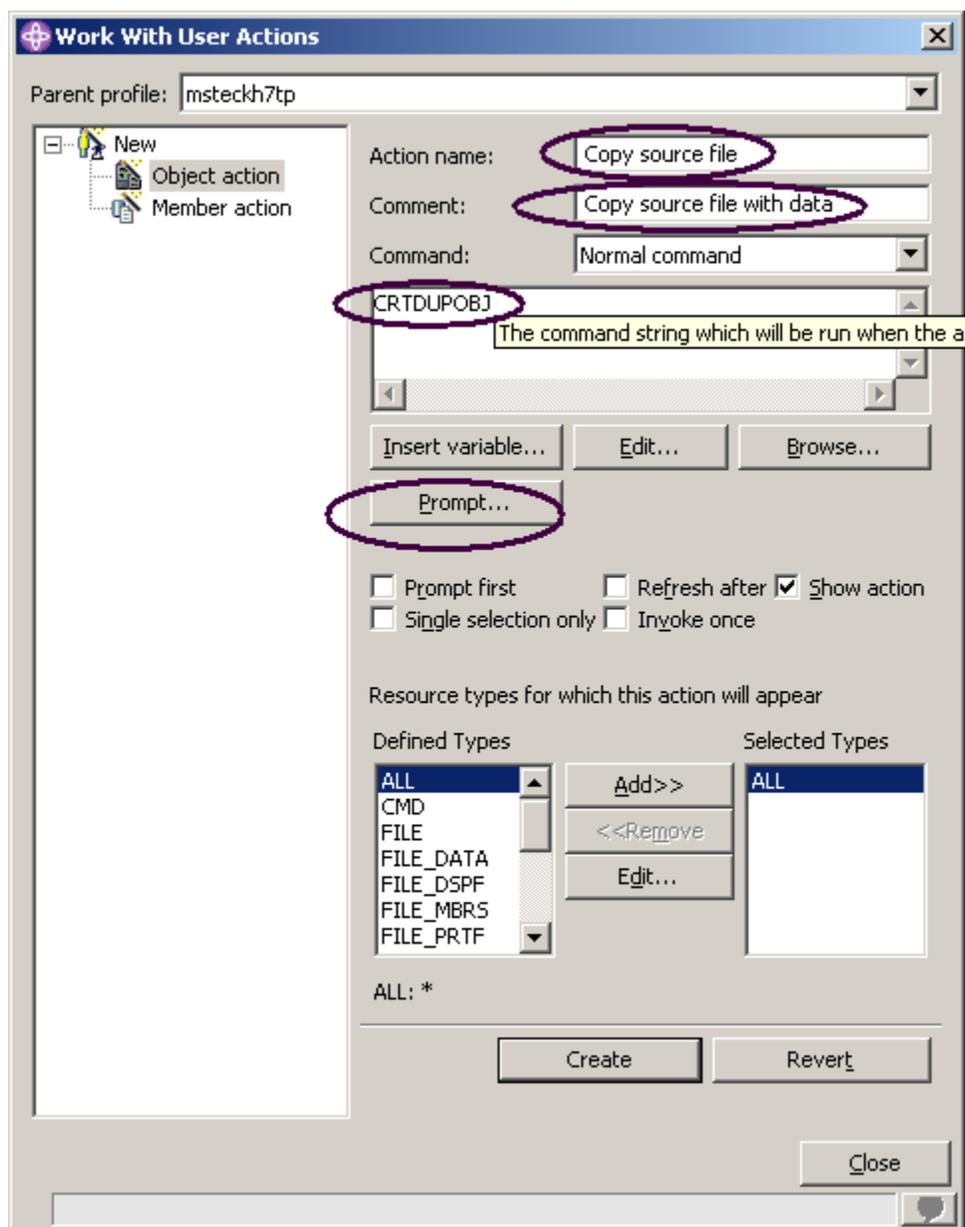


図101: オブジェクト・アクションの作成

ここで、データのあるソース・ファイルを、同じライブラリー内の新規ソース・ファイル QJUNKSRC にコピーするユーザー・アクションを作成します。

- ユーザー・アクションの名前「Copy source file」を「アクション」フィールドに入力します。

8. 「コメント」フィールドにコメントを入力します。
9. 「コマンド」に「CRTDUPOBJ」と入力します。
10. 「プロンプト」をクリックして、このコマンドのコマンド・プロンプターを表示します。

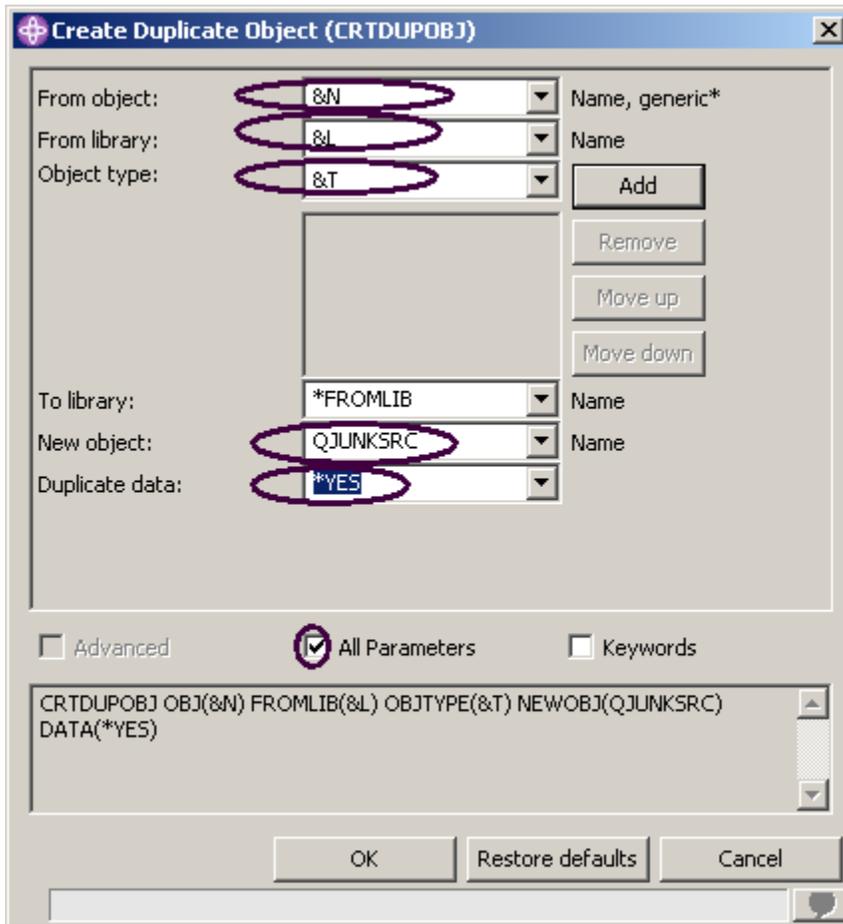


図 102: CRTDUPOBJ コマンドのプロンプト

次のコマンドを実行します。

```
CRTDUPOBJ OBJ(&N) FROMLIB(&L) OBJTYPE(&T) NEWOBJ(QJUNKSRC) DATA(*YES)
```

11. 「元オブジェクト」フィールドに「&N」（「リモート・システム」ビューで選択されたオブジェクトの名前を使用することを示す）と入力します。
12. 「元ライブラリー」フィールドに「&L」（選択されたオブジェクトからライブラリー名を取得する）と入力します。

13. 「オブジェクト・タイプ」フィールドに「&T」(選択されたオブジェクトからオブジェクト・タイプを取得する)と入力します。
14. 「新規オブジェクト」フィールドに、「QJUNKSRC」と入力します。  
他の「重複データ」パラメーターを参照するには、以下のようにします。
15. 「すべてのパラメーター」チェック・ボックスをクリックします。  
ここで、「重複データ」パラメーターがプロンプト・ウィンドウにも表示されます。
16. 「重複データ」パラメーターのプルダウン・メニューから「\*Yes」を選択します。
17. 「OK」をクリックします。  
「ユーザー・アクションの操作」ウィンドウに戻ります。  
アクションが実行された後に「リモート・システム」ビューをリフレッシュするには、次のようにします。
18. 「後でリフレッシュ」チェック・ボックスを選択します。  
ヒント: 「変数の挿入」を押して、有効な置換変数とその説明のリストを表示します。

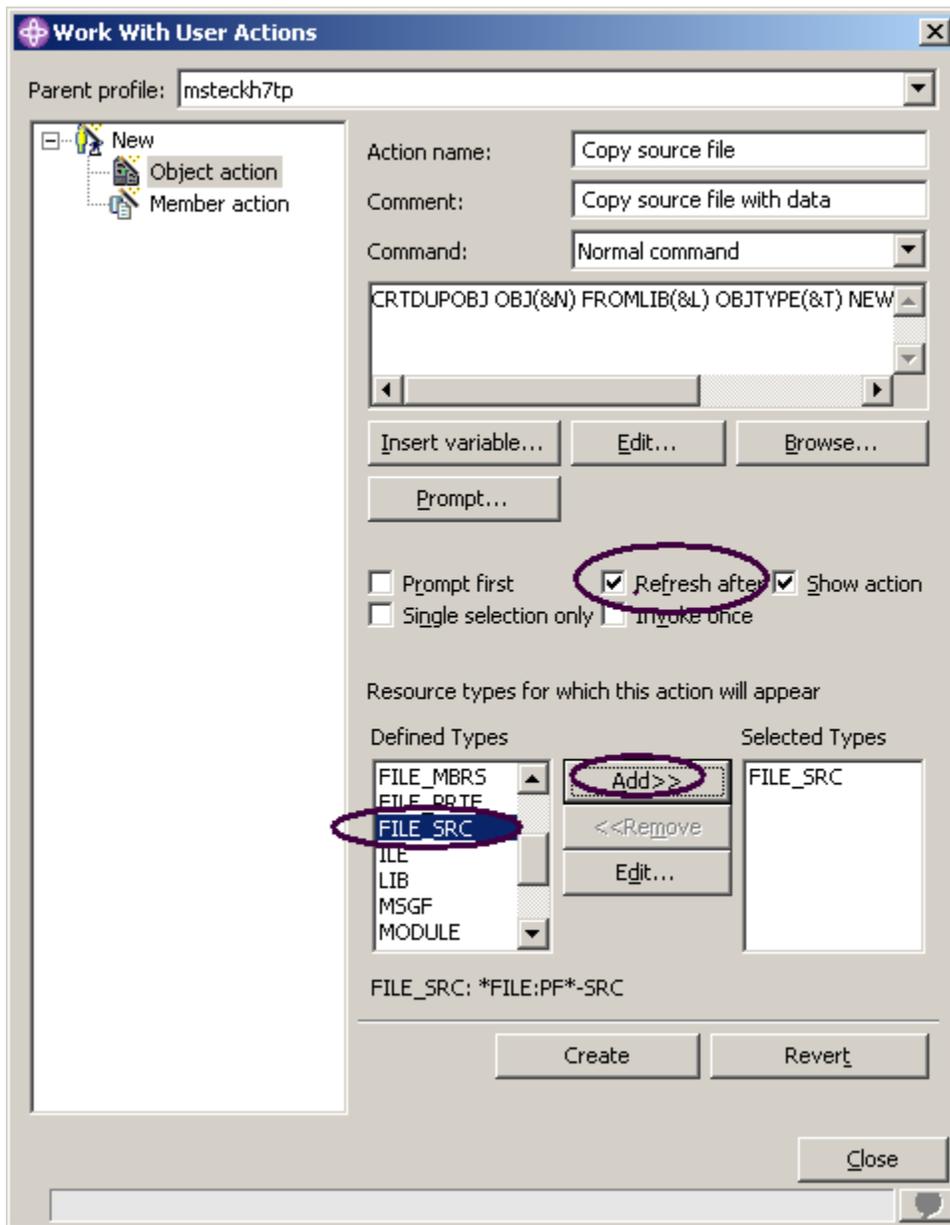


図103: ソース・ファイルに対するユーザー・アクションの制限

このユーザー・アクションは、ソース物理ファイルについてのみ有効です。ソース物理ファイルを右クリックしたときのみ、このユーザー・アクションがポップアップ・メニューに表示されるように、この制限を指定する必要があります。

この制限を指定するには、以下のようにします。

19. 「定義済みタイプ」ボックスに移動します。

20. リストで「**FILE\_SRC**」を選択します。

21. リストの横にある「追加」をクリックします。

FILE\_SRC が、選択されたタイプの 1 つになりました。実際には、これしか選択していないため、これが唯一のタイプとなります。

22. 「作成」をクリックし、次に「クローズ」をクリックします。

これで、ソース・ファイルを右クリックしたときのみ、選択されたポップアップ・メニューにこのユーザー・アクションが表示されます。それ以外のオブジェクト・タイプの場合、このユーザー・アクションは表示されません。

ワークベンチとリモート・システム・エクスプローラー・パースペクティブに戻り、試してみましょう。**注:**すでに PAYROLL メンバーをオープンしている場合は、忘れずにクローズしてください。

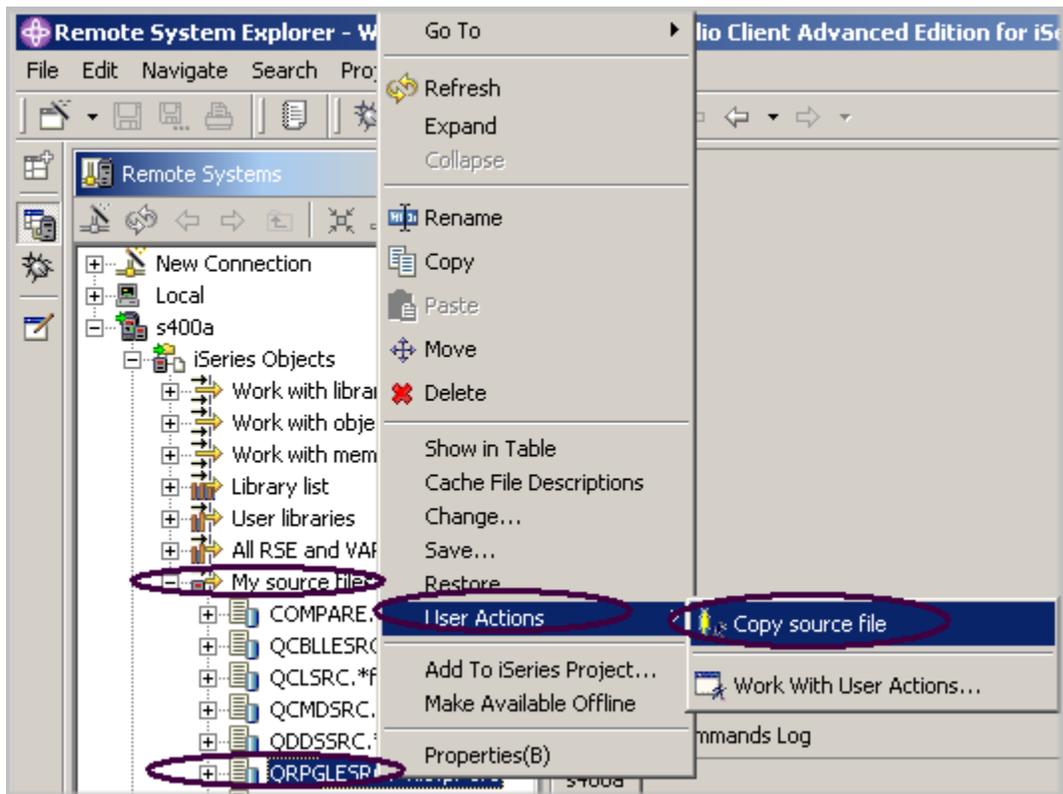


図104: ポップアップ・メニューに表示されたユーザー・アクション

23. フィルター「**My Source files**」を見つけます。
24. そのフィルターを拡張します (まだ拡張されていない場合)。
25. 「**QRPGLSRC**」ファイルを右クリックします。
26. 「ユーザー・アクション」>「ソース・ファイルのコピー」を選択します。

ファイルが複写され、リストがリフレッシュされます。新規のソース・ファイルがリストに表示されます。

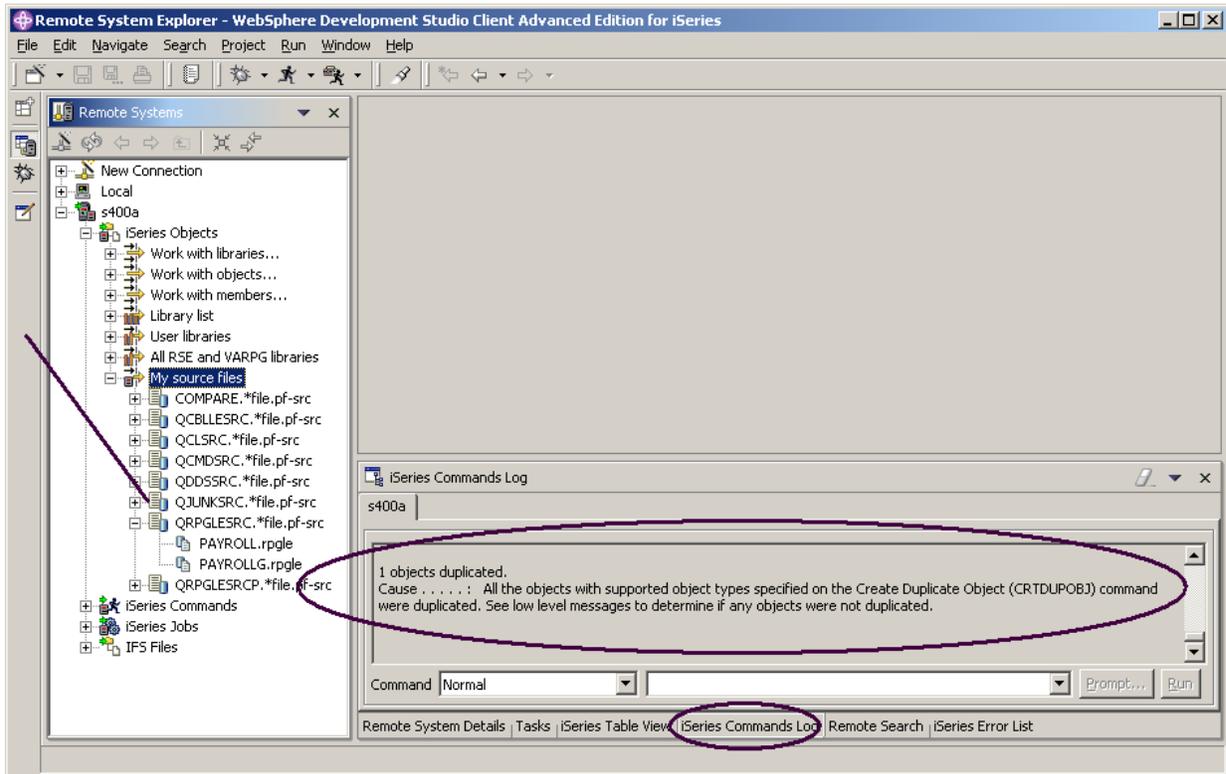


図 105: 正常に実行されたユーザー・アクションの CRTDUPOBJ コマンド

ワークベンチの右下のペインでコマンド・ログを参照すれば、リモート・システム・エクスプローラー・ジョブで実行している CL コマンドのメッセージを確認できます。

作成したソース・ファイル **QJUNKSRC** を削除するには、以下のようにします。

27. 「**QJUNKSRC**」を右クリックします。
28. ポップアップ・メニューから「削除」を選択します。

### リモート・システム・エクスプローラーからのコマンドの実行

コマンド入力は、「リモート・システム」ビューの一部です。

1. 前のタスクでコマンド・ログが表示されていた右下のペインに、「**iSeries テーブル・ビュー**」タブがあるかどうかを確認します。
2. そのタブがあれば、クリックします。
3. そのタブがない場合は、以下のようにします。
  - a) 「リモート・システム」ビューで「**My source files**」フィルターを右クリックします。
  - b) ポップアップ・メニューから「**テーブルへの表示**」を選択します。

iSeries テーブル・ビューで、テーブルの接続先となる iSeries サーバーでコマンドを実行することができます。iSeries テーブル・ビューの下にある「コマンド」フィールドでコマンドを実行し、「メッセージ」フィールドにメッセージを表示できます。テーブルを取り込んだ後に、コマンドを入力し、「プロンプト」を選択してパラメーターを指定してから「実行」を選択するか、または単に「実行」を選択することができます(後者の場合は、プロンプトが出され、通常どおりコマンドが実行されます)。コマンドが実行されると、そのコマンドからのメッセージが「メッセージ」ドロップダウン・フィールドに取り込まれます。メッセージを選択すると、「詳細」ボタンが使用可能になります。このボタンを押すと、メッセージとそのヘルプが表示されます。

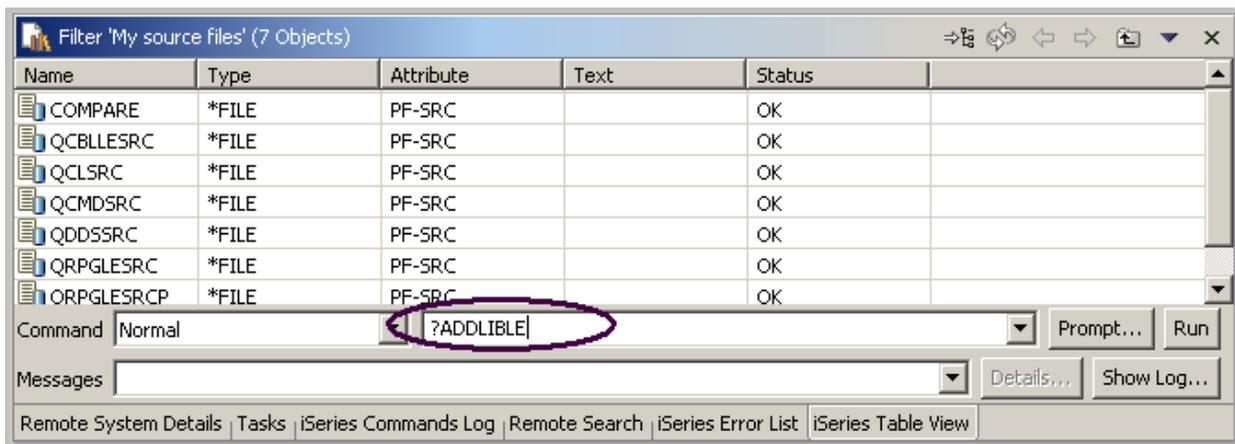


図106: テーブル・ビューとコマンド入力

「コマンド」フィールドで、次のようにします。

- たとえば「?ADDLIBLE」を iSeries コマンドとして入力します。

疑問符 (?) は、プロンプト画面を表示するためにあります。

**ヒント:** 疑問符 (?) を指定する代わりに、「プロンプト」プッシュボタンを使用することもできます。

- 「実行」をクリックします。

ADDLIBLE コマンドのコマンド・プロンプト・ウィンドウが表示されます。

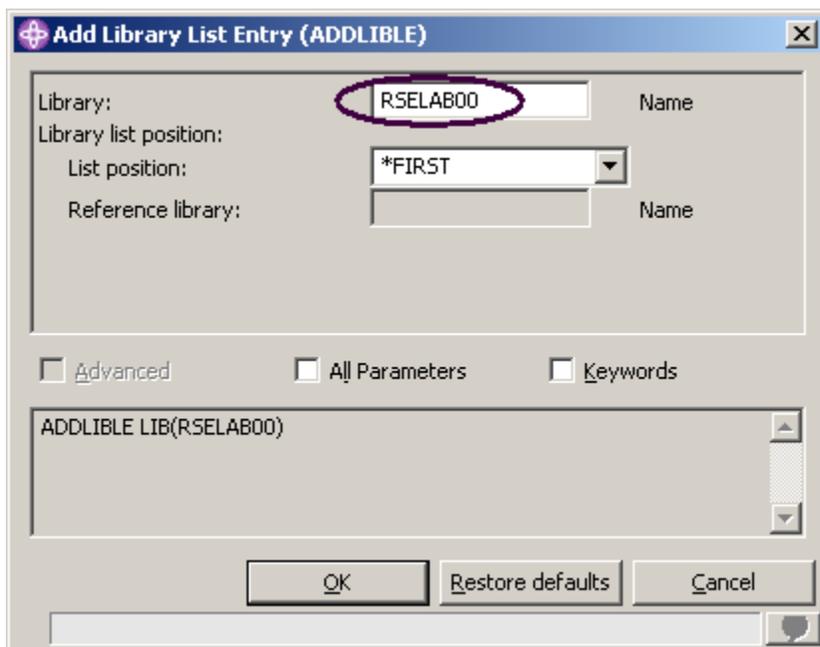


図 107: コマンド・プロンプト・ウィンドウ

プロンプト・ウィンドウで、次のようにします。

6. 「RSELAB00」と入力します。このライブラリーが、iSeries サーバー上のリモート・システム・エクスプローラー・ジョブのライブラリー・リストに追加されます。**注:** ライブラリー RSELAB00 が iSeries システム上に存在しない場合、別のライブラリーを追加しなければならない場合があります。

「メッセージ」フィールドで、このコマンドが正常に完了したことを確認します。

グリーン・スクリーンの F9 と同様に、コマンド履歴に移動するには、「コマンド」フィールドにある下矢印を押します。

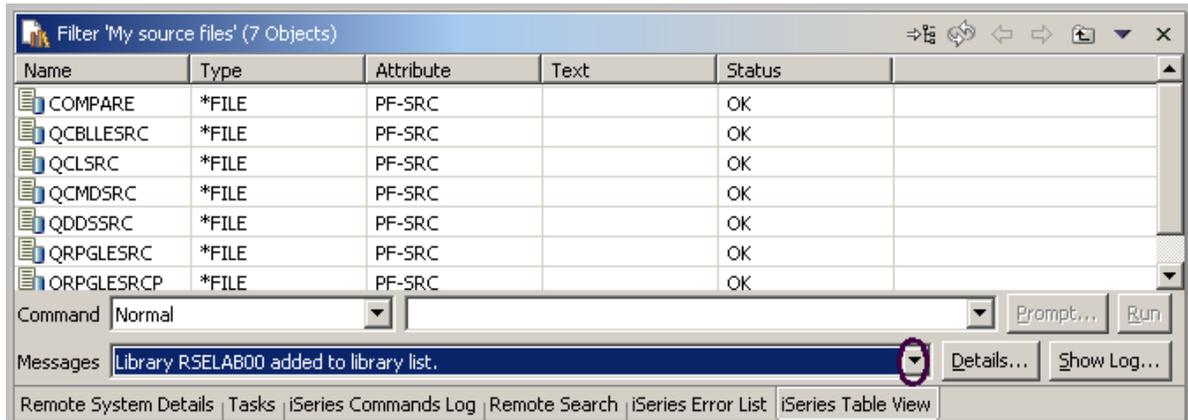


図 108: コマンド履歴

「リモート・システム」ビューで「iSeries オブジェクト」サブシステムの下にある「iSeries コマンド」サブシステムを使用して、事前に定義されているコマンドを実行したり、独自のコマンドを定義したりすることもできます。

この練習問題の内容を十分理解したかどうかを判断するために、以下のチェックポイントを確認してください。

## チェックポイント

- リモート・システム・エクスプローラーは、何を置き換えるものですか。
  - PDM
  - SDA
  - CGU
  - DFU
- フィルターが行う内容:
  - iSeries システムにあるオブジェクトを照会する。
  - システム内のオブジェクトの表示方法を容易に編成する。
  - ネイティブ・ファイル・システム・オブジェクト (QSYS) を表示する。
  - 上記のリモート・オブジェクトに対してアクションを実行できるようにする。
  - 上記すべて
- 「オブジェクトの操作」の拡張は、どれに相当しますか。
  - WKRLBPDM
  - WKROBJPDM
  - WRKMBRPDM
  - 上記すべて

4. リモート・システム・エクスプローラーはフィルターを後で使用するために保管するので、フィルターに名前を付けてください。このことは、フィルターを保管しない PDM とは異なります。(T, F)
5. 作成できるもの:
  - A. ライブラリー・フィルター
  - B. ライブラリー・リスト・フィルター
  - C. オブジェクト・フィルター
  - D. メンバー・フィルター
  - E. ジョブ・フィルター
  - F. 上記すべて
6. PDM では、事前に用意されているシステム・アクションの他に、ユーザー・アクションも作成できます。リモート・システム・エクスプローラーでも、同様のことができます。(T, F)
7. iSeries テーブル・ビューの接続先である iSeries サーバーでコマンドを実行することができます。(T, F)

## 実習

iSeries システムに独自のライブラリーを作成しました。そこで、メンバー・フィルターおよびジョブ・フィルターを作成します。次に、ライブラリー・リスト内でライブラリーを上下に移動します。最後に、フィルター・プールを作成します。タスクの途中で困ったときは、**Development Studio Client for iSeries** のオンライン・ヘルプを使用してください。

## ここで学んだ内容

この練習問題では、リモート・システム・エクスプローラー・パースペクティブを使用して、前の練習問題で使用した iSeries オブジェクトを操作しました。また、アクションの実行、および独自アクションの定義がいかに簡単であるかについても学びました。つまり、リモート・システム・エクスプローラーはどのようにしてユーザーの作業を編成および統合し、どのようにしてその作業をより容易にするのかを確認しました。



## iSeries アプリケーションの保守: リモート・システム・エクスプローラーのチュートリアル要約の紹介

新規または従来の iSeries アプリケーションのビルドをこれまで以上に生産性あるものにするのがいかに簡単かを確認します。リモート・システム・エクスプローラーにより、OS/400 リソースへ PDM のようにアクセスできます。PDM に似たテーブル・ビューを使用するということは、オブジェクトを右クリックして、PDM と同様のアクションを実行し、PDM のようにテーブルの下部にあるコマンド行を使用できるということです。すべての PDM 置換変数をサポートするユーザー定義アクションがありました。もちろん、リモート・システム・エクスプローラーと緊密に統合されたタスクを編集、検査、コンパイル、実行、およびデバッグするアクションもありました。

RPG から Java に至るまで、すべての開発作業に一貫した経験技術を提供する、高度に統合された実働環境を紹介しました。この新世代のツールは、ファイル・システムの探査、コンパイル、ビルド、編集、実行、およびデバッグのためのさまざまなサポートを提供しました。この新規ツールを使用して、非常に高度な編集や DDS 設計のサポートのためにクラシック・ツールを活用することができました。リモート・システム・エクスプローラーのツールは、クラシックでフィーチャー変更不可の ADTS ツールに対して機能のスーパーセットを提供しました。生産性と使用可能性の著しい向上、単独およびチームでの開発のサポート、および IBM とパートナーが提供するサーバー・アプリケーション開発ツールの緊密な統合のための共通ツールが提供されました。これらの新世代ツールを使用すれば、ユーザーの技術も目に見えて向上し、IBM WebFacing ツール、Web、Web サービス、Java、XML などの新しいプログラミング・モデルへの移行がいっそう容易になります。

ちなみに、ADTS はサーバー・サイドの iSeries アプリケーションを開発および保守するための従来のメソッドです。現在、iSeries RPG、COBOL、C、C++、CL、および DDS の開発用に、高度に統合された、拡張性の高い新規のツールを含む Development Studio Client が登場しました。これらの新しいツールは、Java、Web、Web サービス、および XML アプリケーションの開発経験に一貫性のある開発経験を提供し、すべてに関する学習期間を短縮して、これらの e-business アプリケーション・プログラミング・モデルへの移行をいっそう容易にします。

## 練習問題チェックポイントの答え

1	2	3	4	5	6	7	8
1. B	1. D	1. C	1. D	1. D	1. B	1. C	1. A
2. G	2. E	2. A	2. T	2. D	2. D	2. T	2. E
3. G	3. B	3. F	3. A	3. G	3. AC、	3. A	3. A
4. E	4. AA、	4. C	4. T	4. T	BB、	4. C	4. T
5. E	BD、	5. T	5. D	5. T	CA、	5. C	5. F
6. A	CB、	6. E	6. C	6. T	DD、	6. T	6. T
7. T	DC	7. E	7. B	7. F	EF、	7. F	7. T
8. A	5. E	8. C	8. C	8. A	FE、	8. T	
9. T		9. T	9. C	9. D	GG	9. E	
10. E		10. C	10. A	10. D	4. E	10. A	
11. B		11. B	11. D	11. D	5. T		
12. F		12. T	12. C	12. T	6. H		
13. E		13. D	13. T		7. T		
14. T			14. D		8. T		
15. T			15. E		9. T		
			16. D		10. T		
			17. D		11. T		
			18. B				
			19. C				
			20. T				
			21. T				

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032 東京都港区六本木 3 丁目 2-31  
IBM World Trade Asia Corporation  
Intellectual Property Law & Licensing

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。**

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください

IBM Canada Ltd. Laboratory  
Information Development  
B3/KB7/8200/MKM  
8200 Warden Avenue  
Markham, Ontario  
Canada L6G 1C7

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

## プログラミング・インターフェース情報

本書の目的は、クライアント/サーバー環境のワークステーションでアプリケーションおよびユーザー・インターフェースを作成および管理する手助けをすることです。本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。本書は、IBM WebSphere Development Studio Client for iSeries によって提供された一般使用プログラミング・インターフェースおよび関連情報を文書化したものです。

## 商標

以下は、IBM Corporation の商標です。

Application System/400  
AS/400  
AS/400e  
DB2/400  
e-business  
IBM  
iSeries  
Integrated Language Environment  
OS/400  
RPG/400  
VisualAge  
WebSphere

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Active X、Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。