

WebSphere Development Studio Client: Java Development Tools

George Farr
IBM Toronto Laboratory

ITSO iSeries Technical Forum
SP17

WDS for iSeries

New World

New Servers

New Tools

Disclaimer

Acknowledgment:

- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
 - ▶ Don Yantzi, Phil Coulthard, George Farr, Claus Weiss, David Slater, Alison Butteril, Linda Cole

Disclaimer:

- The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Reproduction:

- The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

Agenda



- ▶ Getting Started
 - **Creating a HelloWorld program**
- ▶ The Java Editor
- ▶ Java Views
- ▶ Compile / Run / Debug
- ▶ iSeries Additions
- ▶ Reference



Java Development Tools



iSeries AD, IBM Toronto

▶ Quick Overview

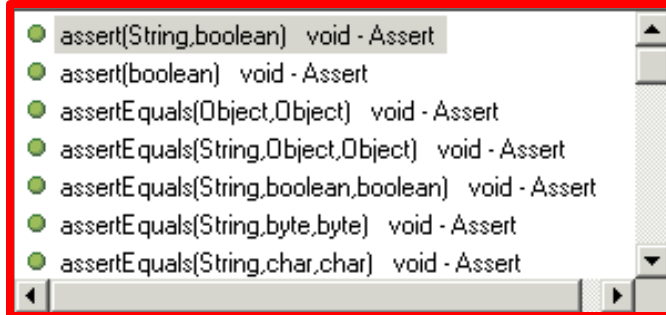


- ✓ Java Development Tools (JDT):
 - **Choice of Java level**
 - ▶ **IBM JRE 1.3 built-in and offered as default**
 - **Java projects**
 - ▶ **Optimized for Java development**
 - ▶ **Specialized views**
 - ✓ Package view and Hierarchy view;
 - ✓ Outline view: many tools launchable from here!
 - ▶ **Java Editor**
 - ✓ Content Assist: Ctrl+Space bar to get list of classes or methods
 - ✓ JavaDoc shown as hover help
 - ✓ F3 to open selected class
 - ▶ **Refactoring support**
 - ✓ Rename updates all references
 - ▶ **Incremental compile**
 - ✓ When Ctrl+S pressed in editor
 - ✓ Or on demand via Build Project action



✓ Java Development Tools (JDT):

• Choice of Java level

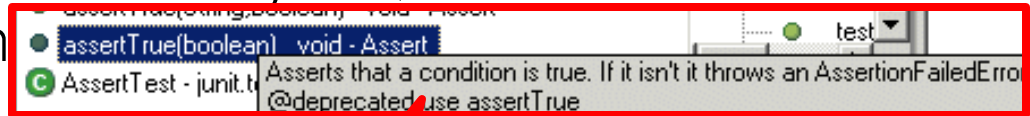


in and offered as default

Java development

✓ Package view and Hierarchy view;

✓ Outline view: m



▶ Java Editor

✓ Content Assist: Ctrl+Space bar to get list of classes or methods

✓ JavaDoc shown as hover help

✓ F3 to open selected class

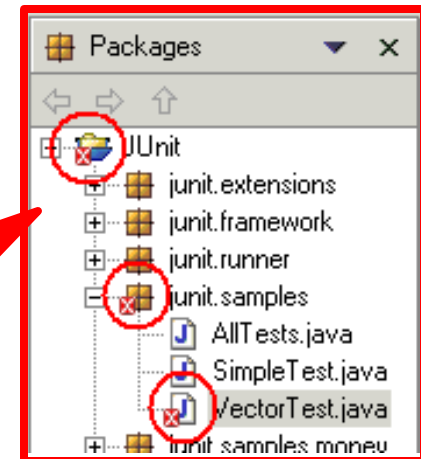
▶ Refactoring support

✓ Rename updates all references

▶ Incremental compile

✓ When Ctrl+S pressed in editor

✓ Or on demand via Build Project action





Java Development Tools



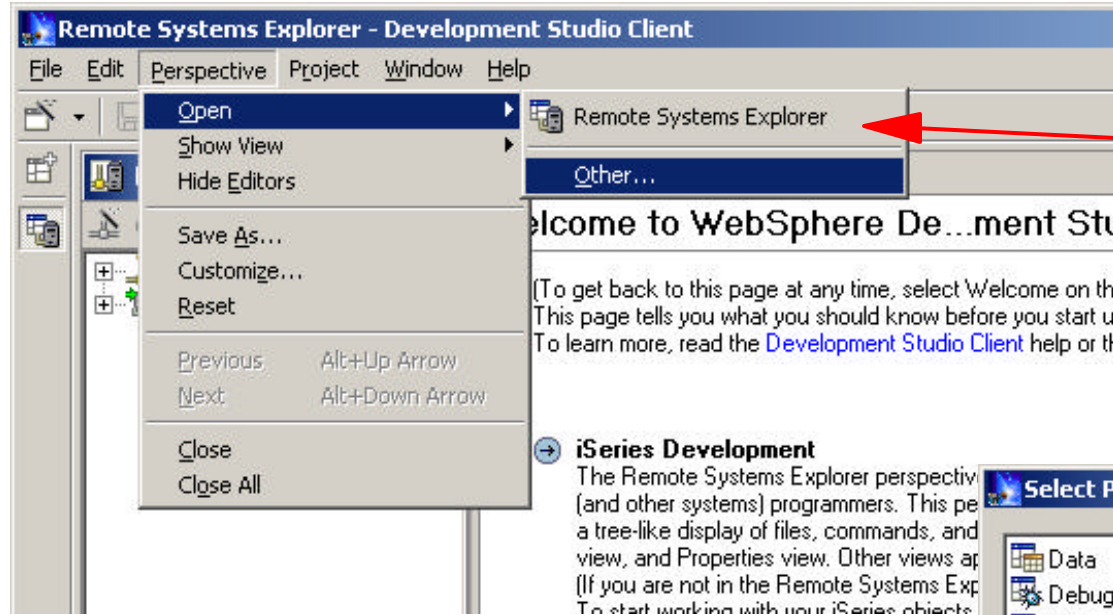
iSeries AD, IBM Toronto

▶ Getting Started



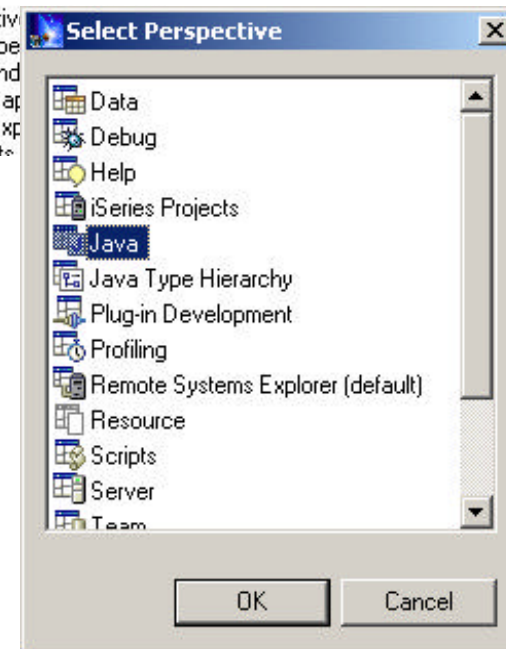
- ▶ Open the Java perspective
 - **Perspectives**
 - ▶ Define which views are shown and their layout in the Workbench
 - ▶ Targeted towards a specific type of task
 - ✓ Java, Help, Web, WebFacing, Remote Systems Explorer, XML, ...
 - **Java perspective contains Java related views**
 - ▶ **Packages**
 - ✓ Shows only the Java projects in your workspace
 - ✓ Collapses package directories into a single folder in the tree view
 - ▶ **Outline (standard workbench view)**
 - ✓ High level view overview of the structure of your Java class
 - ▶ **Hierarchy**
 - ✓ Inheritance hierarchy of the Java class
 - ▶ **Tasks (standard workbench view)**
 - ✓ Shows compile errors and warnings

Java Perspective



Shortcuts to related perspectives

All available perspectives





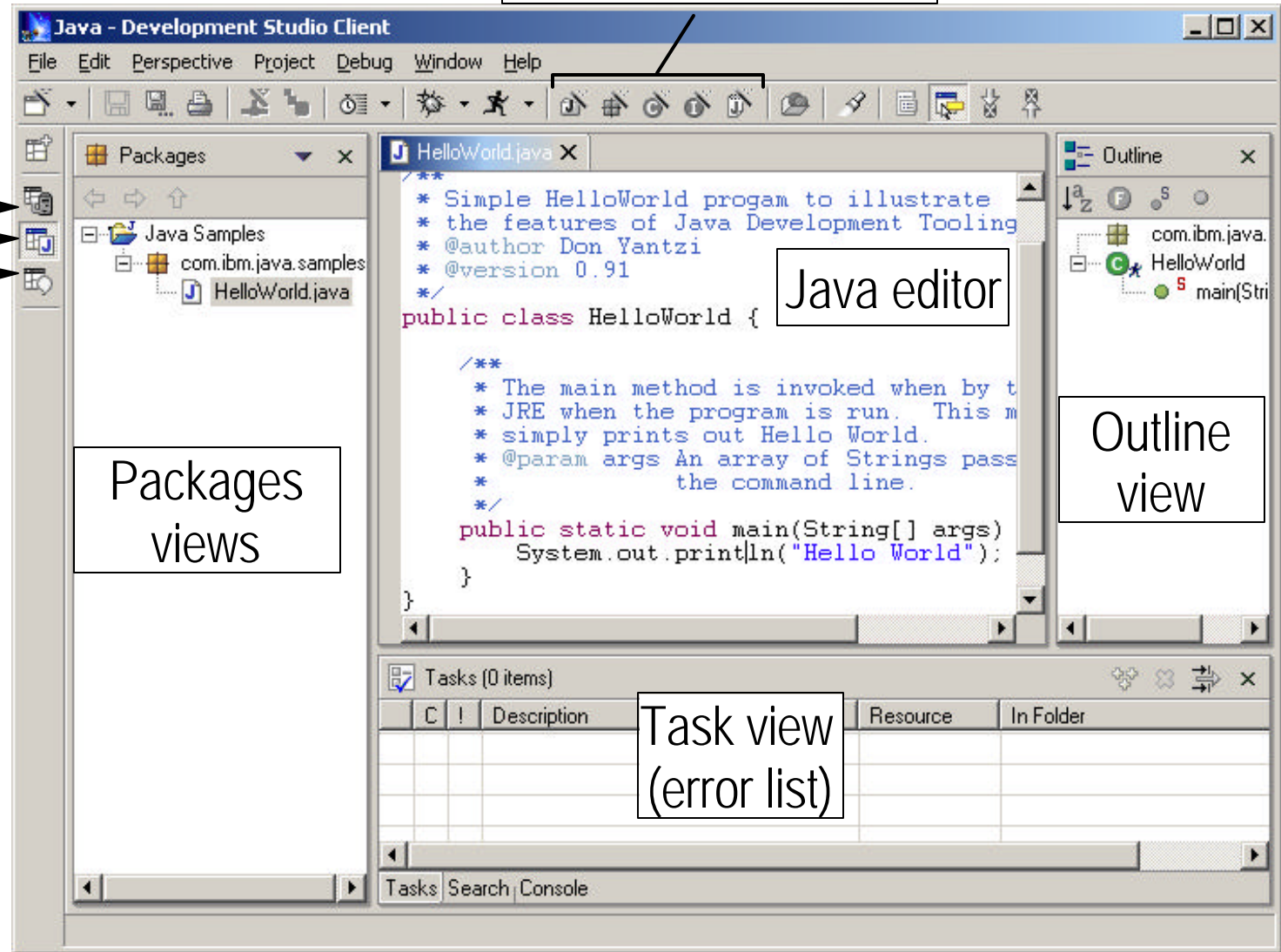
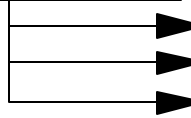
Java Perspective



iSeries AD, IBM Toronto

Java specific wizards

Open perspectives



Packages views

Java editor

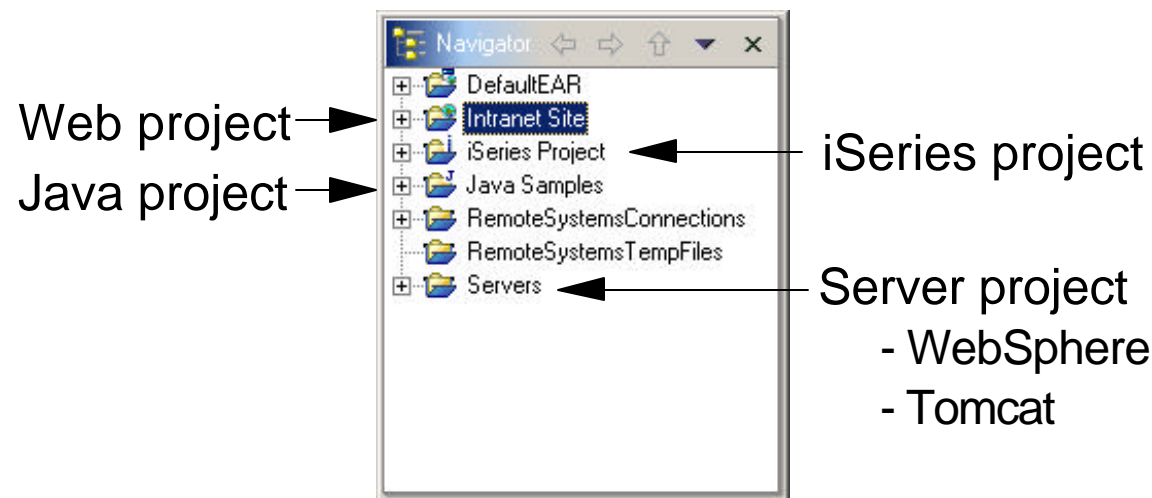
Outline view

Task view (error list)

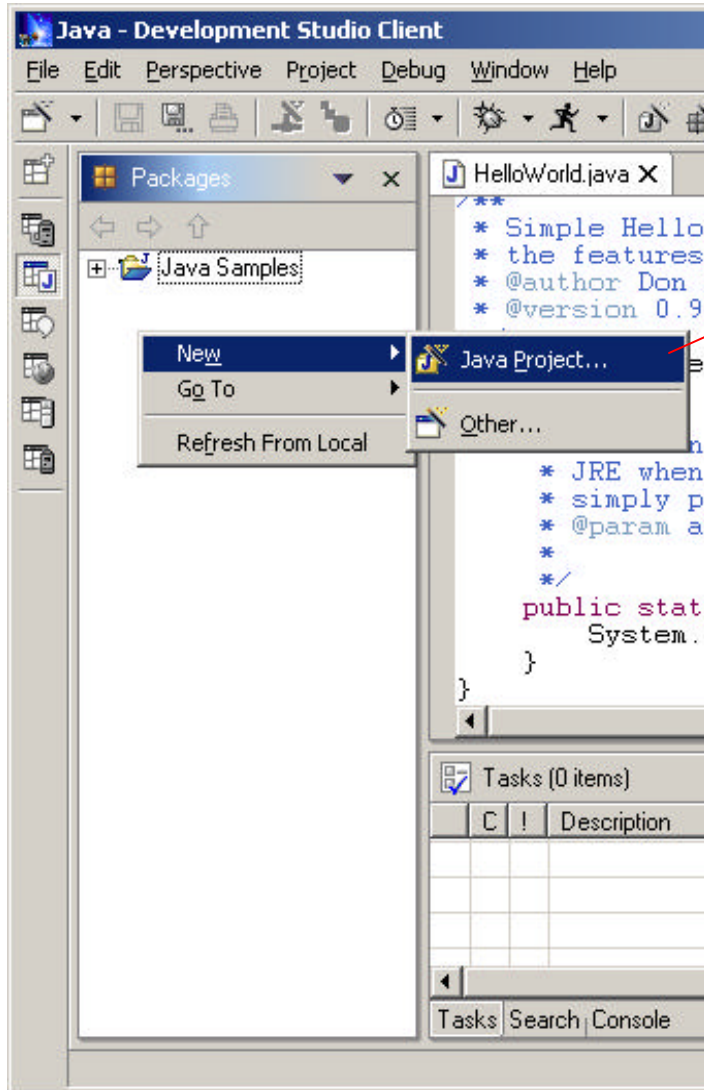


▶ Projects

- **Highest level of organization for resources**
- **Contains files and folders**
- **Projects have**
 - ▶ **A type**
 - ✓ Java, iSeries, Server, Web, Simple (generic)
 - ▶ **Properties**
 - ▶ **Associated builders**
 - ✓ Builders know how to convert source artifacts (files) to executables



New Java Project Wizard



Enter a name for your project.

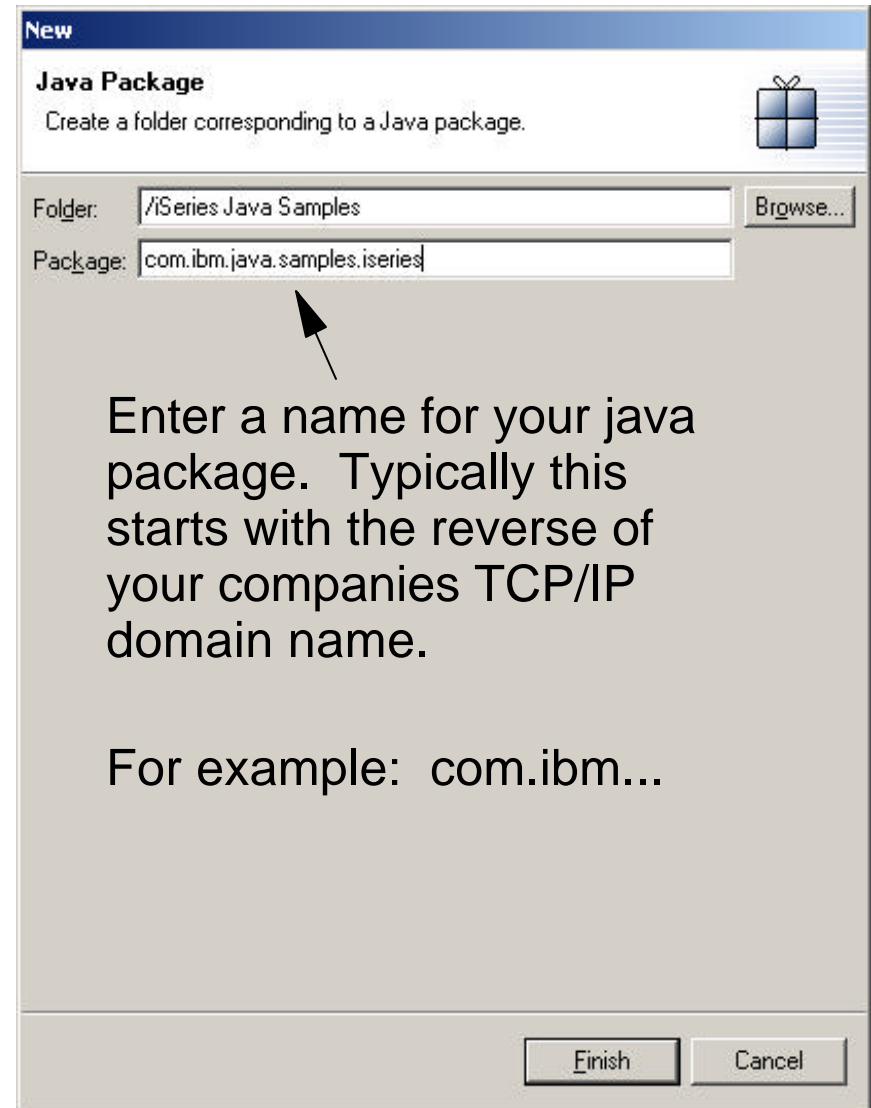
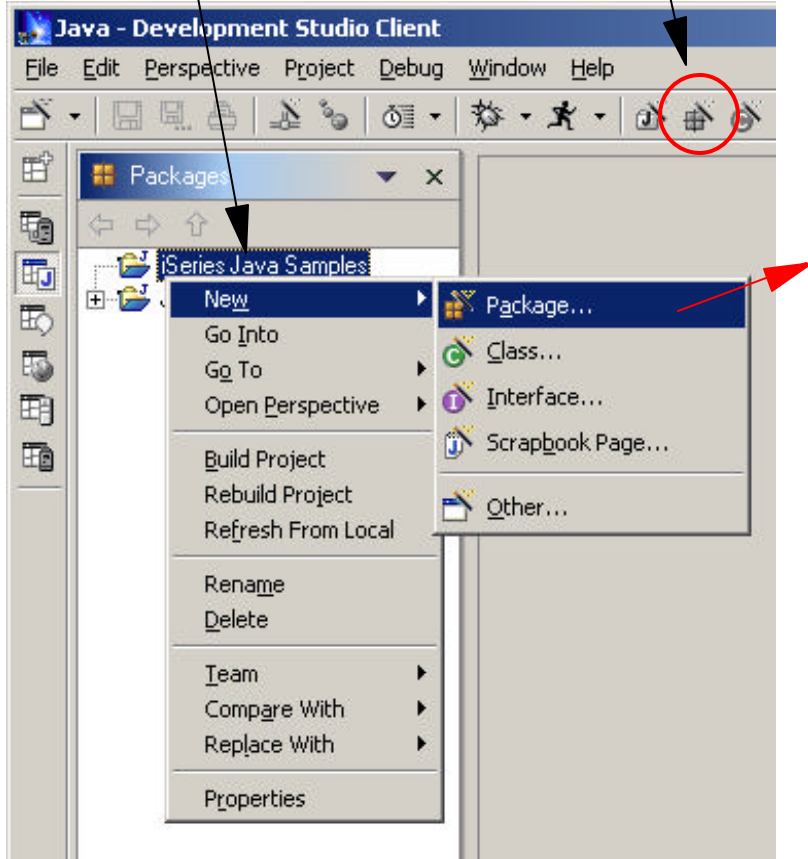


Creating a Java Package



Your Java Project!

Shortcut to Java package wizard



Note: Right click to get popup menu
 - or select shortcut icon
 - or select File -> Java Package

Creating a Java Class



Folder and package names are prefilled from Workbench selection

Create this class as an inner class

Enter a name for the class

Enter any interfaces you want your class to implement

The wizard can generate some common methods if you want them

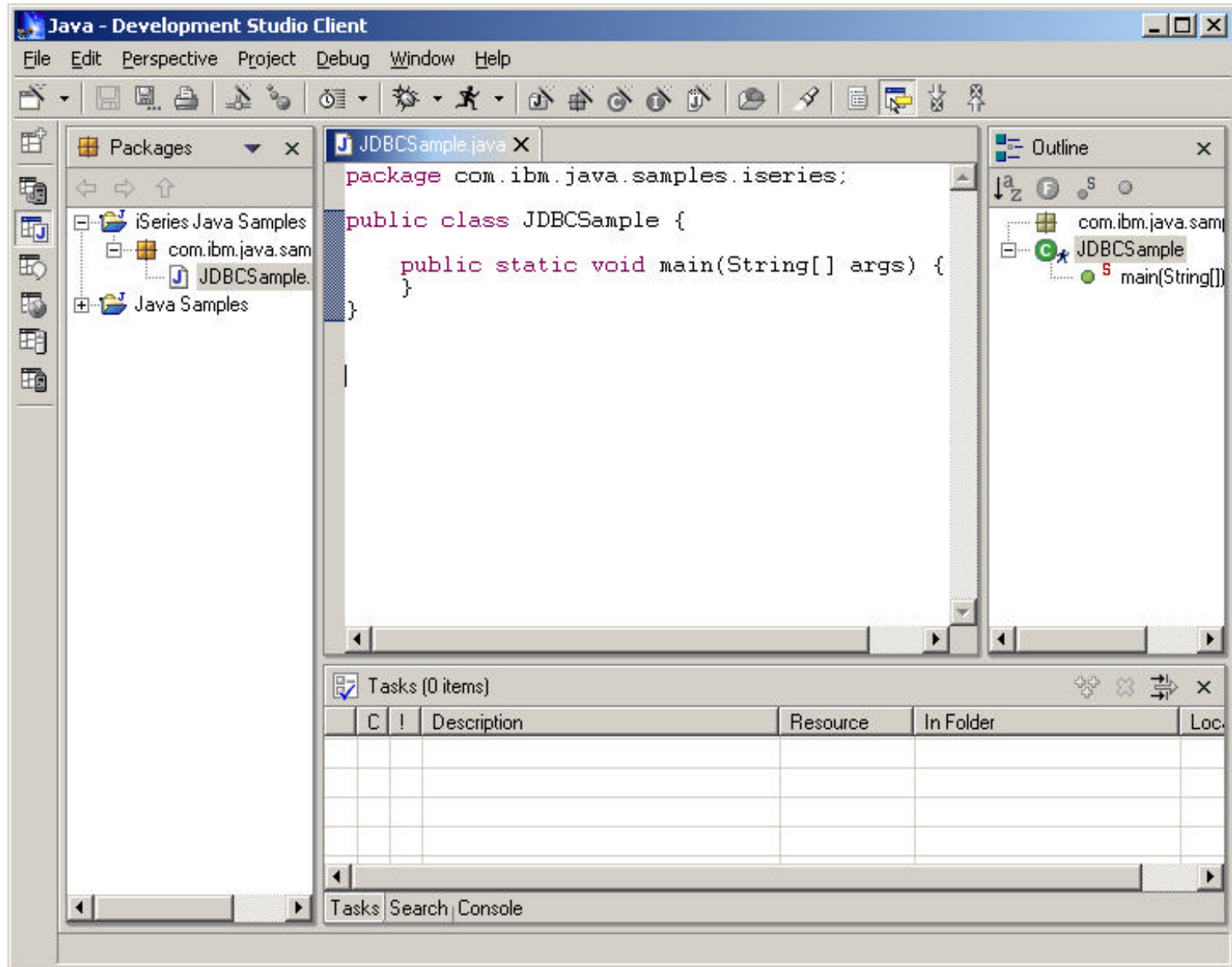
Select any access modifiers for the class

Enter the name of a superclass or click Browse to select one from your workspace

Good So Far ...



iSeries AD, IBM Toronto





Java Development Tools



iSeries AD, IBM Toronto

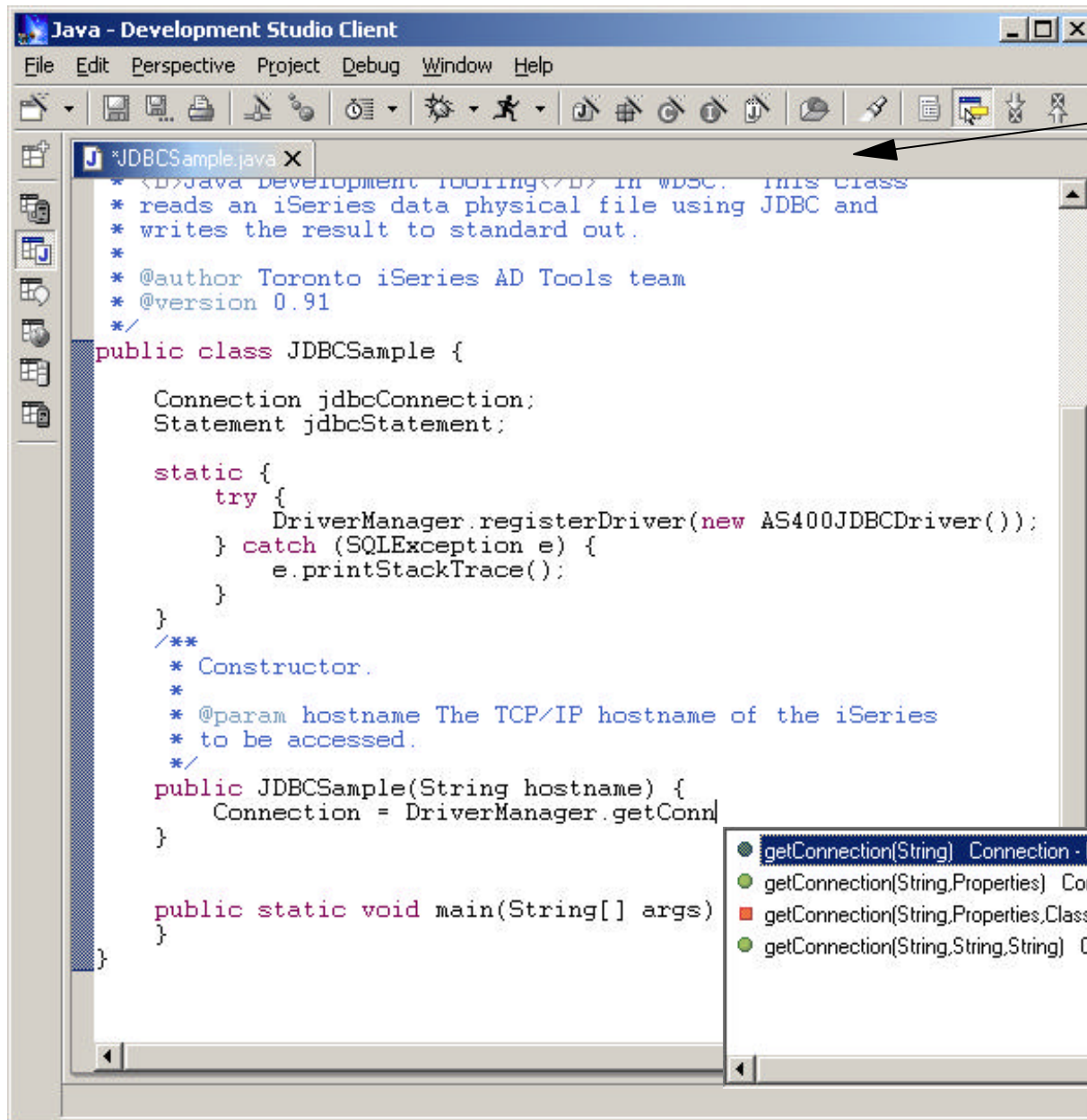
▶ The Java Editor



Code Assist - Java Code



iSeries AD, IBM Toronto



Tip: Double click on title bar to make editors and views full screen

Ctrl + Space brings up the code assist with a list of all possible matches based on what is already typed

Code Assist - JavaDoc



iSeries AD, IBM Toronto

```
Java - Development Studio Client
File Edit Perspective Project Debug Window Help

*JDBCExample.java X
import java.sql.Connection;
import java.sql.Statement;
import java.sql.SQLException;

import com.ibm.as400.access.*;

/**
 * Sample class to illustrate the features of the
 * <b>Java Development Tooling</ in WDSC. This class
 * reads an iSeries data physical
 * writes the result to standard
 *
 * @author Toronto iSeries AD To
 * @version 0.91
 */
public class JDBCExample {

    Connection jdbcConnection;
    Statement jdbcStatement;

    static {
        try {
            DriverManager.registerDriver(new AS400JDBCDriver());
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    /**
     * Constructor.
     *
     * @param hostname The TCP/IP hostname of the iSeries
     * to be accessed.
     */
    public JDBCExample(String hostname) {
        Connection = DriverManager.getConn
    }
}
```

Code assist is also available in JavaDoc comments for:

- HTML
 - ✓ ...
 - ✓ <i> ... </i>
 - ✓ ...
- JavaDoc keywords
 - ✓ @param
 - ✓ @return
 - ✓ ...



> javadoc JDBCSample.java

Class [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

`com.ibm.java.samples.iseries`
Class JDBCSample

`java.lang.Object`
|
+---`com.ibm.java.samples.iseries.JDBCSample`

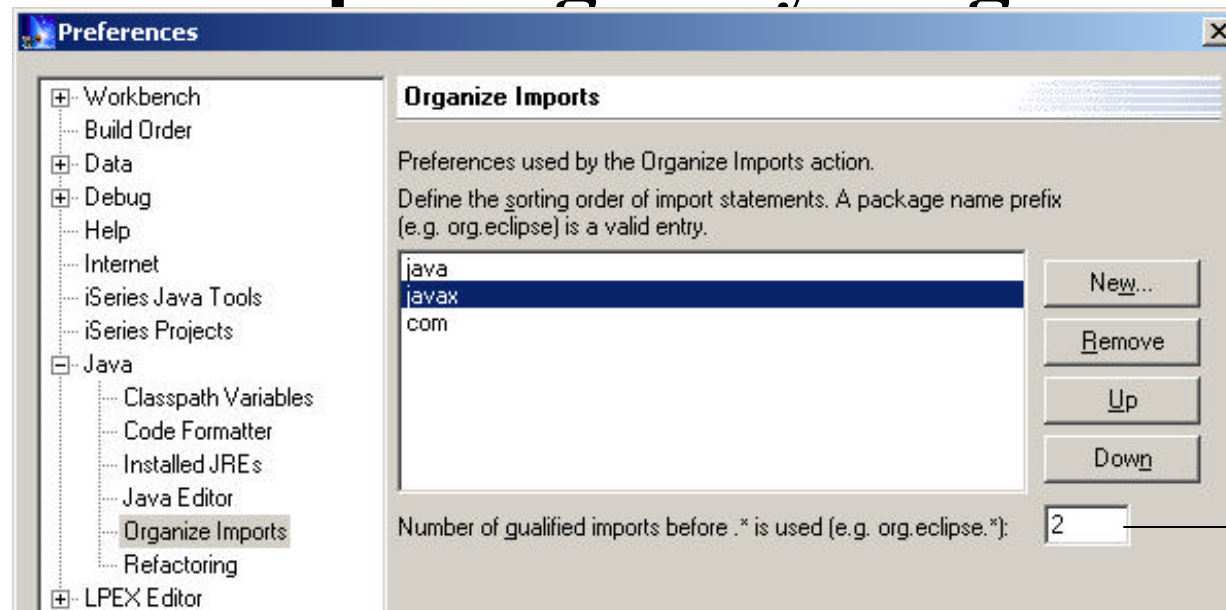
```
public class JDBCSample
extends java.lang.Object
```

Sample class to illustrate the features of the **Java Development Tooling** in WDCS. This class reads an iSeries data physical file using JDBC and writes the result to standard out.

Done My Computer



- ▶ First setup preferences
 - **Window -> Preferences**
 - **Specify order imports should be listed**
 - ▶ For example: `java`, then `javax`, then `com`
 - ▶ Add your own!!!
 - **Specify how many qualified imports to use before importing everything**



See next slide

Managing Your Imports



```

Java - Development Studio Client
File Edit Perspective Project Debug Window Help

JDBCsample.java X
package com.ibm.java.samples.iseries;

import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.SQLException;

import com.ibm.as400.access.*;

/**
 * Sample class to illustrate the features of the
 * <b>Java Development Tooling</b> in WDSC. This class
 * reads an iSeries data physical file using JDBC and
 * writes the result to standard out.
 *
 * @author Toronto iSeries AD Tools team
 * @version 0.91
 */
public class JDBCsample {

```

- Undo
- Redo
- Revert
- Cut
- Copy
- Paste
- Select All
- Print
- Shift Right
- Shift Left
- Comment
- Uncomment
- Format
- Content Assist
- Add Import
- Organize Imports**
- Extract Method...
- Find/Replace...
- Go to Line...

```

Java - Development Studio Client
File Edit Perspective Project Debug Window Help

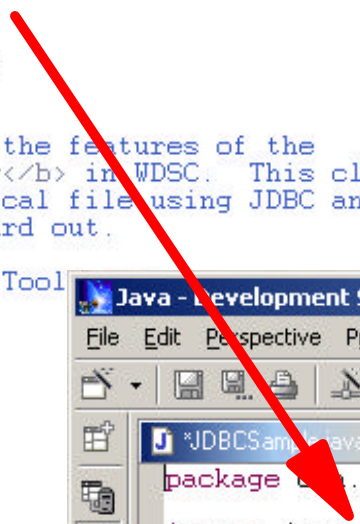
*JDBCsample.java X
package com.ibm.java.samples.iseries;

import java.sql.*;

import com.ibm.as400.access.AS400JDBCdriver;

/**
 * Sample class to illustrate the features of the
 * <b>Java Development Tooling</b> in WDSC. This class
 * reads an iSeries data physical file using JDBC and
 * writes the result to standard out.
 *
 * @author Toronto iSeries AD Tools team
 * @version 0.91
 */
public class JDBCsample {

```



Managing Your Imports



To import a class simply highlight the class name and select "Add Import" from the popup menu.

Tip: Double clicking on a class name or variable will highlight it.

Error in code because the class AS400 has not been imported yet.

The screenshot shows the Eclipse IDE with a Java file named 'JDBCExample.java'. The code contains a class 'JDBCExample' with a constructor that takes a 'hostname' parameter and creates a new 'AS400' object. A context menu is open over the 'AS400' class name, with 'Add Import' selected. The menu options include Undo, Redo, Revert, Cut, Copy, Paste, Select All, Print, Shift Right, Shift Left, Comment, Uncomment, Format, Content Assist, Add Import, Organize Imports, Extract Method..., Find/Replace..., Go to Line..., and Add. A red 'x' icon in the left margin indicates an error at the line where 'AS400' is used.

```

/*
 * <D>Java Development Tooling</D> in WDBC. This class
 * reads an iSeries data physical file using JDBC and
 * writes the result to standard out.
 *
 * @author Toronto iSeries AD Tools team
 * @version 0.91
 */
public class JDBCExample {
    Connection jdbcConnection;
    Statement jdbcStatement;

    static {
        try {
            DriverManager.registerDriver
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    /**
     * Constructor.
     *
     * @param hostname The TCP/IP hostname
     * to be accessed.
     */
    public JDBCExample(String hostname) {
        AS400 as400 = new AS400(hostname
    }

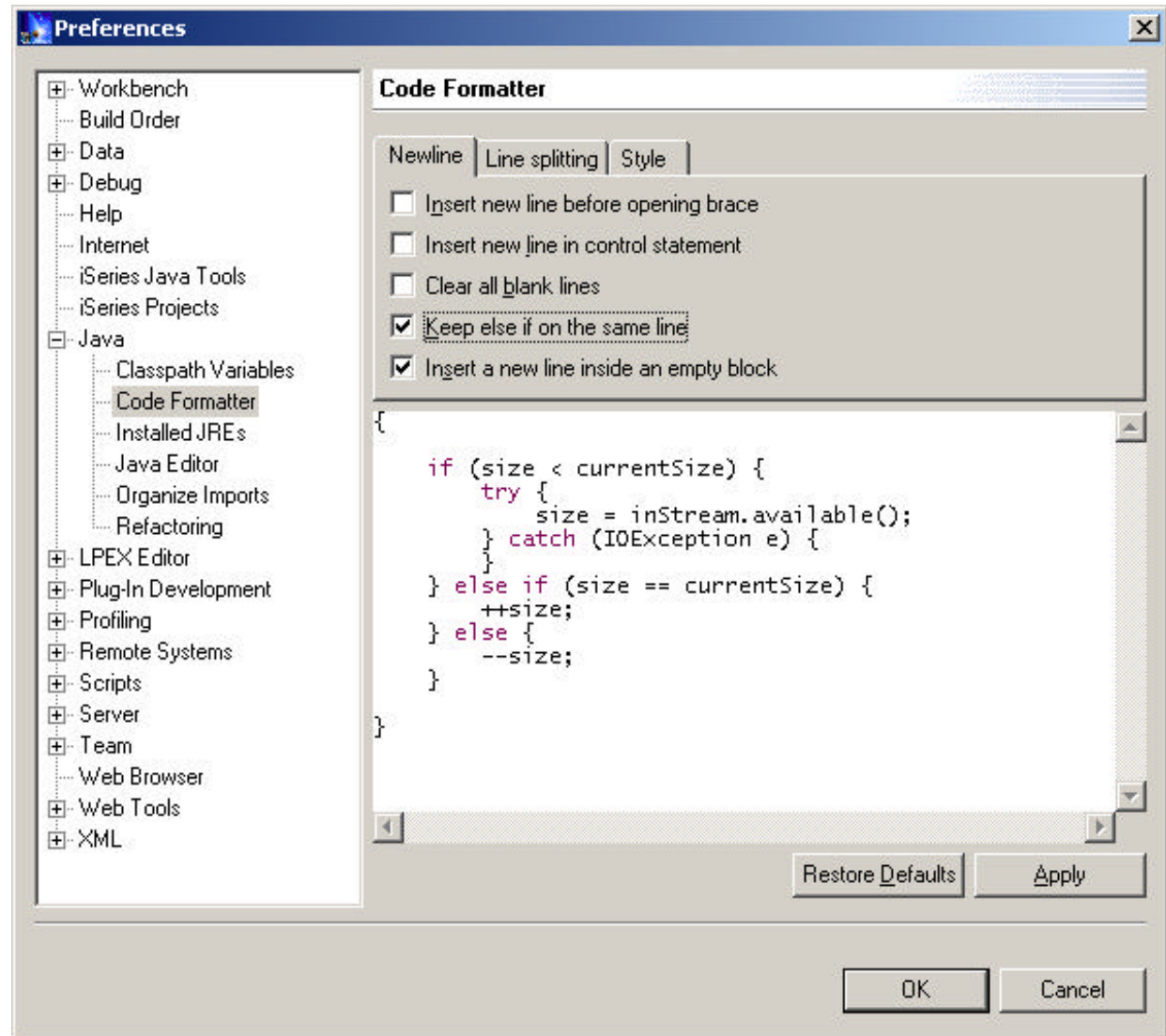
    public static void main(String[] args
    }
    
```



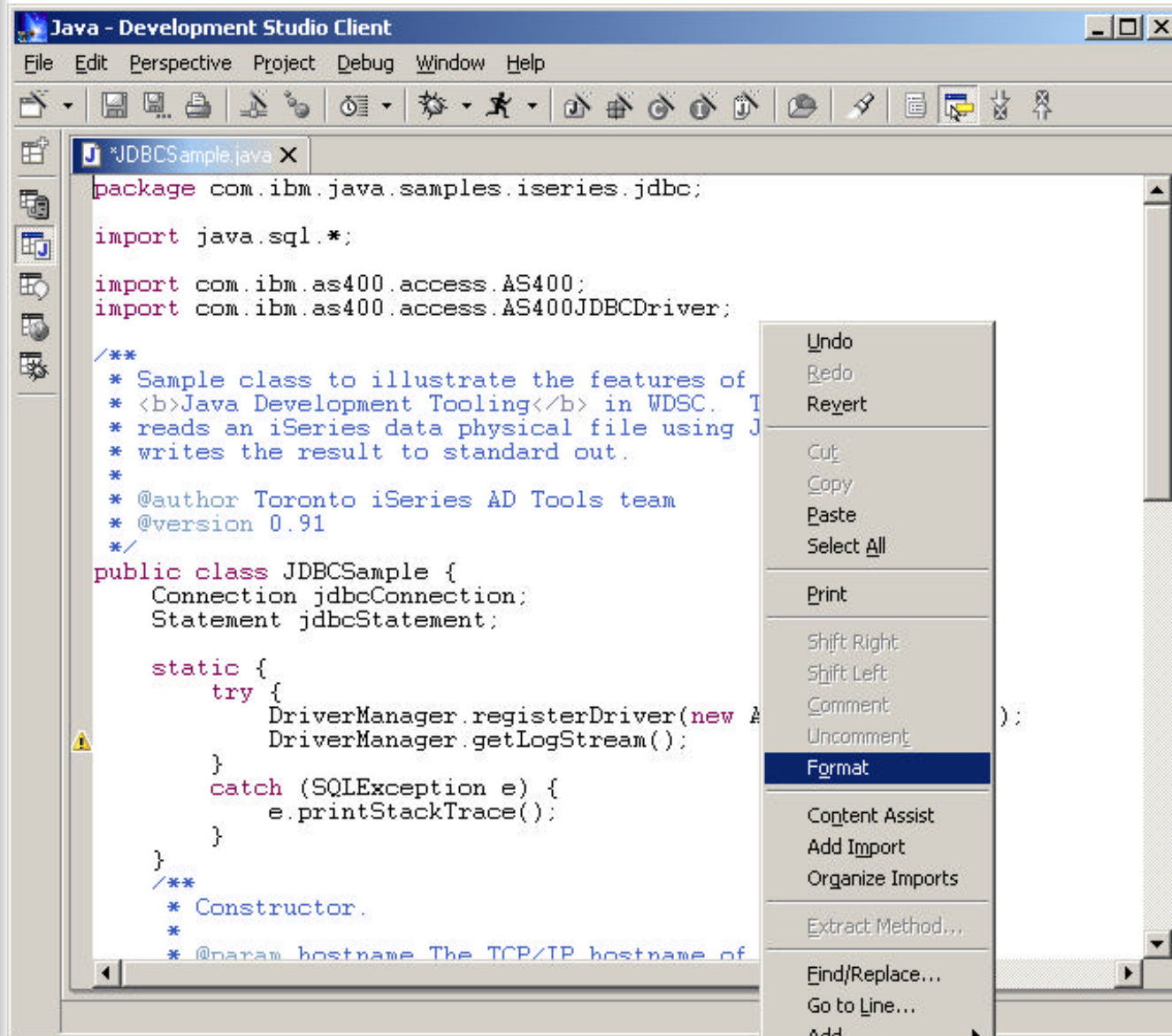
- The Java Editor can automatically format your code however you would like
 - ▶ Maximum line length
 - ▶ { } preferences
 - ✓ New lines
 - ✓ Same line as code
 - ▶ ...

1. First set your preferences in the preferences dialog

Then ...



Formatting Your Code



2. Right click in the editor and select "Format"

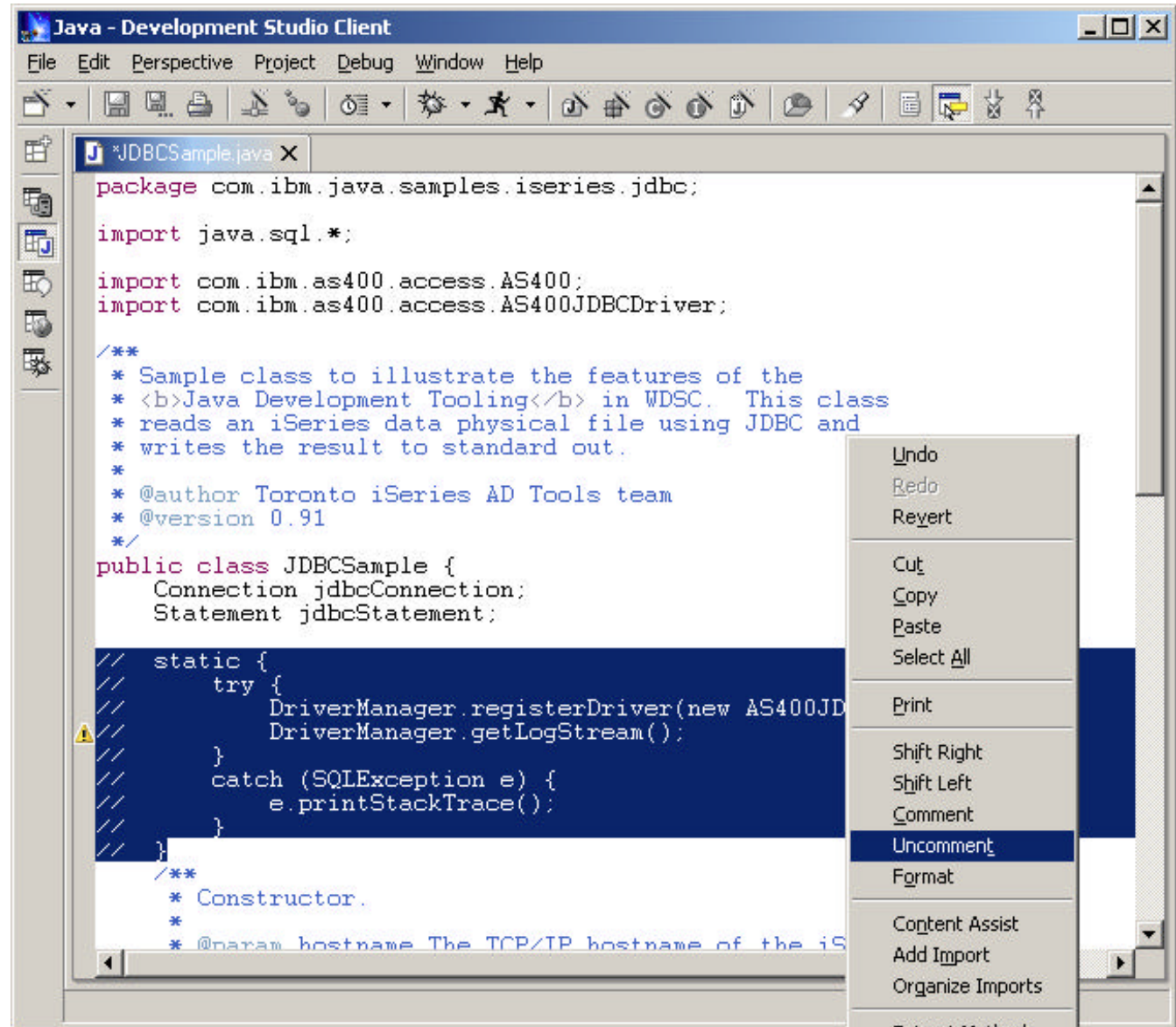
VOILA!

Other Misc. Functions



- Editor also lets you manipulate blocks of code at a time
 - ▶ Shift right
 - ▶ Shift left
 - ▶ Comment
 - ▶ Uncomment

- Select the block of code, right click and select the required function from the popup menu





- ▶ Nominees For Best Supporting View



- ▶ Navigator View
 - The Eclipse tree view for managing all resources in the workspace
 - Shows Java classes in their package directories
 - ▶ This takes a lot of space in the tree view
 - ▶ For example: `com/ibm/java/samples`
- ▶ Packages View
 - Provides central place to manage your Java resources
 - Shows only Java related projects
 - ▶ This includes Web projects!
 - Packages shown as single entry in tree view

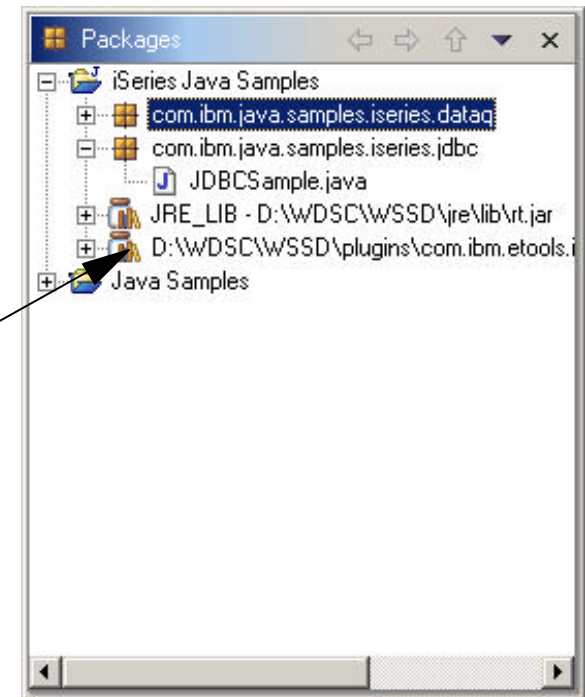
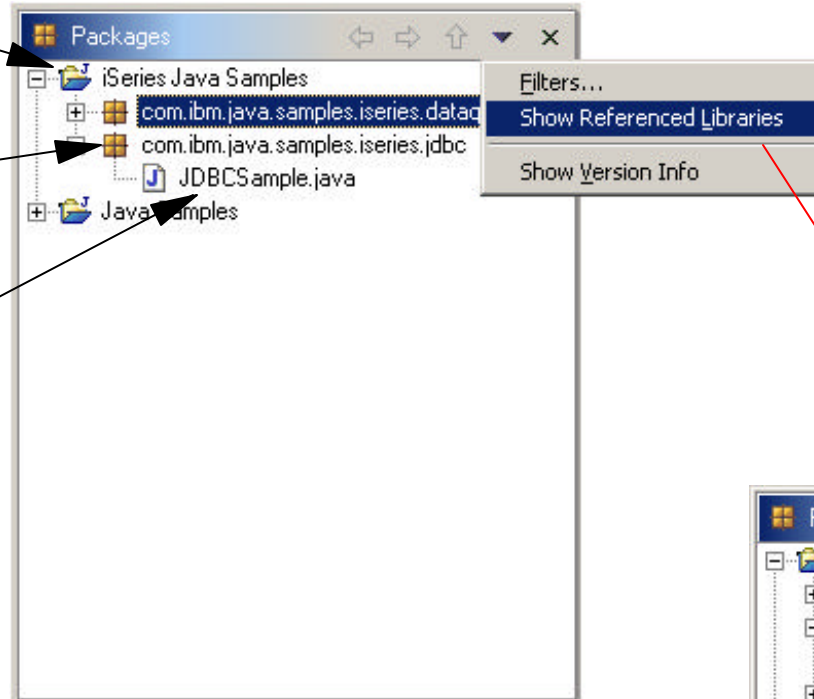
Packages View



Projects

Packages

Source files



Libraries referenced by this project (more on this later when we look at the project build properties)

Tip: These can also be expanded!

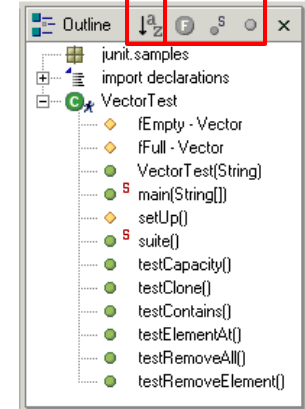
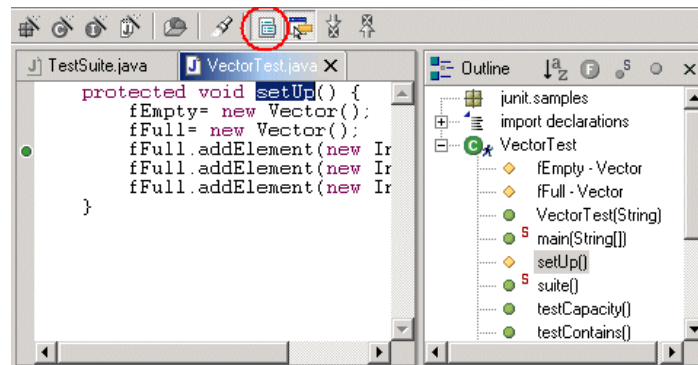
Outline View



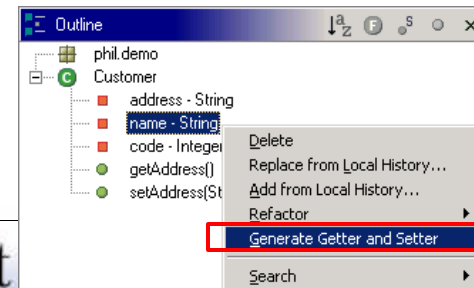
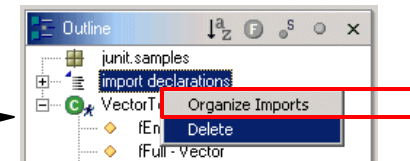
▶ Java Outline View

- Works in concert w/ Java Editor
- Shows imports, fields, methods
 - ▶ Can subset what is shown
 - ▶ Can sort by name

- Can show only selected member in editor



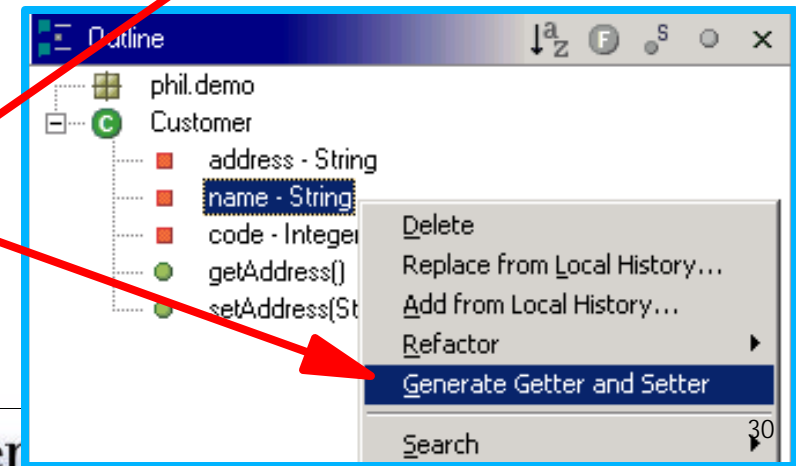
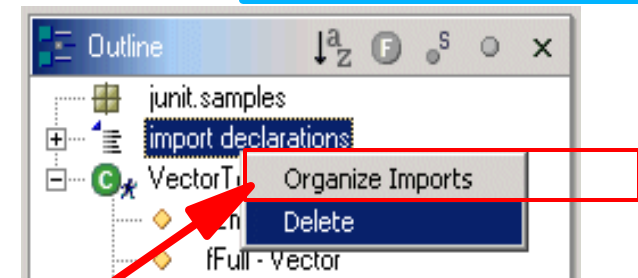
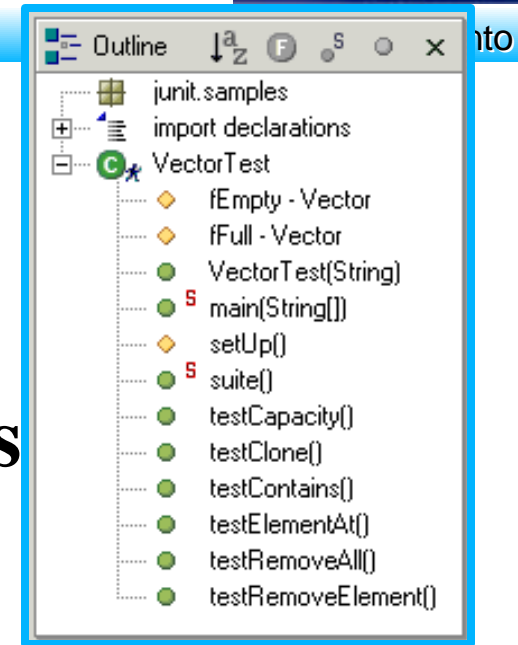
- Can auto-add imports
- Can auto-gen getter and setter methods



WDSsc: Java Tools



- ▶ Java Outline View
 - Works in concert w/ Java Editor
 - Shows imports, fields, methods
 - ▶ Can subset what is shown
 - ▶ Can sort by name
 - Can restrict to selected member only in editor
 - Can auto-add imports
 - Can auto-gen getter and setter methods





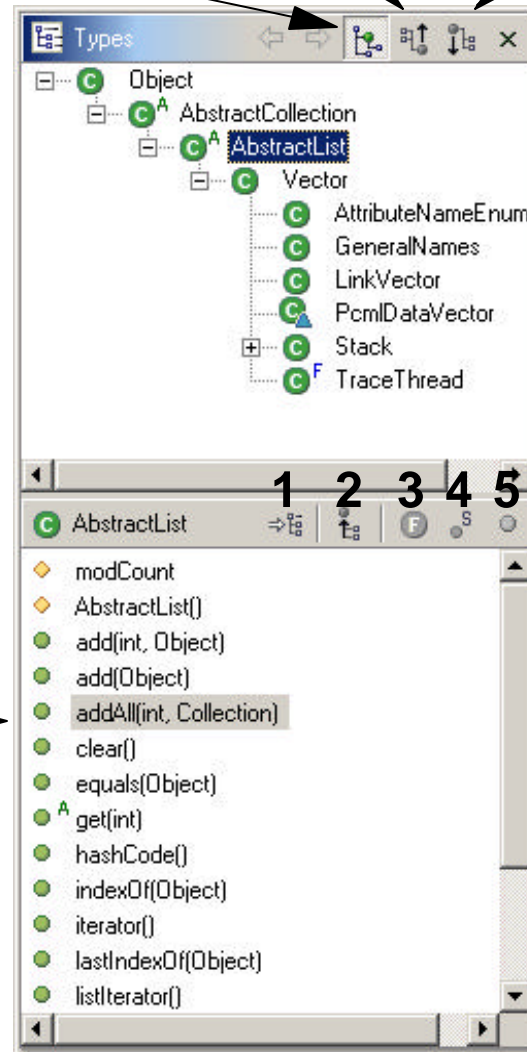
- ▶ Hierarchy View
 - Provides 3 different ways to view a class in its type hierarchy
 - ▶ Show only supertype hierarchy
 - ✓ This includes superclasses and interfaces!
 - ▶ Show only subtype hierarchy
 - ▶ Show class in its full hierarchy tree
 - ✓ Includes both superclasses and subclasses
 - Ability to perform actions
 - ▶ Create copies of methods from superclasses in current class
 - ✓ Right click on method and select "Create in *classname*"
 - ▶ Refactoring
 - ✓ Rename methods and fields
 - ✓ Rename parameters for methods
 - ✓ Generate getter and setter methods for fields

Hierarchy View



Supertype hierarchy
 Type hierarchy
 Subtype hierarchy

To open a class in the hierarchy view highlight the class name in any source file or view and press F4.



Currently showing all methods in AbstractList class

1. Shows selected members in hierarchy at all locations where they are declared.

✓ Lets you see where members are declared and overridden.

2. Show / hide inherited members

3. Show / hide fields

4. Show / hide static members

5. Show / hide nonpublic members



▶ Tasks view

● Shows:

1. All compile errors, warnings and information associated with resources in the workspace
2. Tasks defined by the user
 - ✓ Tasks can be set on a specific line in a resource or globally
 - ✓ "Don't forget to add error checking here!"

● Not Java specific!

- ▶ Shows errors and tasks for Java, HTML, XML, RPG, ...

● Provides ability to filter by

- ▶ Type of problem
 - ▶ Java problem, XML Schema problem, broken links in HTML or JSP
- ▶ Severity of the problem or priority of the task
- ▶ Problem / task description
- ▶ Based on what is selected in the navigator / packages view
- ▶ Status of the task

Tasks View



Global task
→ no resources is specified

Warning

Error

Completed task

Resource where the problem / task occurs

Project and folder(s) where the resource is located

Filter

	Description	Resource	In Folder	Location
<input checked="" type="checkbox"/>	Finish Java Tooling Presentations			
	The method getLogStream() from the type java...	JDBCExample.j...	iSeries Java Samples/com/ibm/ja...	line 27 in J...
	AS400 cannot be resolved or is not a type	JDBCExample.j...	iSeries Java Samples/com/ibm/ja...	line 39 in J...
<input checked="" type="checkbox"/>	Implement main method	JDBCExample.j...	iSeries Java Samples/com/ibm/ja...	line 45

Tip: Double click on the task / problem to open the resource in the editor at the correct line!



▶ Compiling

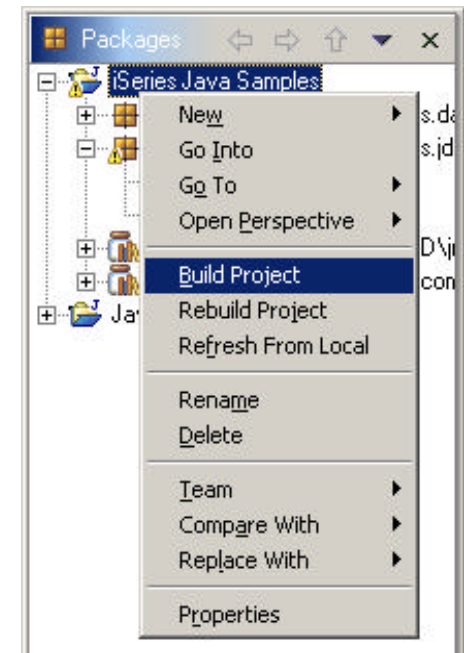


▶ Done!

- **Everytime you save your file it is automatically compiled!**
- **Manually compiling**

▶ Right click on the project, package or class and select either:

- ✓ Build Project
 - ✓ Incremental build
 - ✓ Builds only resources that have changed since the last build
- ✓ Rebuild Project
 - ✓ Full build
 - ✓ Discards previous build state and rebuilds everything





▶ Running

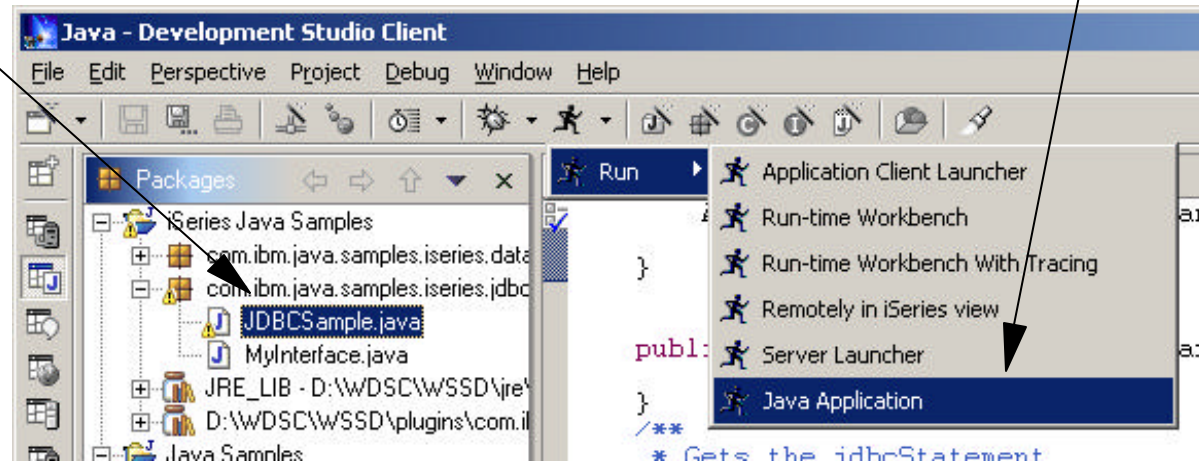


- ▶ How you run your class depends on what it is
 - **Standalone Java application**
 - ▶ **Class implements method:**
 - ✓ public static void main(String[] args)
 - ▶ **Run locally using**
 - ✓ JDK shipped with the workbench
 - ✓ Any locally installed JDK
 - ▶ **Run remotely using iSeries extensions**
 - **Servlet**
 - ▶ **Class extends javax.servlet.http.HttpServlet**
 - ▶ **Servlet can be run in**
 - ✓ Local WebSphere Test Environment
 - ✓ Local Tomcat Test Environment
 - ✓ Local Tomcat Application Server
 - ✓ Remote WebSphere Application Server

Standalone Applications

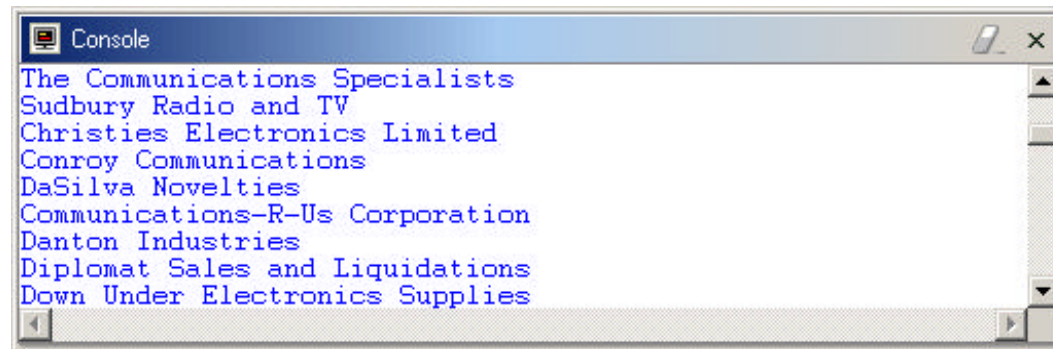


1. Select java file in packages view
2. Select Run -> Java Application



Debug perspective opens and output shown in Console view

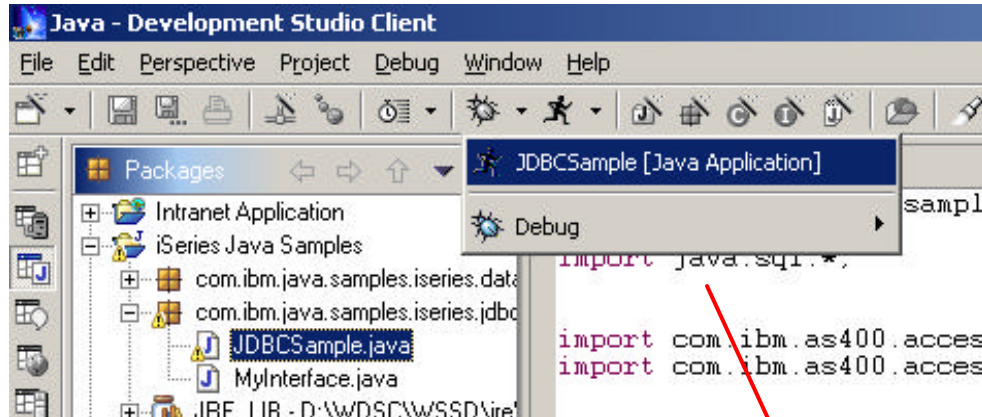
Tip: Turn off debug preference to "Show debug perspective when program launched in run mode".





▶ Java Debug
Not that you will need it!

Debug

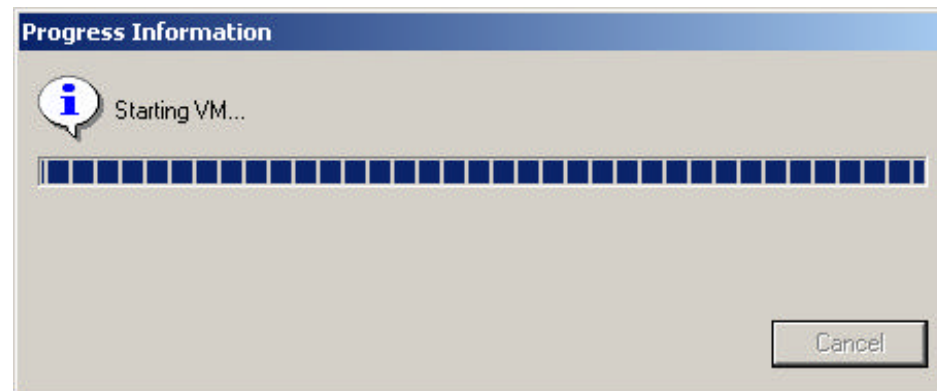


Shortcut: Run and debug menus remember the last few applications that were run.

You can select this instead of selecting the class and then choosing Debug -> Java Application.

Tip: Set breakpoints in your source code before launching the debugger.

You can set breakpoints in the Java editor by double clicking in the left hand margin of the editor.



Opens debug perspective ...



Debug Perspective



iSeries AD, IBM Toronto

Debug - Development Studio Client

File Edit Perspective Project Debug Window Help

Debug Console

JDBCSample [Java Application]

- JDBCSample at localhost:12662
 - System Thread [Finalizer] (Running)
 - System Thread [Reference Handler] (Running)
 - System Thread [Signal dispatcher] (Running)
 - Thread [main] (Suspended (breakpoint at line 52 in JDBCSample.getResultSet(String, String)))

Variables

- this=JDBCSample (id=17)
 - jdbcConnection=AS400JDBCConnection (id=21)
 - jdbcStatement=AS400JDBCStatement (id=23)
 - library="ADTSLAB"
 - file="CUSTOMER"

JDBCSample.java

```
private ResultSet getResultSet(String library, String file)
{
    ResultSet rs = jdbcStatement.executeQuery("SELECT * FROM...");
    return rs;
}

private void displayResultSet(ResultSet rs) throws SQLException
```

Outline

- com.ibm.java.samples
 - import declarations
 - JDBCSample
 - jdbcConnection
 - jdbcStatement
 - initializer

Console

Tasks Console

Details

Debug Perspective



Shortcut buttons for:

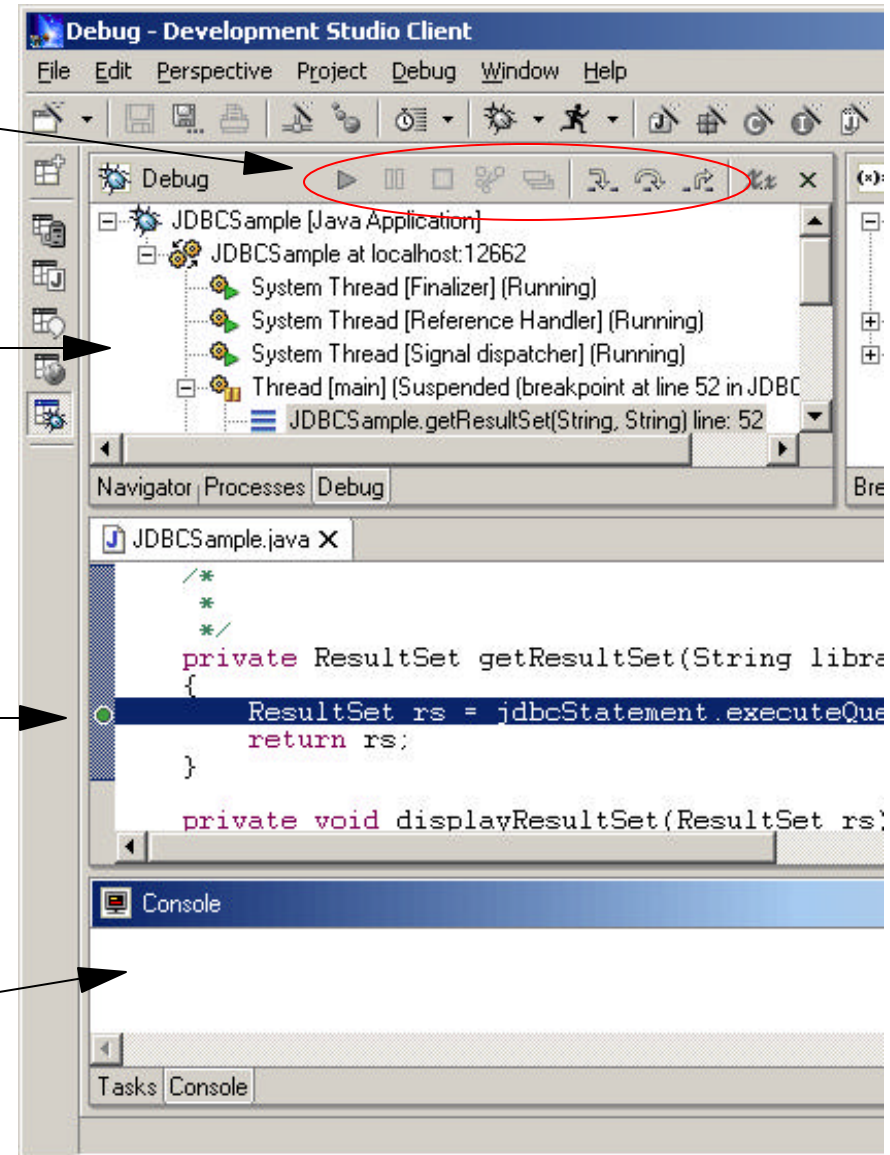
- ▶ Resume
- ▶ Suspend
- ▶ Stop
- ▶ Step into / over / return

Debug view shows:

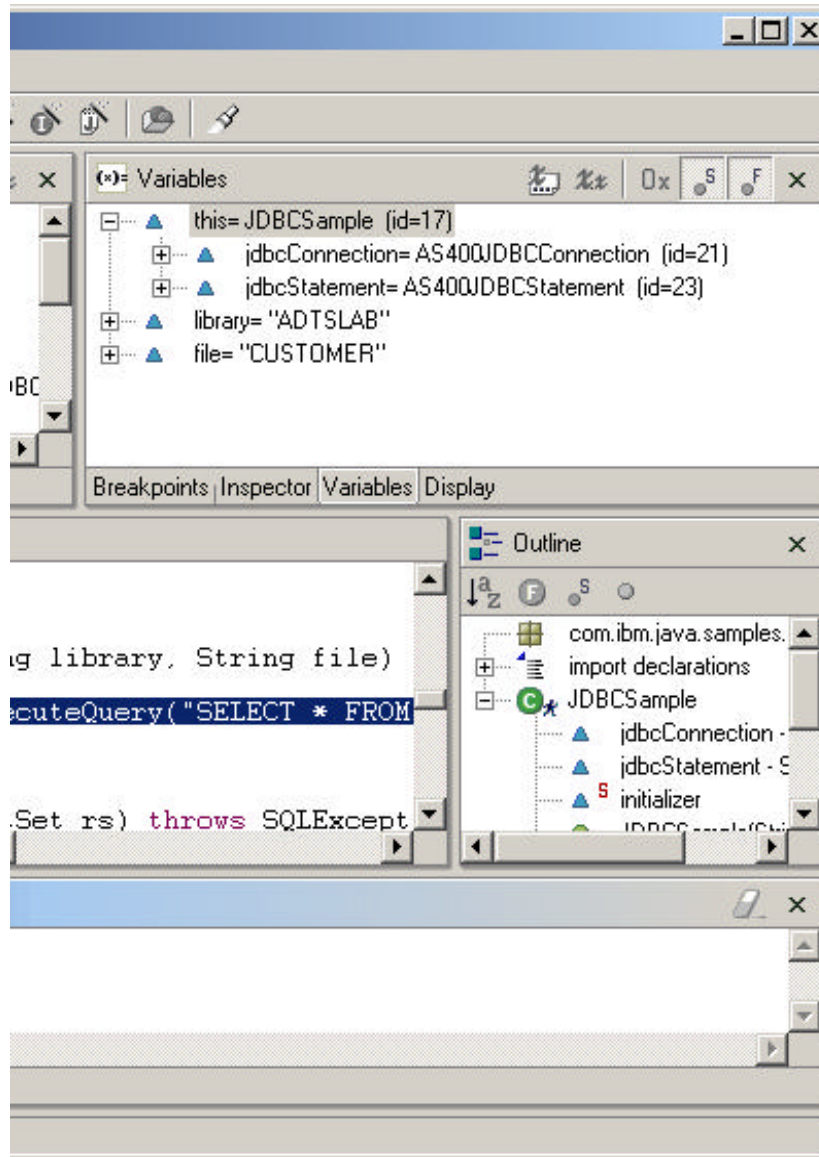
- ▶ All running applications
 - ▶ Threads
 - ▶ Execution stack for suspended threads

Breakpoint

Console View



Debug Perspective



- Breakpoints View

- ▶ Manage breakpoints in the workspace
- ▶ Add Java exception breakpoints
 - ✓ Stop when a NullPointerException is thrown
 - ✓ This can be very handle in locating problems!

- Inspector view

- ▶ Detailed view for inspecting variables

- Variables View

- ▶ Shows all variables that are visible in the current stack frame

- Display View

- ▶ Displays result of executing an expression in context of current stack frame
- ▶ Highlight expression in editor and select "Display" from the popup menu

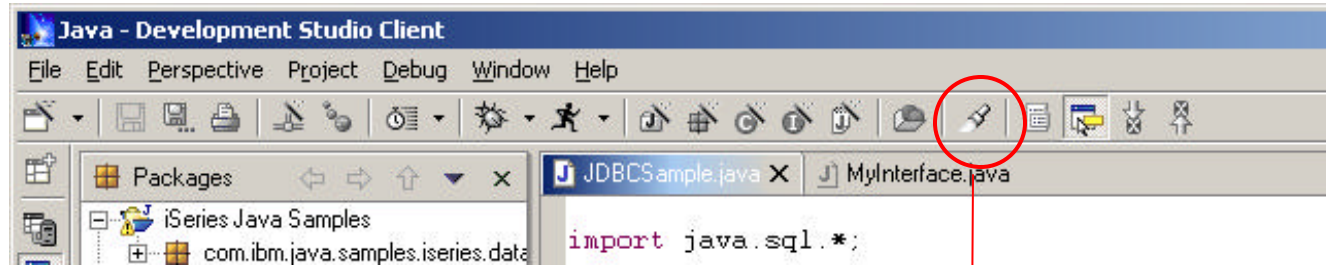


▶ Searching
Now where did that go?



- ▶ Two main ways to search
 - **Global workspace searching**
 - ▶ Search the entire workspace
 - **Context sensitive searching**
 - ▶ Scope to workspace or class hierarchy
 - ▶ Packages / Classes / Interfaces / Variables / Members
 - ✓ Find all places where referenced
 - ✓ Find all places where they are declared
 - ✓ In workspace, or
 - ✓ In class hierarchy
 - ▶ Interfaces
 - ✓ Find all implementors
 - **Results are shown in the "Search" view**

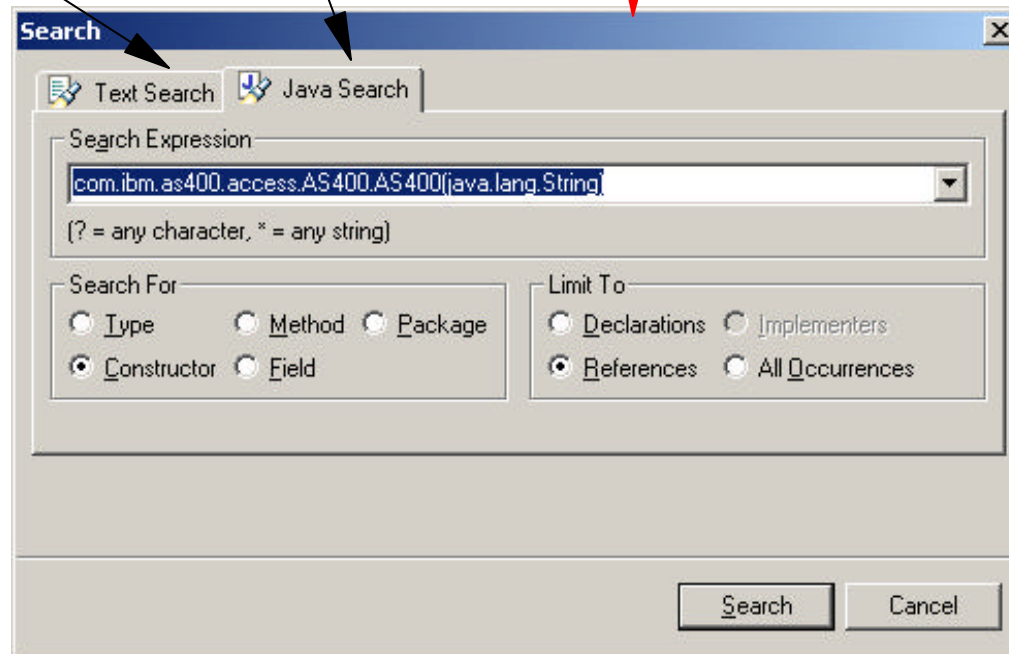
Java Aware Search



Java "aware" searching

Simple text searching

Find all types, methods, packages, constructors or fields that match the search expression.

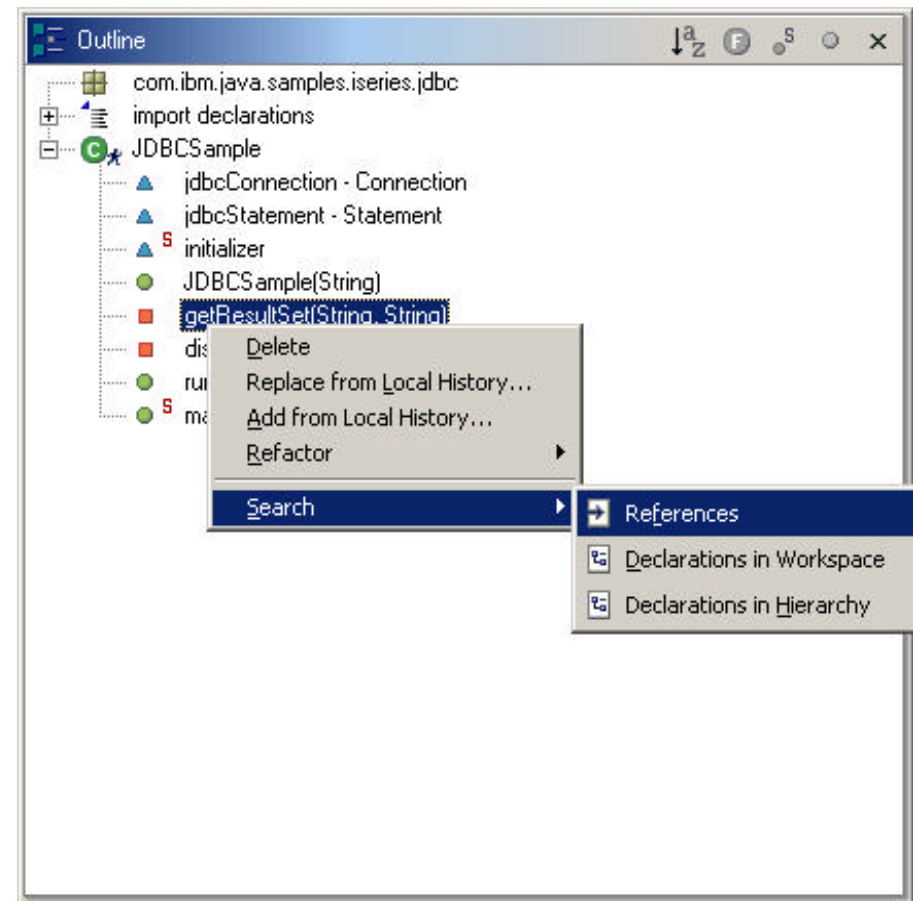


Search only for declarations or references to the specified type, method, package,



Tip: Anywhere you see a package / method / variable name you can right click on the name and select "Search -> ..." from the popup menu.

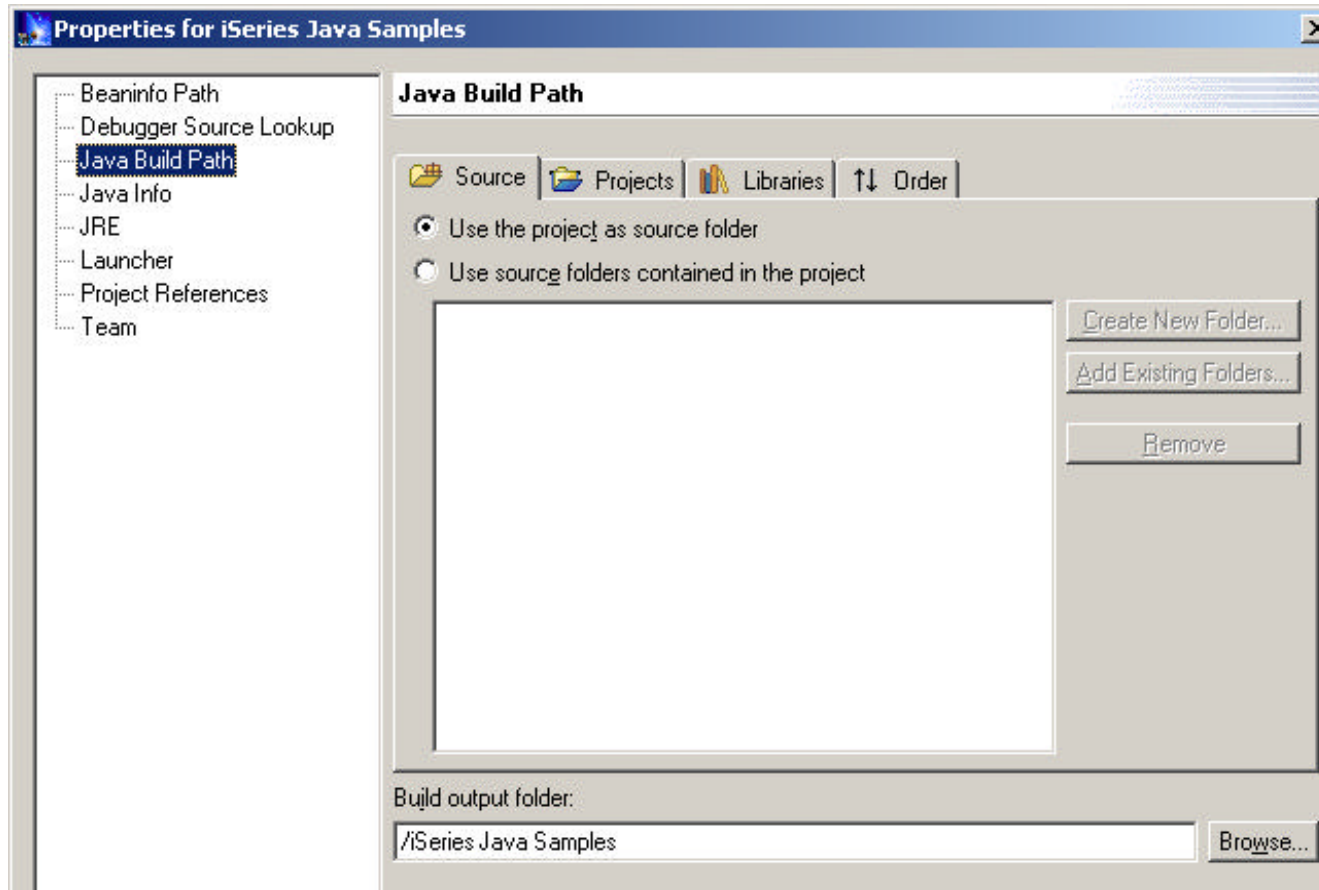
This is equivalent to selecting the search dialog entering the name and setting the "Search For" and "Limit To" fields.





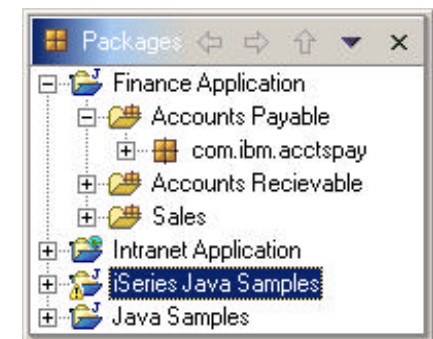
▶ Java Build Path

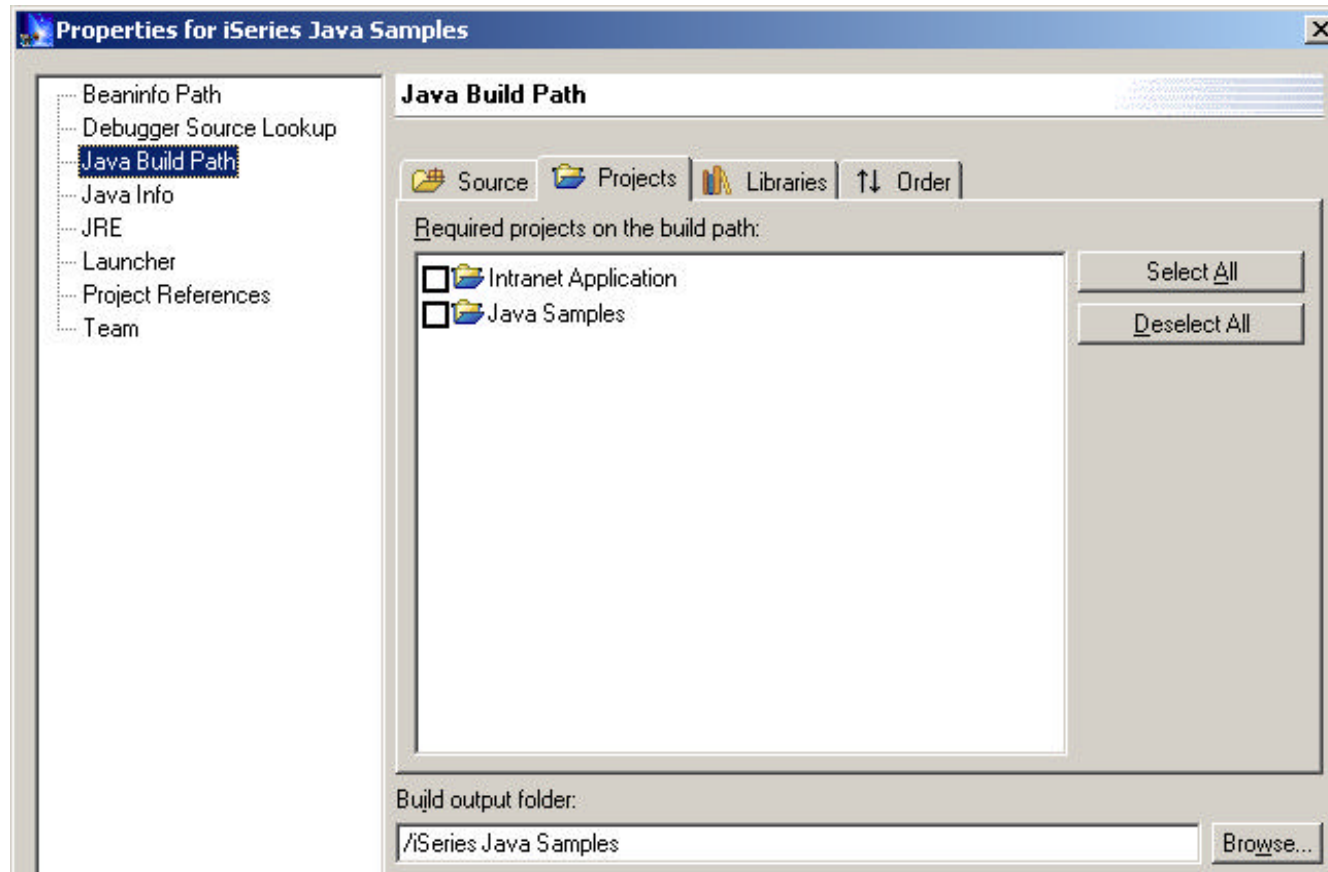
- **Each Java project has a build path that defines**
 - ▶ **How resources are built**
 - ✓ Where output (.class files) are placed after a build
 - ▶ **Where external resources are located**
 - ✓ Source from other projects in the workspace
 - ✓ JAR files from other projects
 - ✓ External JAR files
- **Similar to setting the CLASSPATH when running applications outside the workbench**



■ Source

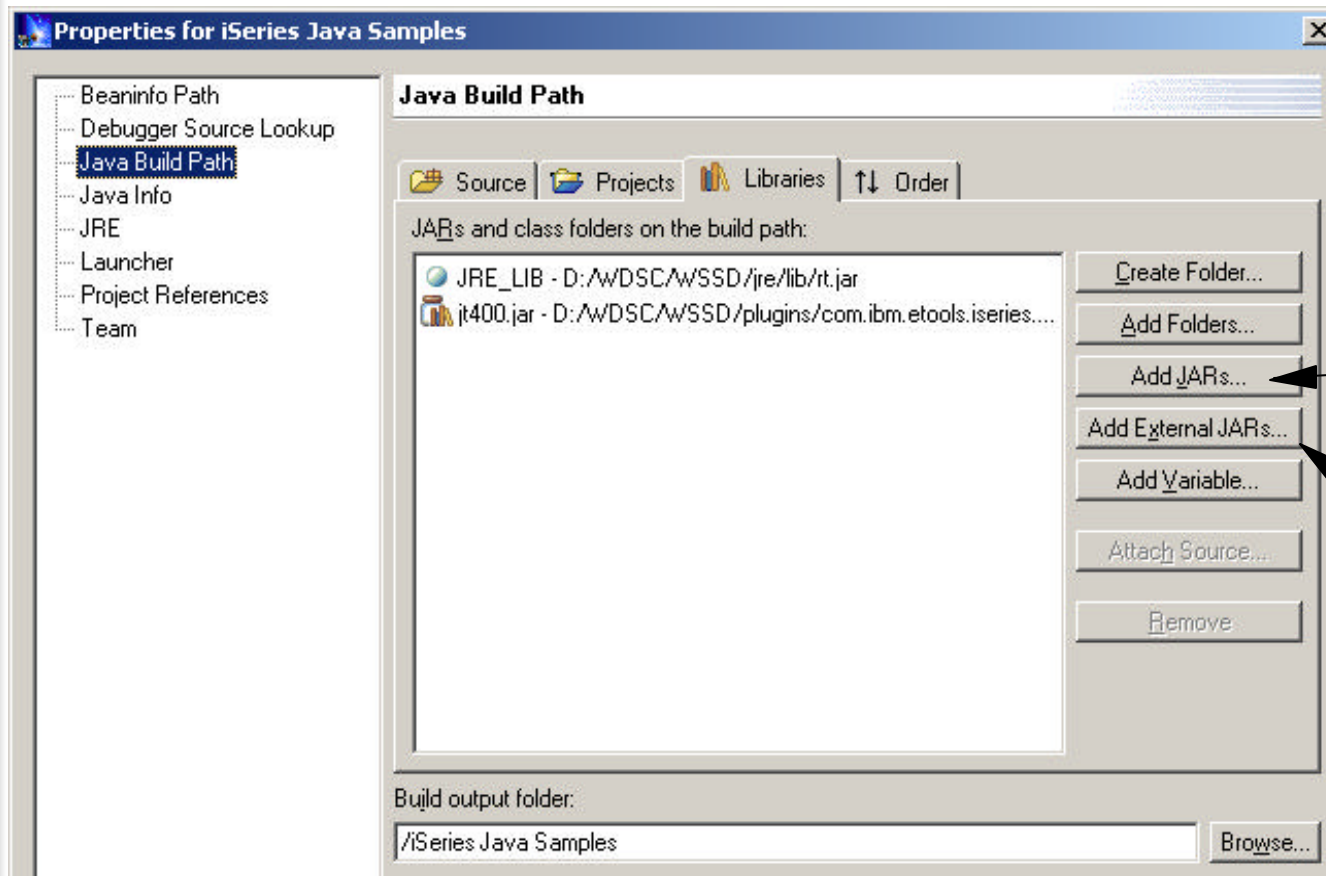
- ▶ Specify whether project uses source folders or not
- ▶ Source folders are recommended for large projects
 - ✓ Provides an additional level of grouping between the project and Java packages





■ Projects

- ▶ Select all projects in the workspace that are required by this project
 - ✓ i.e. This project uses classes in the other project



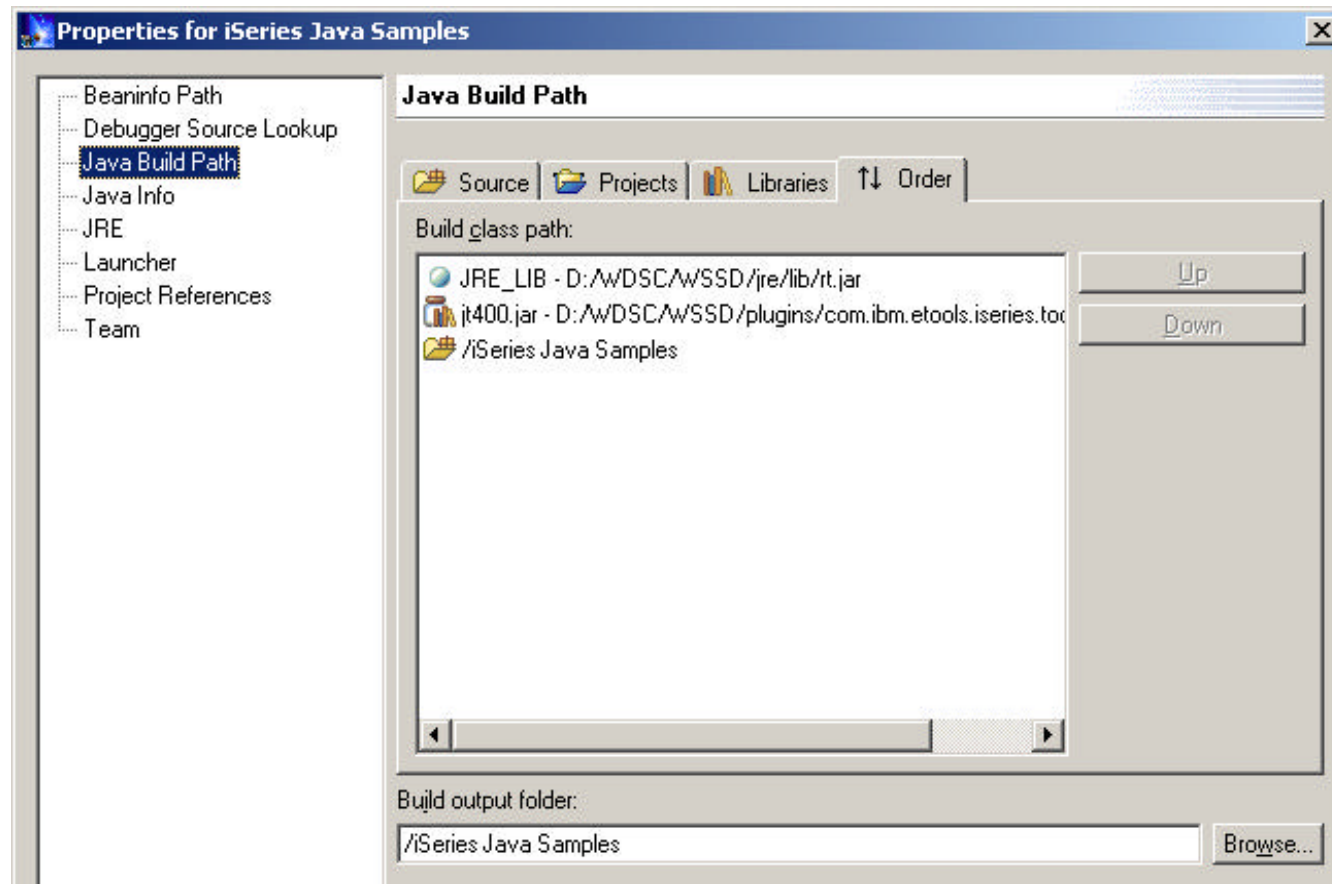
Add JAR file
from another
project

Add JAR file
from the local
filesystem

■ Libraries

- ▶ These are NOT OS/400 libraries!
- ▶ Specify all JAR files required by your Java project
 - ✓ i.e. IBM Toolbox for Java

d:\WDSC\WSSD\plugins\com.ibm.etools.iseries.toolbox\runtime\jt400.jar



■ Order

- ▶ Specify the order the projects and libraries are included in the build path
- ▶ This is important if the different versions of the same class exists in to different entries



Java Development Tools



iSeries AD, IBM Toronto

▶ iSeries Additions



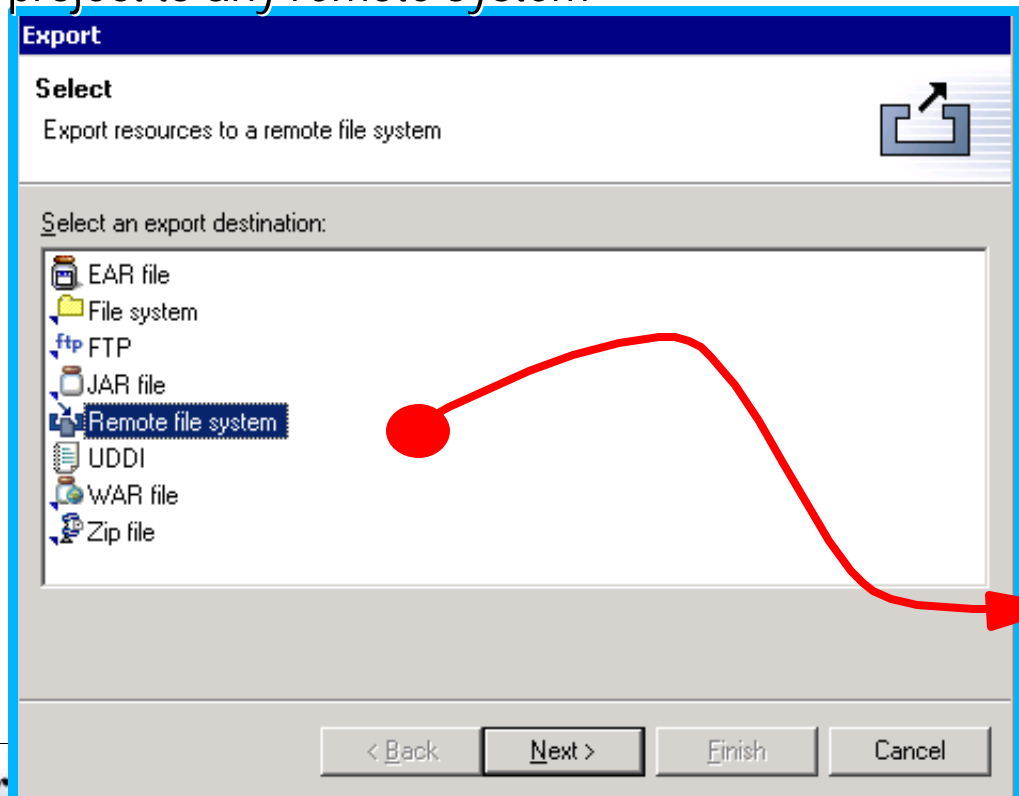
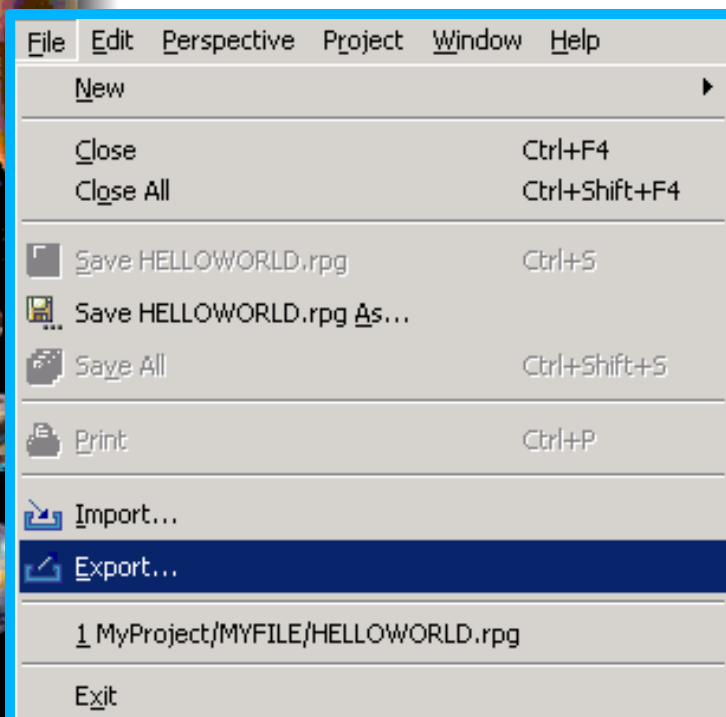
- ▶ iSeries extensions for Java
 - **Overview / Nutshell:**
 - ▶ Import and export from/to remote system
 - ▶ Remote Compile and Run
 - ▶ Remote Debug
 - ▶ Program Call wizard
 - ▶ Toolbox for Java built-in
 - ✓ As an Eclipse Java project (i.e.: easy to add to classpath of your project!)
 - ✓ Integrated help
 - ✓ Runnable samples
 - ▶ **Supplied Java-beans**
 - ✓ Swing GUI beans (dspf-like functionality)
 - ✓ DFU beans (database access)
 - ✓ Object list beans (library, object, member, field, record lists)



▶ iSeries extensions for Java

● Import and export

- ▶ Builds on base support for import/exporting files/jars/zips
- ▶ Add support to import/export files from remote system
 - ✓ Import any files from any remote system (iSeries/Unix/Linux/Windows)
 - ✓ Eg: IFS on iSeries; Linux LPAR on iSeries
 - ✓ Export files from any project to any remote system



Java Tools for iSeries



Export

Remote file system
Export resources to a remote file system.

Select the resources to export:

- DefaultEAR
- My Project
- PhilDemo
- PhilJavaDemo
 - customer
 - demo
- RemoteSystemsConnections
- RemoteSystemsTempFiles
- Servers

Customer.class
 Customer.java

Select Types... Select All Deselect All

Select the export destination:
Folder: My iSeries:/400home/zdemo

Options:
 Overwrite existing files without warning
 Create directory structure

Select files to export

Remembers previous export folders

Browse For Folder

Select a folder

/400home/zdemo

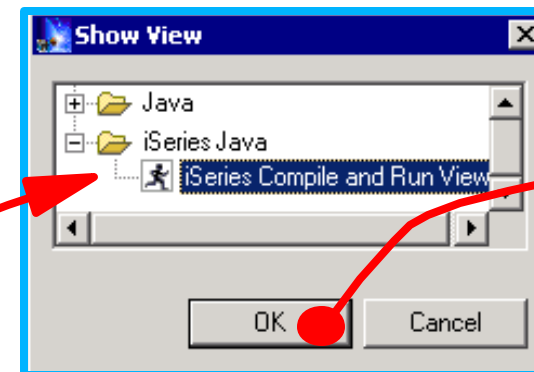
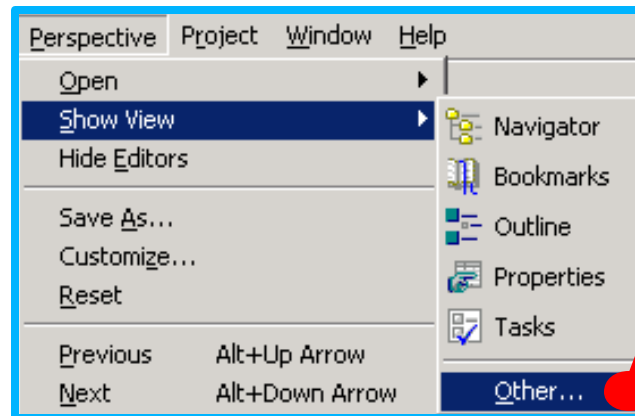
- New Connection...
- My iSeries
 - /QDLS
 - /QFileSvr.400
 - /QNetWare
 - /QNTC
 - /QOpenSys
 - /QOPT
 - /QSYS.LIB
 - 400home
 - LiTest7
 - wasservlets
 - zdemo**
 - 400home1
 - alfonso
 - applets
 - cheng
 - chenali





▶ Remote Compile and Run

- ▶ After exporting Java files to iSeries, you:
 - ✓ Transform them there using CRTJVAPGM
 - ✓ Run them there, see results via Console



Java Tools for iSeries



The screenshot displays the Eclipse IDE interface with three overlapping dialog boxes:

- iSeries Compile and Run View:** Shows the working folder as `My iSeries:/400home/zdemo/customer` and the option set as `Option set 1`. The `Run, debug remotely` tab is selected, with the `Create iSeries Java program` button highlighted by a red circle and a red arrow pointing to the next dialog.
- Select iSeries Options:** A list of option sets is shown, with `Option set 1` selected. Other options include `Option set 2`, `Option set 3`, `Option set 4`, and `Option set 5`.
- Run, debug remotely - Create iSeries Java program:** A dialog box for selecting classes. The `customer` folder is visible in the `Choose classes` list. Buttons for `CRTJVAPGM`, `Stop`, and `Clear Log` are present on the right. A `Log` field is at the bottom.

At the bottom left, the Eclipse logo and the file name `ls.prz` are visible.



▶ Remote Debug

- ▶ Launches debugger for iSeries Java
 - ✓ Same debugger as RPG, CBL, CL, C, C++

The screenshot displays the IBM Distributed Debugger interface. The main window shows the source code for a Java class named `/jsbq/EWJSBQ02`. The code includes a `do` loop and a `changeSomething()` method. A red circle highlights the `Debug ...` button in the bottom right corner of the interface, with a red arrow pointing to it from the right side of the image.

Source Code:

```

Thread 2:*CLASS /jsbq/EWJSBQ02
66     do{
67
68         temp++;
69         temp += 5;
70
71     }while(count++ <= 5);
72
73     // return statment
74     return;
75 }
76
77
78 → public void changeSomething(){
79
80     // local to method, they are not me
81
82     int a = 2;
83     String szString = "EWJSBQ02";
84
85     // implicit return
86 }
    
```

Breakpoints:

- Line Breakpoints
 - Source: *CLASS /jsbq/EWJSBQ02
 - Source: *CLASS /jsbq/EWJSBQ02
 - Source: *CLASS /jsbq/EWJSBQ02

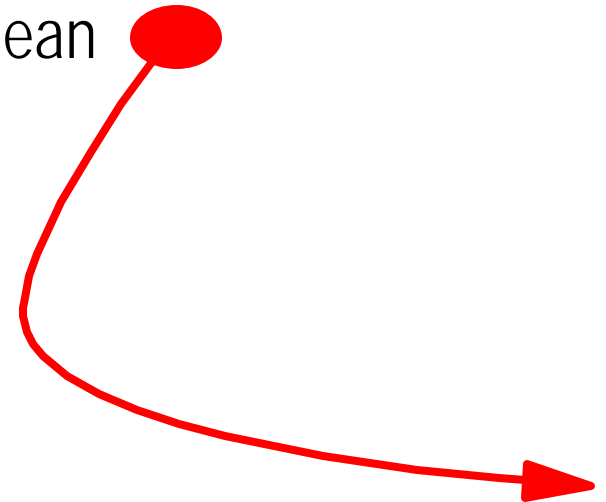
Monitors / Locals:

- Expression Mo...
 - count = Java variable is not available.
 - sb
 - szString = null



✓ Program Call Wizard

- **Given signature of *PGM or *SRVPGM proc:**
 - ▶ Generate Java Bean to call it
- **Steps:**
 - ▶ 1. Select Java/Web/WebFacing project
 - ▶ 2. Select package
 - ▶ 3. Use File->New->Other...
 - ✓ ...iSeries Java Program Call Bean





Java Tools for iSeries



iSeries AD, IBM Toronto

New

Select

This wizard generates Java beans to call iSeries programs or procedures.

- CVS
- Data
- Java
- Plug-in Development
- Remote File Transfer
- Server
- Simple
- Web
- Web Services
- WebFacing
- XML
- iSeries**
- Examples

Program Call Bean

Program Call Wizard

Define program definitions for your program or procedure

Program call definitions

- MyProgram

Edit Program

Java bean name: MyProgram

Program object: GETDLVSTS

Library: PHILDEMO

Program type: *PGM

Entry point:

Return type: void

Thread safe: false

Source location:

Type info, press OK



ls.prz

IBM @server. For the next generation of e-business.

Enter Parameter Info



Add Program... Add Parameter... Add Structure...

Define parameters

Add Parameter

Parameter name:

Data type:

Structure name:

Length:

Precision:

Count:

Usage:

Initial value:

Advanced...

Specify database reference field

Show database field definition

> 0 for array

Define and name structure

Add Structure

Structure name:

Count:

Usage:

Output size:

Min. version:

Max. version:

Offset:

Offset from:

Be sure to press OK for each parameter / structure

Builds up parameter list

Program call definitions

- MyProgram
 - retCode
 - nbrCodes
 - dlvCodes

Next >

Final Page



Program Call Wizard

Create iSeries Program Call Java bean and PCML file
Specify the location where the Java bean and PCML file to be generated.

Folder: **Where to generate Java class** Browse...

Package: Browse...

PCML file name: **Where to generate PCML file**

Generate Java beans and a PCML file for:

Java application

Web services **Select Web Services to generate Java code optimized for use in the Web Services wizard**

These files will be generated by the wizard:

- MyProgram.pcm1
- MyProgram.mpcm1
- MyProgramServices.java
- MyProgram.java

Summary of files that will be generated

The class path of the specified project will be automatically updated.
Generated Java beans require the following JAR files at compile time and run time:

- ECLIPSE_HOME/plugins/com.ibm.etools.iseries.toolbox/runtime/t400.jar
- ECLIPSE_HOME/plugins/com.ibm.etools.iseries.webtools/lib/wdt400rt.jar
- ECLIPSE_HOME/plugins/com.ibm.etools.websphere.runtime/lib/xerces.jar
- ECLIPSE_HOME/plugins/com.ibm.etools.websphere.runtime/lib/j2ee.jar

To change parameters later:

- Return to wizard
- Use Import PCML... button

Runtime files you will need to deploy with your code

< Back Next > **Finish** Cancel

Generated Java Bean



```

package customer.demo;

import java.math.*;
import com.ibm.as400.access.*;
import com.ibm.connector.as400.ProgramCallBean;
import com.ibm.connector.as400.ProgramCallRecord;
import com.ibm.connector.as400.ProgramCallStruct;

public class MyProgram extends ProgramCallBean
{

    public MyProgram()
    {
        super();
        setPcm1Name("customer.demo.MyProgram");
        setProgramName("MyProgram");
    }

    public Integer getRetCode()
    {

```

Wizard adds required files to project classpath

Example using gen'd bean

```

package customer.demo;
public class TestMyProgram
{
    public static void main(String[] args)
    {
        MyProgram test = new MyProgram();
        test.setConnectionData("MYSYSTEM", "MYUSERID", "MYPASSWORD");
        test.setLibraryList(new String[] {"PHILDEMO"});
        test.invoke();
        int nbrResults = test.getNbrCodes().intValue();
        System.out.println("Nbr Codes: " + nbrResults);
        for (int idx=0; idx<nbrResults; idx++)
        {
            System.out.println(" code " + idx + ": " + test.getDlvCodes(idx).trim());
        }
    }
}

```

- ▶ **setConnectionData(...)**
 - ▶ to identify signon info
- ▶ **setLibraryList(...)**
 - ▶ to set runtime library list
- ▶ **setXXX(...)**
 - ▶ to set input parameters
- ▶ **invoke()**
 - ▶ to run program
- ▶ **getXXX(...)**
 - ▶ to get output parameters



- ▶ GUI Beans for iSeries
 - **Java Beans supplied for DDS-like field error-checking and formatting:**
 - ▶ **JFormattedTextField (smart entry field)**
 - ✓ Error checking based on data type, length, decimals
 - ✓ DDS-like validity checking: range or comparison
 - ✓ Editcode or editword formatting and masking(!)
 - ✓ Auto-advance
 - ▶ **JFormattedLabel (smart label constant)**
 - ✓ Editcode or editword formatting
 - ▶ **JFormattedComboBox (smart dropdown)**
 - ✓ Combo of JFormattedTextField and JFormattedLabels
 - ▶ **JFormattedTable (smart multi-column list)**
 - ✓ A "subfile" or multi-column list box
 - ✓ Each cell is a JFormattedTextField or JFormattedLabel or JFormattedComboBox



▶ GUI Beans Examples

- **File->New->Other->Examples->iSeries**

- ▶ JFormattedTable

ItemNo	Model	Size	Price
00001	2000-AB-1	small	100
00002	2000-AB-2	small	200.50
00003	2000-AB-3	medium	369
00004	2000-AB-4	big	700
00005	2000-AB-5	small	405
00006	2000-AB-6	small	110
00007	2000-AB-7	small	230.49
00008	2000-AB-8	medium	268
00009	2000-AB-9	big	732

- ▶ JFormattedText

EQUITY ORDER ENTRY

Symbol:

Price:

Quantity:

Account No.:

Password:

Contact Phone:



- ▶ DFU Beans for iSeries (non GUI)
 - **Java Beans supplied for querying data from DB2/400, using direct record access (versus SQL):**
 - ▶ **RecordIOManager:**
 - ✓ Query, add, update or delete records in a database
 - ✓ Vastly simplifies task of writing database access code using Toolbox for Java classes
 - ▶ **ListManager:**
 - ✓ Maps list of records from RecordIOManager to a Swing GUI list
 - ✓ Makes it easy to populate GUI list with DB2/400 data
 - ▶ **FormManager**
 - ✓ Maps details of a single row from RecordIOManager to a single GUI dialog page
 - ✓ Makes it easy to populate GUI form with DB2/400 data



▶ DFU Beans Examples

● File->New->Other->Examples->iSeries

▶ ListManager

ListManagerJFormattedTableSample

PARTNO	MODEL	PARTPRI	PARTMSR	PARTDIS	PARTSHIP
00562	AR-10	26.10	29.00	Y	1991-05-10
00562	AR-11	32.04	36.00	Y	1992-12-02
00562	AR-12	7.92	9.00	N	1994-05-07
00562	AR-13	11.31	13.00	N	1991-09-25
00074	AR-1	35.64	36.00	N	1990-07-07
00074	AR-11	31.15	35.00	N	1990-01-08
00074	AR-12	29.92	34.00	N	1994-03-19
00074	AR-14	39.56	46.00	N	1993-12-22
01807	AR-10	24.30	27.00	N	1994-06-16
01807	AR-11	20.47	23.00	N	1993-10-13

readAllRecords

FormManager with a multiple record format logical file

Order: DEF456

Order Date: 1998-03-24

Customer Order: aaaaaaa

Customer: bbbbbbbbbbbb

Ship Via: ccc

Ship To: ddddddd

Price: 55555.33

Goods: 6666

▶ FormManager



- ▶ List Beans for iSeries (non-GUI)
 - **For querying lists of libraries, objects, members, records or fields**
 - ▶ **AS400ListLibraries**
 - ✓ List libraries given simple or generic or special name
 - ▶ **AS400ListObjects**
 - ✓ List objects given simple or generic library and object name
 - ✓ Can also subset by type and attr (one or multi, simple or generic)
 - ▶ **AS400ListMembers**
 - ✓ List members given simple or generic library, file, member name
 - ✓ Can also subset by member type (one or multi, simple or generic)
 - ▶ **AS400ListRecords**
 - ✓ List records given simple or generic library, file, record name
 - ▶ **AS400ListFields**
 - ✓ List fields given simple or generic library, file, record, field name
 - **Four levels of detail possible per object**



Java Development Tools



iSeries AD, IBM Toronto

▶ Summary

Java Tools Not There



- ▶ Not all VAJava function is available yet:

- × Visual Composition Editor

- × DataBase Beans

- ✓ Actually, there are replacements! Just no visual metaphor

- ✓ See com.ibm.db.beans: DBSelect, DBProcedureCall, DBModify

- ✓ There is also a Database access wizard in Web Tools

- × SQLJ support

- × Stored Procedure Builder

- × Live editing while debugging

- ▶ For VAJava migration...

- ▶ www.ibm.com/software/wsdd

- ✓ "VisualAge for Java coexistence with WebSphere Studio Application Developer"

- ✓ http://www7b.boulder.ibm.com/wsdd/library/techarticles/0110_searle/searle.html

Summary



- ▶ Java Development Tooling included in WDSC
 - **Part of WebSphere Studio Site Developer Advanced**
- ▶ Java Perspective provides optimized view for Java development
 - **Java Editor + Java Views = Easier Development**
 - **Integrated Compile / Run / Debug**
 - **iSeries Additions for using Java with iSeries**
 - **Lots more ...**
 - ▶ Team development
 - ▶ Integrated, online help system
 - ▶ Extendable platform
 - ✓ You can add your own tools
 - ▶ Integrated Java, Web, XML, iSeries development
- ▶ Enjoy!



- ▶ Eclipse.org
 - Website for the open source Workbench
- ▶ WDSC
 - ibm.com/software/ad/wds400/
- ▶ WebSphere Developer Domain
 - ibm.com/websphere/developer
- ▶ IBM Developer Domain
 - ibm.com/developer
- ▶ Java
 - java.sun.com



Trademarks & Disclaimers



iSeries AD, IBM Toronto

© IBM Corporation 1994-2002. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.

Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.