LAB : Introduction to VisualAge for Java Enterprise Toolkit for AS/400

Maha Masri
IBM Toronto Laboratory

and

Pramod Patel
IBM Toronto Laboratory

and

George Voutsinas
IBM Toronto Laboratory

Fall COMMON, 1999

© Copyright IBM Corp. 1999
Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Lab 2: Introduction to VisualAge for Java Enterprise Toolkit for AS/400 2-4
Overview 2-4
Goal 2-4
Tool 2-4
Exercise 1: Overview of Enterprise Toolkit for AS/400 Examples 2-5
Introduction 2-5
Goal 2-5
Introduction to VA Java Integrated Development Environment (IDE) 2-6
Starting VA Java Integrated Development Environment (IDE) 2-6
Overview of IBM Enterprise Toolkit for AS/400 2-8
Starting IBM Enterprise Toolkit for AS/400 On-line Help 2-10
Starting AS/400 Toolbox for Java On-line Help 2-12
Running the Getting Started Example 2-13
Exercise 2: Creating Classes for the Inventory Application 2-17
Introduction 2-17
Goal 2-17
Converting the Inventory List Display File 2-18
Modify the Converted Screen 2-20
Creating a Subfile from a Database File 2-23
Creating a Program Call Class to Call an AS/400 Print Inventory Function 2-25
Exercise 3: Using the Beans you Created 2-27
Introduction 2-27
Goal 2-27
Creating a Graphical User Interface (GUI) Using the VCE 2-28
Creating the Labels 2-29
Adding the Inventory List Class as a Bean to the Application 2-30
Adding the Inventory Part Details Class as a Bean 2-31
Adding the Print Inventory Class as a Bean 2-32
Adding a List Button for the Inventory List 2-33
Adding a Status Bar 2-33
Exercise 4: Adding Connections to Enable Functions 2-35
Introduction 2-35
Goal 2-35
Opening the File for Reading and Writing 2-36
Closing the File 2-38
Displaying Message “Getting data from AS/400, please wait..” 2-39
Removing All Records from the Inventory List 2-39
Populating the Inventory List with Records 2-39
Displaying the Message “Data retrieved” 2-40
Testing What You Have Done So Far 2-41
Copying Methods from Another Project 2-42
Populate the Part Details when the Part is Selected in the Inventory List 2-42
Adding Buttons for the Part Details 2-43
Enabling All the Buttons 2-43
Testing What You Have Done So Far 2-44

 © Copyright IBM Corp. 1999
Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 5: Add, Update, Delete Records (OPTIONAL -- ADVANCED SECTION) 2-45
Introduction 2-45
Goal 2-45
Enabling the Add Button 2-46
Displaying Message “Record added” 2-46
Enabling the Update Button 2-46
Enabling the Delete Button 2-46
Enabling the Clear Button 2-47
Testing What You Have Done So Far 2-48
Exercise 6: Print Record Information (OPTIONAL -- ADVANCED SECTION) 2-49
Introduction 2-49
Goal 2-49
Making an AS/400 Connection to call the Print Function 2-50
Sending the Part Number to the Print Function 2-50
Running the Print Inventory Program on AS/400 2-51
Printing the Program Completion Message 2-51
Testing Your Completed Application 2-52
Appendix 1. Adding a Feature into the Workbench 2-54

 © Copyright IBM Corp. 1999
Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Lab 2: Introduction to VisualAge for Java Enterprise Toolkit for AS/400

Overview

This lab will provide an overview of:

- The **VisualAge for Java** (VA Java) Integrated Development Environment (IDE).
- The Visual Composition Editor
- The basic concepts of VisualAge for Java projects, packages and classes.
- The **IBM Enterprise Toolkit for AS/400 (ET/400)** feature of **VisualAge for Java, Enterprise Edition Version 3.0** (VA Java).

You will build some basic GUI components by dragging and dropping the beans onto the workspace. In the exercises, the concept of component wiring will be introduced. By the end of this lab, you will have created an application capable of accessing AS/400 data and programs.

Goal

In this lab, you will be introduced to the **getting started** example shipped as part of **Enterprise Toolkit for AS/400** feature. You will learn how to use the **Convert DisplayFile, Create Subfile**, and **Program Call** SmartGuides. These SmartGuides (wizards) generate Java code that makes use of the **AS/400 Toolbox for Java** record-level access classes to access data from AS/400 database files and the program call classes. You will develop a client based GUI (*Graphical User Interface)* application that will make use of the code generated by the SmartGuides to add, update, delete and print records from a database file on the AS/400.

Tool

This lab uses **VisualAge for Java, Enterprise Edition 3.0** to develop Java code. You will use the **Integrated Development Environment** (**IDE**), the **Visual Composition Editor** (**VCE**) commonly known as the **Visual Builder**, the **Enterprise Toolkit for AS/400** feature and the **AS/400 Toolbox for Java** (shipped as part of the **Enterprise Toolkit for AS/400** feature). On the AS/400 host, you need to install the ADTSLAB library by applying one of the following PTF (*Program Temporary Fix*) to the Application Development ToolSet/400 (ADTS/400) Licensed Program Product (LPP). For this Lab, we have done this for you.

OS/400 Release	LPP Number	PTF Number
V3R2M0	5763PW100	SF45556
V3R6M0	5716PW100	SF45554
V3R7M0	5716PW100	SF51022
V4R2M0	5769PW100	SF49832
V4R3M0	5769PW100	SF49832

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 1: Overview of Enterprise Toolkit for AS/400 Examples

Introduction

This section will introduce you to:

- **VisualAge for Java Integrated Development Environment (IDE)**
- **IBM Enterprise Toolkit for AS/400** Feature
- Online help on **IBM Enterprise Toolkit for AS/400** feature
- Online help on **AS/400 Toolbox for Java**
- Some of **IBM Enterprise Toolkit for AS/400** Examples

Goal

At the end of this exercise, you will be able to:

- Start the**VisualAge for Java Integrated Development Environment (IDE)**
- Understand the **IBM Enterprise Toolkit for AS/400** feature
- Start help on the **IBM Enterprise Toolkit for AS/400** feature
- Start help on the **AS/400 Toolbox for Java**
- Run the **IBM Enterprise Toolkit for AS/400** examples

What are we waiting for! Let us begin...

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Introduction to VA Java Integrated Development Environment (IDE)

NOTE: **"Mouse Button 1"** is the left mouse button, and **"Mouse Button 2"** is the right mouse button. Click always means click **Mouse Button 1**, unless the mouse button is explicitly specified.

Starting VA Java Integrated Development Environment (IDE)

The **VA Java IDE** is also called **Workbench**. It is started as follows.

1. If there is no **Workbench** entry on the task bar at the bottom of your (screen) Windows desktop, then to start **VisualAge for Java:**

- Click on the **Start** button on the left side of the task bar.
- Select the **Programs -> IBM VisualAge for Java for Windows -> IBM VisualAge for Java.**

  After a short delay, the **Workbench** window pops up, and you will see a **"What do you want to Do"** dialog. Select the **Go to the workbench** radio button. Click **OK**.

2. If there is already a **Workbench** entry on the task bar at the bottom of your (screen) Windows desktop, then move the cursor to it and double click **Mouse Button 1** to restore the window and bring it into focus.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

If your **Workbench** window is not modified, then you will notice two horizontally divided panes. The top pane is a hierarchy browser displaying the project hierarchy, when the project page is selected. The bottom pane is a source browser displaying the description of the selected (focused) projects or packages on the top pane. If a class or a method is selected, then its source is displayed in the bottom pane.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Overview of IBM Enterprise Toolkit for AS/400

This section will show you how the **IBM Enterprise Toolkit for AS/400** feature is organized on the **VA Java Workbench**.

1. Click on the + symbol beside the **IBM Enterprise Toolkit for AS400** project to expand it. (If you see − symbol, then someone has already expanded it before you).

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

You will notice a list of packages under **IBM Enterprise Toolkit for AS400** project.

com.ibm.as400.access	AS/400 Toolbox for Java access classes
com.ibm.as400.demo.toolbox	AS/400 Toolbox for Java demo programs
com.ibm.as400.examples.toolbox	AS/400 Toolbox for Java on-line help examples
com.ibm.as400.vaccess	AS/400 Toolbox for Java visual access classes
com.ibm.ivj.et400.util	ET/400 runtime utility classes
com.ibm.ivj.examples.et400.displayFile	ET/400 Display File example
com.ibm.ivj.examples.et400.gettingStarted	ET/400 Getting Started example
com.ibm.ivj.examples.et400.jformattedBeans	JFormattedBeans examples
com.ibm.ivj.examples.et400.objectList	ObjectList bean example
com.ibm.ivj.examples.et400.subfile	ET/400 Subfile examples
com.ibm.ivj.examples.et400.subfileColumnHidden	Subfile column hiding example
com.ibm.ivj.examples.et400.subfileFocusRecordNo	Subfile record number example
com.ibm.ivj.examples.et400.toolboxDemo	Demo accessing various Toolbox resources
com.ibm.ivj.examples.et400.toolboxKeyedDataQueue	Keyed dataqueue example
com.ibm.ivj.examples.et400.toolboxProgramCall	Parameter passing example
com.ibm.ivj.examples.et400.toolboxRecordIO	Data access using record I/O examples

2. Click on any package to review its description in the bottom pane. Click on the symbol beside the package name to expand it to see all the classes it contains.

3. VisualAge for Java's Enterprise ToolKit for AS/400 includes tools and SmartGuides that allow you to perform the following tasks:
   - Export Java class and source files to the AS/400. (Source files are required to debug AS/400 Java applications from VisualAge for Java.)
   - Compile Java code optimized for the AS/400.
   - Run Java code as interpreted bytecode in the AS/400 Java Virtual Machine (JVM), or as compiled programs.
   - Run and debug AS/400 Java applications from the VisualAge for Java IDE.
   - Convert an AS/400 display file to Java AWT code.
   - Create a subfile from an AS/400 database file.
   - Create a program call class that calls an AS/400 program.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Starting IBM Enterprise Toolkit for AS/400 On-line Help

Before you can start your work you should know how to start the help in case you need it. **VA Java** makes it easy for you to start help. **VA Java** uses HTML (*HyperText Markup Language*) and relies on the default web browser of the workstation to display it. There are different ways you can start **VA Java** help.

- From the Help menu of the **Workbench**
- By pressing F1 from any SmartGuide (wizard)
- By clicking on IDE Help in Release Notes

In this section, you will start the help using the first method.

1. Select **Help -> Help Home Page** from the Help menu on the **Workbench** to open the main home page of the **VA Java** help. **VA Java** home page is displayed in the default web browser. Browse the topics.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

2. Click on the **Topics** link (button) on the left hand column. This will expand the **Topics** category. Note that this page has two vertical panes.
3. Click on the + symbol beside **Tutorial** link in the left pane to expand it.
4. Click on the + symbol beside **Enterprise ToolKits** in the left pane to expand it. You will notice a list of all the toolkits.

5. Click on the **Enterprise ToolKit for AS/400** to open its The Enterprise Toolkit For AS/400 web page. Browse the topics.
6. You can click on other links, such as, **Concepts, Tasks, References**, from the menu bar on top of the page to view them.

   Leave the browser running.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Starting AS/400 Toolbox for Java On-line Help

Since this lab and **Enterprise Toolkit for AS/400** make extensive use of the **AS/400 Toolbox for Java** classes, it is worthwhile knowing how to start the Toolbox help.

1. Move the focus onto the **Workbench** by clicking on the **Workbench** on the task bar at the bottom of your Windows desktop.
2. Select **Help -> Tools -> AS/400 Toolbox for Java** from the Help menu to open the main page of the toolbox help. Browse the topics.
3. Since you will be working with Record-level access classes, click on the symbol beside the **Access classes** (or on the **Access classes**) in the left column to expand the list.
4. Click on the **Record-level access** from the expanded list to open its web page. Browse the topics.

Leave the browser running.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Running the Getting Started Example

The purpose of this section is to make you familiar with the examples shipped with the **IBM Enterprise Toolkit for AS/400**. They are provided to make some of the concepts easier to understand and to help you become productive quickly. You will concentrate on the getting started example here. This example demonstrates how to change the visual subfile attributes, such as colors and horizontal and vertical separator lines.

1. Move the focus onto the **Workbench** by clicking on the **Workbench** on the task bar at the bottom of your Windows desktop.
2. Click on the + symbol beside **com.ibm.ivj.examples.et400.gettingStarted** package to expand it. Browse the description in the bottom pane.

Before you start running the example, you may look at the help for the example.

1. Click on the **IBM VisualAge for Java - Netscape** on the task bar at the bottom of your Windows desktop to bring the browser in the focus.
2. Click on the + symbol beside **Samples** in the left column of the web page. The Samples category is expanded.
3. Click on the + symbol beside **Enterprise Toolkit for AS/400** in the left column of the web page.
4. Click on the + symbol beside **Samples** in the left column of the web page.

Click on the **Create a Java Client/Server Application** link. A description of the tutorial is displayed on the right frame of the browser.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Now you are ready to run the example.

1. Move the focus to the **Workbench** by clicking on the **Workbench** on the task bar at the bottom of your Windows desktop.
2. Select the **IBM Enterprise Toolkit for AS400** project and expand it by clicking on the '+' symbol.
3. Select the **com.ibm.ivj.examples.et400.gettingStarted** package and expand it by clicking on the '+' symbol.
4. Click on the **InventoryList** class in the gettingStarted package. This example runs a Java client application that accesses AS/400 data and resources.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

5. Click on the symbol on the toolbar to run it. (Alternately, once the class is selected, you can click **Mouse Button 2** to bring up a pop-up menu and then select "**Run -> Run main**" from it).

6. Click on the **List** button. The **Signon to AS/400** dialog is displayed.

Enter the System name, Userid and Password provided to you leaving the **Default User ID** and **Save password** check boxes checked. Click **OK** on the dialog. The subfile is populated with data.
The actions performed during this step are: an AS/400 object is created and a connection is made to the AS/400 host, the database file is opened, all records are retrieved and the visual subfile is populated.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

7. Select various records in the Inventory List. Click on the add, update, delete, and clear buttons to see how the application works.

Click on the Exit button (the X in the upper right-hand corner of the window) to close this example and return to the Workbench.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 2: Creating Classes for the Inventory Application

Introduction

In this exercise, you are going to take a 5250 screen on an AS/400 and convert it to a Java AWT screen. You will then create a subfile in Java to access and display data from the PRODDTL file on an AS/400. Finally, you create a program call class to call an RPG program on an AS/400 to print a record you selected. All of this is accomplished using the Enterprise ToolKit for AS/400 (ET/400) feature of VisualAge for Java.

Goal

At the end of this exercise, you should be able to:

- Use the Convert DisplayFile SmartGuide to generate code for display file records
- Use the Create Subfile SmartGuide to generate the subfile code from a database file
- Use the Program Call SmartGuide to generate code to call an AS/400 program

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Converting the Inventory List Display File

You will begin by converting the Inventory List display file record using the Convert SmartGuide.

A project **VA Java for AS400 Labs** with **VJ400LAB2** package in it is already created for you. Select the project **VA Java for AS400 Labs** then the package **VJ400LAB2**. Right click on the

package to display a pop-up menu. Click on **Manage** then **Create Open Edition** (if not greyed out) to ensure that you can add new classes to it.

Select the package **VJ400LAB2** again if necessary. Right Click on **Tools** then **ET/400** then **Convert Display File...**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

On the Convert SmartGuide specify the following information:

1. A server such as **AS400A** (not TORAS015) from the **Server** list
2. A library called **JAVAxx** (not ADTSLAB) in the **Library** field
3. Expand the tree to see a list of the display files
4. Double-click on display file WRKINVFRM to display its record list
5. Click on the parts detail record INVDETL

Leave all the other fields with their default values and select **Next** until you can select **Finish.** Java classes are generated and placed under the selected project and package in the Workbench.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Modify the Converted Screen

You will take the generated bean and bring it into the Visual Composition Editor (VCE) and visually delete the text and buttons which are no longer needed. You will also change the properties of the entry fields to accept input.

1. Select the generated class INVDETL in the package **VJ400LAB2** under the project **VA Java for AS400 Labs**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

2. Double click on this class. The browser opens on the methods in the new class.
3. Click on the Visual Composition tab. Select **Bean** then **Construct Visuals from Source** from the **Bean** pull-down menu. Click **Proceed** to replace the visual contents. The generated class is presented as separate components. Resize the frame as necessary to show all the components so that you can click on them. You can click on each component and delete the components you do not want to keep.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

4. Press the Shift key and select each of the following INVDETL components, then right click and from the pop-up menu select **Delete**:
   - Mode: and the entry field for Mode
   - CelDial - Part Detail
   - SYSNAME
   - Press Enter to add part
   - F12=Cancel

   Press the Shift key and select the following ButtonPanel components, then right click and from the pop-up menu select **Delete**:
   - CA12
   - HELP

5. Right click on the **Part number** entry field and select **Properties**. Click the field next to the **enabled** value and select **True** from the list. This enables the text field to be input capable. Leave the Properties window open. Repeat this step for each field.
6. Close the Properties window.
7. Save the work you have done. To save your bean, from the **Bean** menu select **Save Bean**.
8. Close the Visual Composition pane.

NOTE: It's O.K. to find **X** in front of your class. It is a symbol to tell you that it cannot find some methods the class refers to as we have just deleted some of them. This would not affect your code.

Next you will create a subfile class for the Inventory List.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a Subfile from a Database File

Click on the package **VJ400LAB2**. Click on **Tools** then **ET/400** then **Create Subfile...**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

On the Create Subfile SmartGuide specify the following information:

- **Create a new subfile class** radio button, press **Next**
- A server _____ from the **Server** list (AS/400 system name)
- A database file PRODDTL in the **File** field
- A library called JAVAxx in the **Library** field
- Default settings for the class and package names, press **Next**
- Select and add columns PARTNO and PARTD to the columns create list

Leave all the other fields with their default values and select **Next** until you can select **Finish.** Java classes are generated and placed under the selected project and package in the Workbench.

You have finished creating a subfile class for the Inventory List. Next you will create a program call class to call the print inventory function to print an inventory item in the Part Details.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a Program Call Class to Call an AS/400 Print Inventory Function

Select the package, **VJ400LAB2**. Right click on the package to display a pop-up menu then click on **Tools** then **ET/400** then **Create Program Call...**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

On the Create Program Call SmartGuide specify the following information:

- **Create a new program call class** radio button, press **Next**
- A server _____ from the **Server** list (AS/400 system name)
- A program called PRTINV in the **Program** field
- A library called JAVAxx in the **Library** field
- A name for the generated class such as PRTINV in the **Class** field, press **Next**

- Click 'insert' button to add a parameter with the name partno, datatype of character and size of 5, press **OK**.

Leave all the other fields with their default values and click **Finish.** Java classes are generated and placed under the selected project and package in the Workbench.

Next you will add the classes generated by the SmartGuides as beans in the Visual Composition Editor to build your client interface visually.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 3: Using the Beans you Created

Introduction

In this exercise you add beans for the Inventory List, Part Details and call to the print inventory function visually to the InventoryList class you will be creating. This exercise also introduces you to the concept of visual construction of visual and non-visual parts/beans.

Goal

At the end of this exercise you should be able to:

- Create a class
- Create the Labels for the **Part Inventory** application, the **Inventory List** bean, and the **Parts Detail** display file bean
- Add the Inventory List subfile bean
- Add the Part Details display file bean
- Add the Program Call bean (PRTINV1)
- Add a push button for the Inventory List
- Create a status label for the application

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a Graphical User Interface (GUI) Using the VCE

Select the **VA Java for AS400 Labs** project and then the **VJ400LAB2** package. Select Add from the Selected pull-down menu. Then select Class.... On the Create Class SmartGuide specify the following information:

- **InventoryList** in the **Class Name** field
- **com.sun.java.swing.JFrame** in the **Superclass** field
- Select the **Compose the class visually** check box

Click **Finish**. The SmartGuide disappears and the Visual Composition Editor appears.
To resize the frame window to cover three quarters of the work area , click on the title bar of the JFrame and then drag the corners of the frame (Don't click on the content pane of the frame).

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Creating the Labels

Use a Label bean to provide a title for the screen (**Part Inventory**).

1. Select the Swing category from the Beans Palette list.
2. Click on the **JLabel** bean and drop the **JLabel** bean onto the upper area of the frame by moving the '+' cursor to the desired position and single click using the left mouse button. **Note:** If you move the cursor to any bean on the Beans Palette, help will show you the name of the bean.
3. Right click on the **JLabel1** bean and select **Properties**.
4. Click the field next to the **text** value and type **Part Inventory**.
5. Click the field next to the **horizontalAlignment** value and select **CENTER** from the list.
6. Click the field next to the **font** value and click the list to open the Font window. Select **Bold** from the **Style** list.
7. Click **OK**. Leave the Properties window open.
8. Use the handles to increase the width of the label bean to accommodate the text.

Use a Label bean to provide a title for the **Inventory List**.

1. Click on the **JLabel** bean and drop the **JLabel** bean onto the upper left area by moving the '+' cursor to the desired position and single click using the left mouse button.
2. Right click on the **JLabel2** bean.
3. On the Properties window change the **text** value to **Inventory List** and the **horizontalAlignment** value to **CENTER**.
4. Use the handles to increase the width of the label bean to accommodate the text.
5. Leave the Properties window open.

Use a Label bean to provide a title for the **Parts Details**.

1. Click the **JLabel** bean and drop the **JLabel** bean onto the upper right area by moving the '+' cursor to the desired position and single click using the left mouse button.
2. Right click on the **JLabel3** bean and on the Properties window change the **text** value to **Part Details** and the **horizontalAlignment** value to **CENTER**.
3. Resize the label as necessary to show the text.
4. Close the Properties window.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Now it is the time to add the beans generated using ET/400 feature in exercise 1 .

Adding the Inventory List Class as a Bean to the Application

This class is the Inventory List generated by the Convert SmartGuide which contains a parts inventory list.

1. Select **Choose bean...** from the Beans Palette.
2. Click on the **Browse...** button and Choose Bean window appears.
3. Key in **PRODDTL** in the Pattern field. Select the class from the VJ400LAB2 package. Click **OK**. The **VJ400LAB2.PRODDTL** name has been added to the Choose Bean window as the Class name. Do not fill in the Name. Click **OK** to add the bean.
4. Move the crosshairs inside the frame and left click to drop the bean.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

The Inventory List PRODDTL bean is added to the frame. Resize the frame as necessary to show the bean.

Adding the Inventory Part Details Class as a Bean

This class is the detail form generated by the Create Subfile SmartGuide which contains the details of one inventory part.

1. Select **Choose bean...** from the Beans Palette.
2. Click on the Browse... button and enter INVDETL as the class name. Select the class from the VJ400LAB2 package. Click **OK**. The **VJ400LAB2.INVDETL** name has been added to the Choose Bean window as the Class name. Click OK to add the bean.
3. Move the crosshairs inside the frame and left click to drop the bean.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

The Inventory Part Details is now added to the frame. Resize the frame as necessary to show the bean.You are now ready to add the Print Inventory Bean.

Adding the Print Inventory Class as a Bean

This class contains the Java code to call the ILE RPG program PRTINV and passes a parameter partno to print the part number detail on AS/400. If the print is successful two spool files result. They are QSYSPRT and INVPRINT. QSYSPRT contains information indicating the program call was successful. INVPRINT contains the part detail.

1. Select **Choose bean...** from the Beans Palette.
2. Click **Browse...** and enter a few characters of the class name. Select the class. Click **OK**. The **VJ400LAB2.PRTINV** name has been added to the Choose Bean window as the Class name. Click **OK** to add the bean.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

3. Move the crosshairs to the bottom of the frame in the free-form surface and left click to drop the bean. Scroll the window or resize the frame as necessary to show the bean.

Adding a List Button for the Inventory List

1. Click on the **JButton** bean and drop the **JButton** bean just below the PRODDTL1 bean on the left of the frame.
2. Right click on the button. Select Properties from the pop-up menu.
3. On the Properties window, click on the field next to the **text** value and type List.
4. Click on the field next to the **beanName** value and type JButtonList.

Adding a Status Bar

Use another Label bean to provide a status bar to display information messages for the application.

1. Select the **JLabel** bean. Drop the **JLabel** bean on the bottom area of the frame. Use the handles to increase the width of the **JLabel** bean to the size of the frame.
2. Right click on the **JLabel** bean. Select Properties from the pop-up menu.
3. On the Properties window, click the field next to the **beanName** value and type JLabelStatus.
4. Blank out the field next to the **text** value.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

5. Click **Show expert feature** check box. Click the field next to the **Border** and click the button. Select **Bean Implementing Interface** radio button. Select **BasicFieldBorder** from the combo box by clicking on the drop down arrow. Click **OK**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 4: Adding Connections to Enable Functions

Introduction

In this exercise you add connections to open the database file for reading and writing, removing records, populating records and adding and enabling buttons for the Part Details.

Goal

At the end of this exercise you should be able to:

- Open a database file for reading and writing
- Display messages on the status line
- Remove records from the Inventory List
- Copy methods from another project
- Populate the Part Details
- Add buttons for the Part Details
- Enable all the Part Details buttons

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Opening the File for Reading and Writing

When you open the window you want to open the data file automatically and when you close the window the file will be closed as well.

1. Click on the title bar of the application frame, then right click. (NOTE: The bottom left-hand corner of the VCE will display the selected bean: **InventoryList [com.sun.java.swing.JFrame] selected**.) In the pop-up menu that appears, select **Connect** then **Connectable Features...**.

2. Click **Event** radio button. Select the event **windowOpened**. Click **OK**. The mouse pointer changes, indicating that you are in the process of making a connection.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

3. Move the mouse pointer to the Inventory List (PRODDTL1) bean then right-click.

4. Select **Connectable Features...** from the pop-up menu that appears. Select the method **OpenSequentialFileReadWrite().** Click **OK**. This connection opens the database file for reading and writing records when the application window opens.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

NOTE: When you select the green connection line, the bottom left-hand corner of the VCE will display: **connEtoM1: (InventoryList,windowOpened --> PRODDTL1,openSequentialFileReadWrite()) selected**.

Closing the File

1. Click on the title bar of the application frame, then right click. In the pop-up menu that appears, select **Connect** then **Connectable Features...**. Select the event **windowClosing**. Click **OK**.
2. Move the mouse pointer to the Inventory List (PRODDTL1) bean then right-click. Select **Connectable Features...** from the pop-up menu that appears. Select the method **closeSequentialFile().** Click **OK**. This connection closes the database file for reading and writing records when the application window closes.

NOTE: When you select this green connection line, the bottom left-hand corner of the VCE will display: **connEtoM2: (InventoryList,windowClosing --> PRODDTL1,closeSequentialFile()) selected**

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Displaying Message "Getting data from AS/400, please wait.."

1. Select the **List** button, click the right mouse button, and select **Connect** from the pop-up menu.
2. Position the mouse pointer over **actionPerformed** and left click.
3. Make a connection from the **List** button to the Status label bean and click on the left mouse button and a pop-up menu appears. Select **Connectable Features...** and the End connection to window appears. Select the **setText(String)** method. Click **OK**.
4. Right click on the incomplete connection line, and select **Properties**. Deselect the **Pass event data()** check box if visible.
5. Select the **Set parameters...** button.
6. Click on the field next to the **text** value and type **Getting data from AS/400**, **please wait...**
7. Click **OK**. Click **OK** again.

Now when the **List** button is pushed, a message is sent to the status area saying **Getting data from AS/400, please wait...**

Removing All Records from the Inventory List

Before you populate the subfile with data records, you have to remove any records in the list.

1. Connect **actionPerformed** event of the **List** button to the Inventory List (PRODDTL1) bean.
2. Select **Connectable Features...** on the pop-up menu, and select the **removeAllSubfileRecords()** method on the methods list. Click **OK**.

Now when the **List** button is pushed, the records in the subfile InventoryList will be removed.

Populating the Inventory List with Records

Then you populate the subfile with data records from the **PRODDTL** file.

1. Connect **actionPerformed** event of the **List** button to the Inventory List (PRODDTL1) bean.
2. Select **Connectable Features...** on the pop-up menu, and select the **readAllRecords()** method from the method list. Click **OK**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Displaying the Message "Data retrieved"

1. Connect actionPerformed event of the List button to the Status label. Select Connectable Features... on the pop-up menu, and on the End connection to window, select the setText(String) method. Click OK.
2. Right click on the incomplete connection line. Select Properties. Deselect the Pass event data() check box if visible. Then click on the Set parameters... button. Type **Data retrieved** for the text value. Click OK. Click OK again.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Testing What You Have Done So Far

1. From the **Bean** menu select **Save bean...**.
2. From the **Bean** menu select **Run** and then **Check Class Path**. Click **Class Path** tab. Check **Project path:** check box. Click **Compute now** button. Ensure that the IBM Enterprise Kit for AS400 project, JFC class libraries project and the project where the class resides, **VA Java for AS400 Labs** are added to the **Project path**. Click **OK**.
3. Click on the **(Run)** button on the tool bar. VisualAge for Java saves the bean, compiles the class and runs the compiled bean.
4. A window appears displaying the GUI you have just updated. The Signon to AS400 window appears. Type your user ID and password. Click on the **List** button. You should see "Getting data from AS/400, please wait ...". Once the data has been retrieved from the AS/400, the list box is populated with data. You should see "Data retrieved" on the status bar.
5. Close the window to end the test.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Copying Methods from Another Project

You need several new methods to populate or clear the Parts Detail INVDETL bean, to add or update records from the Inventory List PRODDTL bean, and to enable all the buttons. Usually, you would have to create these methods yourself, however, you can copy the methods from the IBM Enterprise Toolkit for AS400 project.

1. Click on the Workbench pane.
2. Expand the **VA Java for AS400 Labs** project in the project window.
3. Expand the **ZY400APP** package.
4. Expand the **InventoryList** class.
5. Select **addRecord()**, **clearForm()**, **enableAllButtons()**, **showDetails(String[])**, and **updateRecord(int)** methods by holding down the CTRL key and left clicking on each method. Once you have selected all the methods, right click to pull up the pop-up menu and select the **Reorganize** option then **Copy...**.
6. The Copying Methods window appears. Click **Browse...**. Type **InventoryList** in the pattern. Select **InventoryList** from the **Type Names** list and **VJ400LAB2** from the **Package Names** list. Click **OK**. The VJ400LAB2.InventoryList name has been added to the Copy Methods window. Deselect the **Rename(copy as)** checkbox. Click **OK**. Click **OK** again.
7. Note that there are red Xs in front of the methods you have just copied. These errors are due to undefined variables and will resolve as soon as you add the appropriate connections. When you add buttons from the Part Details ensure that you use the correct bean name for each button so that these methods work correctly.

Populate the Part Details when the Part is Selected in the Inventory List

Next you can connect the Inventory list PRODDTL bean to Part Details INVDETL bean so that the detail informtion of the selected part will be shown on the Parts Details screen.

1. Click on the InventoryList icon on the desktop taskbar. Connect **ItemStateChanged** event of the Inventory List PRODDTL bean to the free form surface. Select **Event to Code...** on the pop-up menu, and on the Event-to-Code Connection from PRODDTL1 window click on the Method drop down and select the **showDetails(String[])** method. Deselect the **Pass event data()** check box if visible. Click **OK**.
2. Right click on the previous connection and select **Connect** then **rowData**. Connect to the Inventory List PRODDTL bean. Select **Connectable Features...** and on the End connection to window select the Method radio button and then select the **getRecordAsStrings(int)** method. Click **OK**.
3. Right click on the incomplete connection and select **Connect** then **arg1**. Connect to the Inventory List PRODDTL bean. Select **Connectable Features...** and on the End

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

connection to window select the Method radio button and then select the **getFocusRecordNumber()** method. Click **OK**.

The three connections you have just made allow the parts item selected on InventoryList screen to display its detail information using the showDetails method. The record number of the selected record is passed as an arguement for the method. Then the subfile class gets the record details (using the getRecordAsStrings method) and writes to the Parts Details screen (using the showDetails methods) .

Adding Buttons for the Part Details

1. Click on the **JButton** bean and drop it within the frame at the bottom.
2. Repeat this action to add another four push buttons. All the buttons may not fit - just drop them overlapping if necessary and then resize them all. Now you have added all the remaining buttons for the application. Next you add names for each of the buttons.
3. Right click on the first button and select **Properties** and the Properties window for the button appears.
4. Click on the field next to the **text** value and type Add.
5. Click on the field next to the **beanName** value and type JButtonAdd.

   Note: You must use the name JButtonAdd because the method enableAllButtons( ) uses the JButtonAdd bean.

6. Select the **False** value for the **enabled** field.
7. Leave the Properties window up. Move it so that you can see all the buttons.
8. Repeat this process for each button, naming them Update, Delete, Clear and Print. Remember to change the beanName accordingly, JButtonUpdate, JButtonDelete, JButtonClear and JButtonPrint.
9. Close the Properties window.
10. To save your work, from the **Bean** menu select **Save Bean**. All the errors in the class should be fixed. (You may verify it on the workbench).

Enabling All the Buttons

1. Connect **actionPerformed** event of the **List** button to the free form surface. Select **Event to Code...** on the pop-up menu, and on the Event Code window select the **enableAllButtons()** method. Click **OK**.

When you click on the **List** button all buttons are enabled.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Testing What You Have Done So Far

1. Click on the **(Run)** button on the tool bar. VisualAge for Java saves the bean, compiles the class and runs the compiled bean.
2. A window appears displaying the GUI you have just updated. The Signon to AS400 window appears. Type your user ID and password. Click on the **List** button. You should see "Getting data from AS/400, please wait ...". Once the data has been retrieved from the AS/400, the list box is populated with data. You should see "Data retrieved" on the status bar.
3. Close the window to end the test.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 5: Add, Update, Delete Records (OPTIONAL -- ADVANCED SECTION)

Introduction

In this exercise you make connections between each of the buttons under the Parts Details screen and the methods which performs the different Add, Update, Delete and Clear functions. You enable the Print button in the next exercise.

Goal

At the end of this exercise you should be able to:

- Use a method to add a record
- Display messages on the status line
- Use a method to update a record
- Use a method to delete a record
- Use a method to clear a record

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Enabling the Add Button

The **Add** button first gets the data from the Part Details INVDETL, adds the record to the database, and then refreshes the Inventory List PRODDTL bean.

1. Connect **actionPerformed** event of the **Add** button to the free-form surface.
2. Select **Event to Code...** on the pop-up menu, and on the Event to code window deselect the **Pass event data()** check box if visible. Select the **addRecord()** method. Click **OK**.

Displaying Message "Record added"

You can send a message " Recod Added" to the status area when the Add button is pushed and a record is added to the database.

1. Right click on the previous connection. Select **Connect** then **normalResult**. Position the mouse over the Status label bean and left click to bring up the pop-up menu. Select **text**.
2. Right click on the incomplete connection line, and select **Properties**. Deselect the **Pass event data()** check box if visible. Click on the **Set parameters...** button. Type **Record added** for the **text** value. Click **OK**. Click **OK** again.

Enabling the Update Button

Now you connect the **Update** button to a method to retrieve the data from the Part Details, update the database and update the Inventory List PRODDTL bean.

1. Connect **actionPerformed** event of the **Update** button to the free-form surface. Select **Event to Code...** on the pop-up menu, and on the Event to code window deselect the **Pass event data()** check box if visible. Select the **updateRecord(int)** method. Click **OK**.
2. Right click on the incomplete connection line, and select **Connect** then **recNum**. Make a connection to the Inventory List PRODDTL bean. Select **Connectable Features...** on the pop-up menu, and on the End connection to window, select the **getFocusRecordNumber()** method. Click **OK**.

The connections you have just completed allow you to take the new parts details information on the Parts Details screen and update the database on the AS/400 using the updateRecord method. Then it refreshes the InventoryList screen to display the updated record.

Enabling the Delete Button

Next you add a connection between the **Delete** button and the deleteRecord method to delete the record from the database. It requires the record number of the record at focus to be passed as a paramenter.

1. Connect **actionPerformed** event of the **Delete** button to the Inventory List bean. Select **Connectable Features...** and on the End connection to window select the **deleteRecord(int)** method. Click **OK**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

2. Right click on the incomplete connection line, and select **Connect** then **recordNumber** and make a connection to the Inventory List bean. Select **Connectable Features...** then the **getFocusRecordNumber()** method.

After you delete the record, you want to refresh the inventory list by reading the record again.

1. Connect **actionPerformed** event of the **Delete** button to the Inventory List bean. Select **Connectable Features...** and then select the **readAllRecords()** method from method list. Click **OK**.

Enabling the Clear Button

Connect **actionPerformed** event of the **Clear** button to the free-form surface. Select **Event to Code...** on the pop-up menu, and on the End connection to window deselect the **Pass event data()** check box if visible. Select the **clearForm()** method. Click **OK**.

Now when the **Clear** button is pushed the fields in the Part Details screen are cleared.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Testing What You Have Done So Far

You test all the connections and buttons except the **Print** button for the Parts Details.

Select **Run**. A window appears displaying the GUI you have just updated. Test all the buttons currently enabled except **Print**. Close the window to end the test.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Exercise 6: Print Record Information (OPTIONAL -- ADVANCED SECTION)

Introduction

In this exercise you are going to add connections between the Print button and the PRTINV1 bean generated using the Program Call Smart Guide in exercise 1. When the Print button is pushed, a program call is made to a RPG program on AS/400 which prints the detail part information of the selected record.

Goal

At the end of this exercise you should be able to:

- Display messages on the status line
- Call the print function
- Send the part number to the print function
- Run the print inventory program

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Making an AS/400 Connection to call the Print Function

1. Connect **actionPerformed** event of the **Print** button to the PRTINV1 bean. Select **Connectable Features...** on the pop-up menu, and on the End connection to window select the **connect(com.ibm.as400.access.AS400)** method. Click **OK**.
2. Right click on the incomplete connection line, and select **Connect** then **obj**. Connect to the Inventory List PRODDTL bean. Select **Connectable Features...** and on the End connection to window select the **getAS400Server()** method. Click **OK**.

You have just added the connection between the **Print** button and PRTINV1 bean to make an AS/400 connection.

Sending the Part Number to the Print Function

1. Right click on the connection between the Inventory List PRODDTL bean and the showDetails( ) method, and select **Connect** then **normalResult**. Connect to the Part Details INVDETL bean. Select **Connectable Features...** and select the **entryFieldGetStringValue (com.ibm.ivj.as400.util.AS400SVisualTextField)** method. Click **OK**.
2. Right click on the incomplete connection line, and select **Connect** then **entryField**. Connect to the Part Details INVDETL bean. Select **Connectable Features...** and on the End connection to window select the **getEntryField1PARTNO()** method. Click **OK**.

Select the previous connection and select **Connect** then **normalResult**. Connect to the PRTINV1 bean. Select **Connectable Features...** and on the End connection to window select the **partnoAsString()** method. Click **OK**.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

The connections you just made between Inventory list PRODDTL and the Part Details INVDETL bean allow the data for the part to be assigned to each field in the Part Details screen, when a part is selected in the Inventory list. The record number of the selected record is obtained by getEntryField1PARTNO() method and passes on to the program call bean..

Running the Print Inventory Program on AS/400

Finally, you invoke the runProgram method in the PRTINV1 bean to run the ILE RPG print program on AS/400.

1. Connect **actionPerformed** event of the **Print** button to the PRTINV1 bean. Select **Connectable Features...** on the pop-up menu, and on the End connection to window select the **runProgram()** method. Click **OK**.

Printing the Program Completion Message

Now you want to send a message to the status label after the RPG print program is completed.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

1. Connect **actionPerformed** event of the **Print** button to the PRTINV1 bean. Select **Connectable Features...** on the pop-up menu, and on the End connection to window select the **getCompletionMsg()** method. Click **OK**.
2. Right click on the previous connection. Select **Connect** then **normalResult**. Position the mouse over the Status label bean and left click to bring up the pop-up menu. Select **text**.

The finished application looks like this:

Next you test the finished application. Check that the **Print** button functions as designed.

Testing Your Completed Application

Select **Run**. A window appears displaying the GUI you have just updated. Click on **List** to retrieve the data from the AS/400. Select a part number from the Inventory List and click on the **Print** button. The Signon to AS/400 window may appear if the server name for the PRTINV1 bean and the INVDETL1 bean are identical. If required, type your user ID and password. The status label shows "Program called ok".

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Go to an AS/400 session and check the spool files QSYSPRT and INVPRINT. QSYSPRT should indicate the program was called successfully. INVPRINT should indicate the part number matches the part number selected in the Inventory List.

Close the window to end the test.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

Appendix 1. Adding a Feature into the Workbench

This section will show you how you can install a feature after you install **VA Java** for the first time.

- Select **File -> Quick Start** from the menu bar on the **Workbench**. The **Quick Start** dialog is displayed.

- Click on **Features** from the left hand list and then on **Add Feature** from the right hand list. Now click **OK**. The **Selection Required** dialog is displayed.

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.

- Scroll down the list on the **Selection Required** dialog. You may not see the entry **IBM Enterprise Toolkit for AS/400 3.0,** since it is already added to the **Workbench**.
- Click **Cancel** to close the dialog. **<u>DON'T CLICK OK</u>** (it will overwrite any fixes that might have been applied to the feature and it is time consuming).

 © Copyright IBM Corp. 1999

Course material may not be reproduced in whole or in part without the prior written permission of IBM.