General Introduction to the Common Java Labs

Overview

There are six Java labs being offered at this COMMON conference:

1. Introduction to Java
2. Introduction to VisualAge for Java
3. AS/400 Toolbox for Java - Data Queues and Program Calls
4. Java Programming for RPG Programmers
5. AS/400 Data Access Using VA Java Enterprise Toolkit for AS/400

These labs are arranged in order, and although taking them in sequence is not required, it is strongly recommended!

To promote some continuity throughout the Java labs, the same application will be used. To allow the entire application to be referenced in every lab, some labs will provide pre-packaged or pre-generated Java Beans. This will allow the focus to be directed to the correct area for each lab exercise.

Lab 1: Introduction to Java

This lab will provide an overview of Java. During this lab you will learn how to create a Java application, a Home page to place a Java applet, and a Java applet. You will also see the finished Java application that you will be building during the next three labs.

Lab 2: Introduction to VisualAge for Java

This lab will provide an overview of the Integrated Development Environment provided by VisualAge for Java. Included is an overview of the Visual Composition Editor, the basic concepts of VisualAge for Java projects, packages and classes. 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, programs and data queues.

Lab 3: AS/400 Toolbox for Java

This lab will provide an overview of the AS/400 Toolbox classes. You will use the application you previously built in lab three except there will be one difference, the beans that were already built for you in lab three are missing. In this lab you use the AS/400 Toolbox classes to create these beans.

Lab 4: Java for RPG Programmers

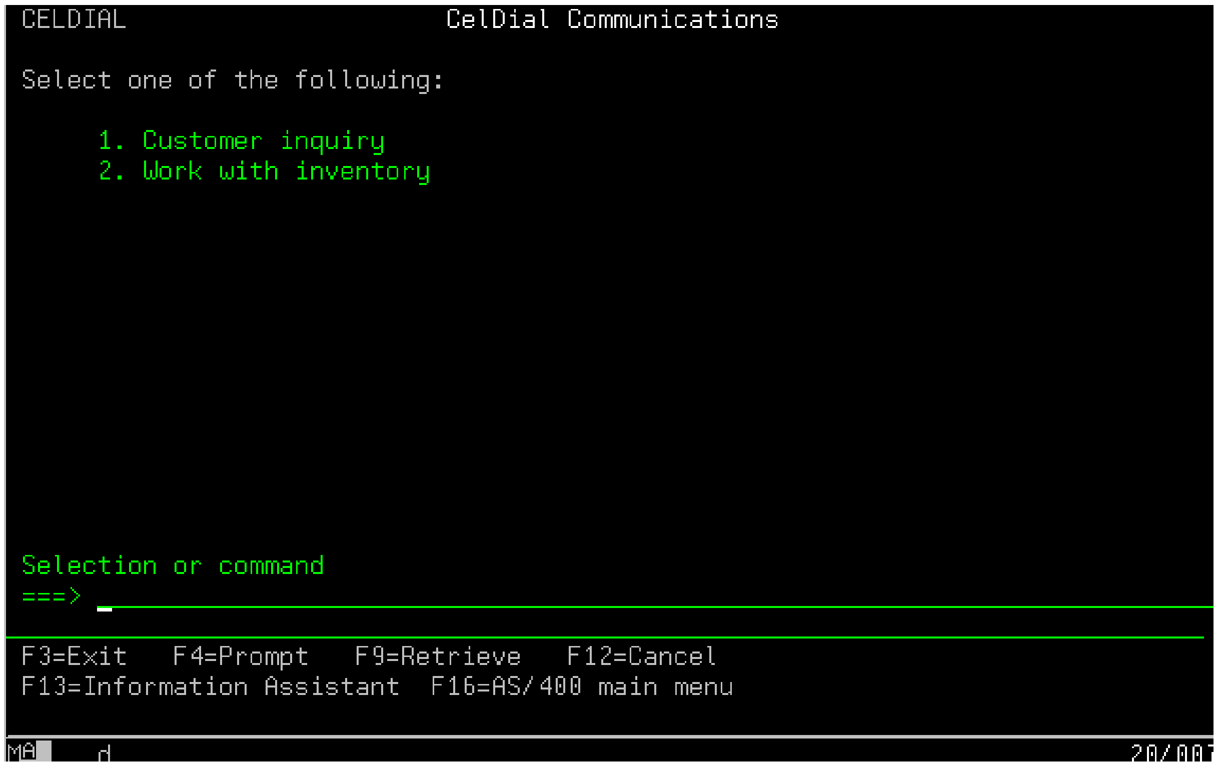
This lab will provide an overview of the Java language. It will also contrast the Java language with the RPG language. In this lab you will create a Java application that runs on the workstation but accesses data on the AS/400.

Lab 5: AS/400 Data Access Using VA Java Enterprise Toolkit/400

This is a new lab added in this COMMON. It will provide an overview of the VisualAge for Java Enterprise Toolkit for AS/400 feature available in the VA Java Enterprise Edition V2.0. In this lab you will build application that uses record-level data access to access data on the AS/400. You will use the Create Subfile SmartGuide to create the subfile classes.

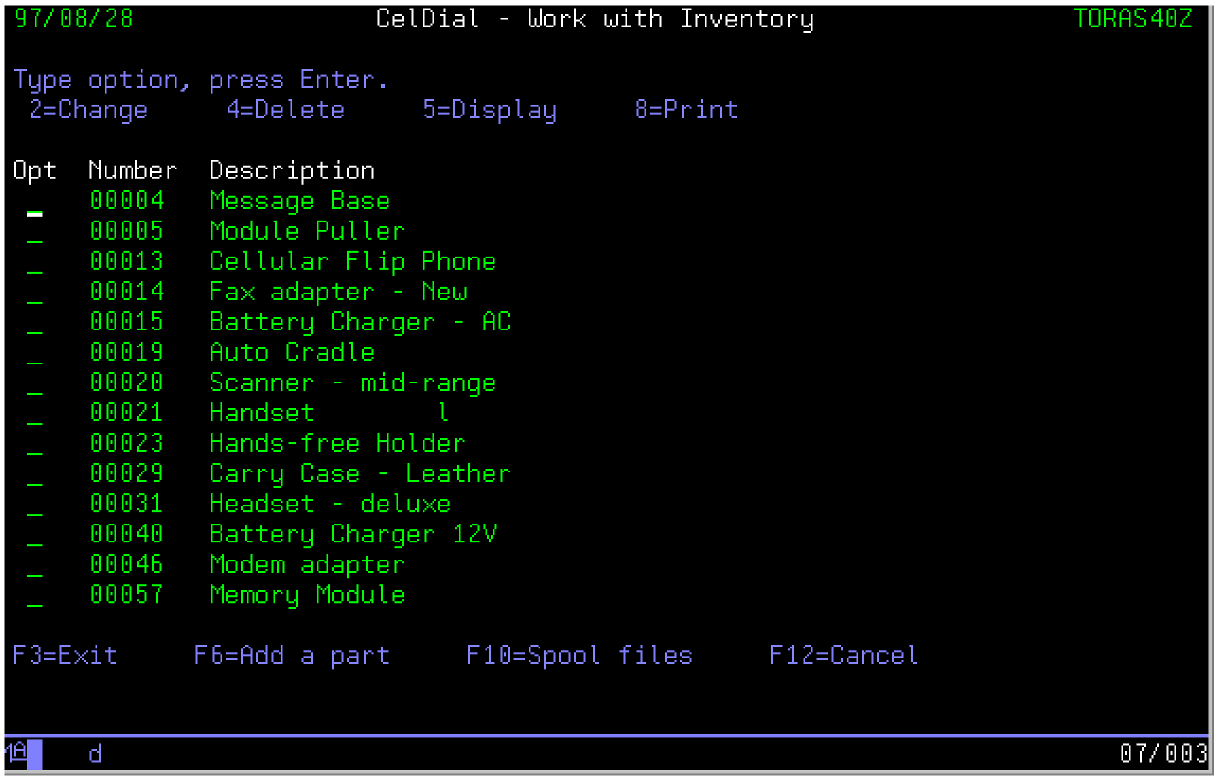
The Parts Inventory Application

This application was originally written as a green screen, 5250 based, RPG/400 application. The application accesses the AS/400 database - the parts file. You can view detailed information about specific parts, update that information or delete the part from the inventory. The company is CelDial Communications and the parts in inventory are cellular telephones and other accessories. The first screen of the application is a menu screen allowing selection of whether to do a customer inquiry or to work with inventory.



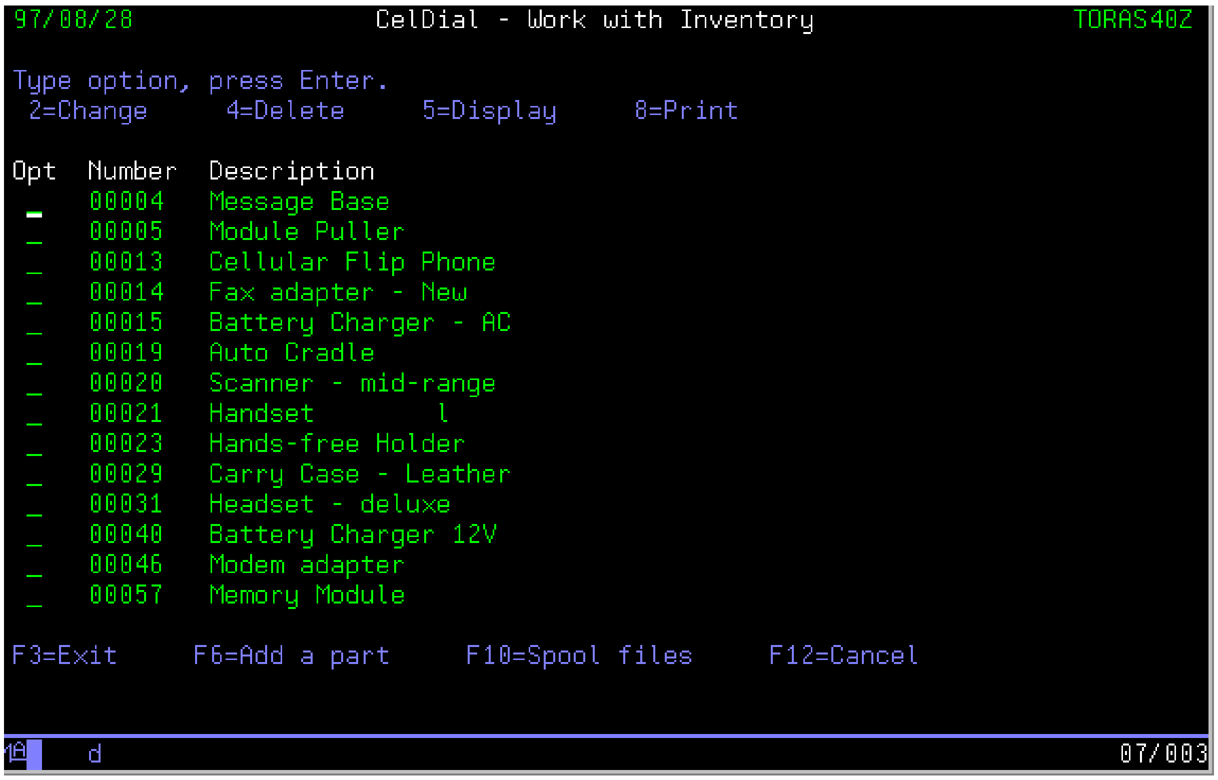
```
CELDIAL                      CelDial Communications

Select one of the following:

     1. Customer inquiry
     2. Work with inventory











Selection or command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=Information Assistant  F16=AS/400 main menu

MA      d                                                          20/007
```

The Work with Inventory screen uses a subfile to list the various parts available.



```
97/08/28                    CelDial - Work with Inventory                  TORAS40Z

Type option, press Enter.
 2=Change      4=Delete      5=Display     8=Print

Opt  Number  Description
 _   00004   Message Base
 _   00005   Module Puller
 _   00013   Cellular Flip Phone
 _   00014   Fax adapter - New
 _   00015   Battery Charger - AC
 _   00019   Auto Cradle
 _   00020   Scanner - mid-range
 _   00021   Handset        l
 _   00023   Hands-free Holder
 _   00029   Carry Case - Leather
 _   00031   Headset - deluxe
 _   00040   Battery Charger 12V
 _   00046   Modem adapter
 _   00057   Memory Module

F3=Exit     F6=Add a part     F10=Spool files     F12=Cancel


MA     d                                                                 07/003
```

From there, a user will select the part to be viewed or updated, and a subsequent screen of information will be displayed. This additional screen is used whether view, update delete or add has been selected.


```
97/08/28                     CelDial - Work with Inventory                     TORAS40Z

Type option, press Enter.
 2=Change     4=Delete     5=Display     8=Print

Opt  Number  Description
 _   00004   Message Base
 _   00005   Module Puller
 _   00013   Cellular Flip Phone
 _   00014   Fax adapter - New
 _   00015   Battery Charger - AC
 _   00019   Auto Cradle
 _   00020   Scanner - mid-range
 _   00021   Handset         l
 _   00023   Hands-free Holder
 _   00029   Carry Case - Leather
 _   00031   Headset - deluxe
 _   00040   Battery Charger 12V
 _   00046   Modem adapter
 _   00057   Memory Module

F3=Exit     F6=Add a part     F10=Spool files     F12=Cancel


     d                                                                          07/003
```

The Java application which will be used in the Java lab exercises will be based on this application. The Java application, however, will be created and will execute in a client environment, accessing the AS/400 data, programs and data queues.

Your Lab Environment and Requirements for the VisualAge Java development environment:

AS/400 - RISC model OS/400 V3R6+ or IMPI V3R2+
PC - Intel based; 48 Mg Memory (min); 200 Mg hard drive
OS/2, Windows 95 or Windows NT
VisualAge for Java - Enterprise Edition
AS/400 Toolbox for Java
Client Access/400

VisualAge for Java - What you need to know to get started

IBM's VisualAge for Java is a complete integrated development environment for the creation of both Java applications and Java applets. It is designed not only to create Java applets, dowloadable from an Internet server, but also to create full function business applications. Developers can extend server-based applications to communicate with Java clients on the Internet or intranet. VisualAge for Java creates 100% pure Java compatible applications, applets and JavaBeans.

VisualAge for Java is a member of the VisualAge family of products and uses the same Visual Composition Editor as in the other VisualAge products such as SmallTalk, C++, and Generator. This Visual Composition Editor can be considered the 'heart' of VisualAge for Java.

Unlike some other development environments, VisualAge for Java enables a developer to build and run applications, applets and code snippets interactively without the need to compile the code. Along with the execution of the applications and applets, there is a full debug capability.

In fact, VisualAge Java comes with the following core components:

1. Integrated Development Environment
    a. Hierarchy browser
        i. projects
        ii. packages
        iii. classes
        iv. methods
    b. Editor
    c. Debugger
    d. Applet viewer
    e. Team support
    f. Java Class Libraries
    g. Visual Composition Editor
2. Enterprise Access Builder (EAB)
    a. Data Access Builder (DAX)
    b. Proxy Builder, J2C++ Builder, CICS Builder

Java Support

Java is a collection of classes built from the ground up, following Object Oriented principles. So, the lowest component (part in some other languages such as SmallTalk) is a class. A class may be joined with other classes and stored as a composite class. These classes are grouped together and placed into packages. The packages can be further grouped into projects. On the VisualAge Java workbench, the structure of your application, the components that you import and the components that you create are displayed according to this hierarchy.

As an application is created in VisualAge Java, each object is given some features - properties, events and methods. We will not cover these in detail here, other than to say that when creating the application, it is necessary to connect the features of one part to the features of another part. This is how the application will ultimately execute. The connections (or wires as some developers call them), that are drawn visually by the developer are used to generate some underlying Java code. This code can be viewed in the VisualAge Java script editor.

VisualAge for Java Workbench

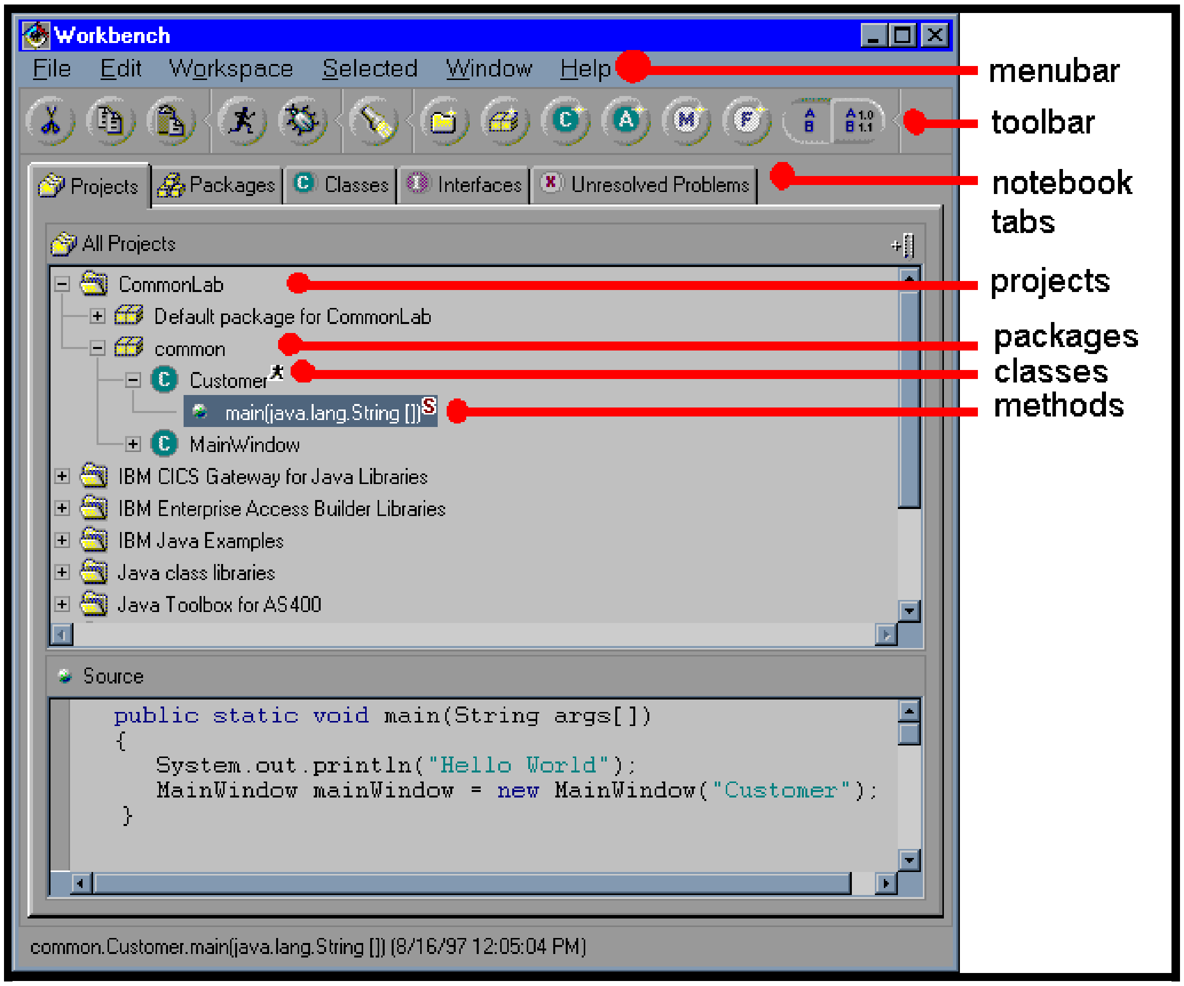
The VisualAge for Java **Workbench** is a multiple paned **IDE** (*Integrated Development Environment*) for Java development. It divides your Java applications into:

- **Projects**. These, like AS/400 libraries, allow you to partition your applications into manageable units. These are VAJava-unique constructs. Projects contain multiple packages.

- **Packages**. These, like AS/400 ILE RPG service programs, allow you to divide your application pieces into easily reused units. These are Java language constructs. Packages contain multiple classes.

- **Classes**. These, like AS/400 ILE RPG modules, allow you to divide your source code into functions (methods in Java, procedures and subroutines in RPG) and variables those functions need. These are typically self-containing groupings. Classes contain multiple fields (variables) and methods.

- **Methods**. These, like AS/400 ILE RPG procedures and subroutines, contain all the actual code your program or application will use. Unlike RPG, in Java executable code can only exist in methods. And methods can only exist inside classes.


Workbench
File Edit Workspace Selected Window Help
menubar
toolbar
Projects Packages Classes Interfaces Unresolved Problems
notebook tabs
All Projects
CommonLab
Default package for CommonLab
common
Customer
main(java.lang.String [])
MainWindow
IBM CICS Gateway for Java Libraries
IBM Enterprise Access Builder Libraries
IBM Java Examples
Java class libraries
Java Toolbox for AS400
projects
packages
classes
methods
Source

```
public static void main(String args[])
{
    System.out.println("Hello World");
    MainWindow mainWindow = new MainWindow("Customer");
}
```

common.Customer.main(java.lang.String []) (8/16/97 12:05:04 PM)

The Visual Composition Editor

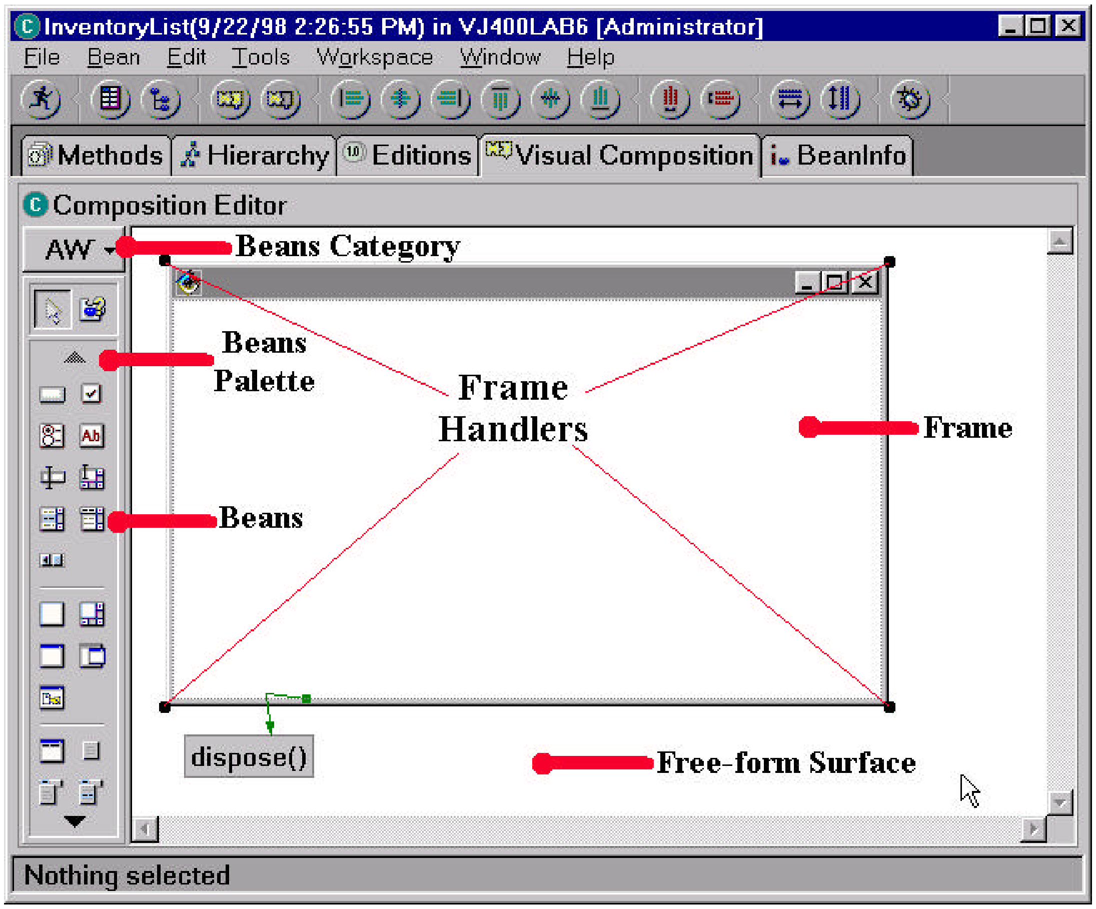
There are various components that make up the Visual Composition Editor:

Free-form surface, the white space surrounding the window being built. The free-form surface is usually used to drop non-visual parts which you will want to use in your class, but don't want to show the end-user.

Frame, the window within the free-form surface, on which you place all the visual elements of your GUI.

Beans palette, the area on the left containing: Bean categories, a container for beans; Beans, the parts to build your GUI; and the Sticky check box that will allow you to perform multiple drops without returning to click on the bean.

Handlers, shown around the frame, that you can use to manipulate the size of beans such as a frame and labels.


InventoryList(9/22/98 2:26:55 PM) in VJ400LAB6 [Administrator]
File Bean Edit Tools Workspace Window Help
Methods
Hierarchy
Editions
Visual Composition
BeanInfo
Composition Editor
AW
Beans Category
Beans Palette
Frame Handlers
Frame
Beans
dispose()
Free-form Surface
Nothing selected