Session: 46CE

Building Java GUIs Using XML

IBM eserver iSeries

Dennis Schmidt
User Technologies
IBM Rochester
dschmidt@us.ibm.com

IBM eserver. For the next generation of e-business.

Agenda

IBM eserver iSeries

- **Overview**
  - XML & PDML overview
  - PCML
  - How it all fits together
  - Operations Navigator plug-in support
  - Where to get more information
- **What is the Java GUI Builder**
  - Advantages
  - Java development tools
  - GUI Builder requirements
  - Customizing the GUI Builder
- **Creating & editing panels**
  - Window types
  - Control / field types
  - Field properties
  - Recommended Naming conventions
  - Selected & Deselected
  - Tab order
  - Cut, copy, paste
  - Alignment and spacing
- **Window Types**
  - Property sheets
  - Wizard
  - Menus & Menu bars
  - Toolbars
  - Deck panes
  - Split panes
  - Tabbed panes
- **Generating Files**
  - Generated files
  - Translatable text
  - Databean generation
  - Event handler generation
    - Buttons
    - Activation
    - Double-click
    - Slider
    - Context menu
  - Help generation
- **Exercises**

IBM eserver. For the next generation of e-business.

XML and PDML Overview

IBM @server iSeries

- **What is XML?**
  - XML (Extendible Markup Language) allows you to define your own tagged-based language.
- **What is PDML?**
  - PDML (Panel Definition Markup Language) is a set of tags that define a panel, what is on the panel, how it is laid out, etc.
  - PDML is a language that we developed using XML. It's similar in structure to HTML and SGML.
  - Here are some examples of PDML tags:
    - <panel> -- defines a panel
    - <title> -- specifies the title of the panel or field
    - <size> -- specifies the size of the panel or field
    - <label> -- defines a label on the panel (static text field)
    - <location> -- specifies the location of the field on the panel
    - <button> -- defines a button on the panel
    - <textfield> -- defines a textfield on the panel
    - and many more...

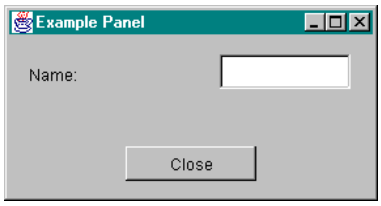
IBM @server. For the next generation of e-business.

PDML Example

IBM @server iSeries

- **Here's what the PDML looks like for the simple panel shown below:**

```
<PDML version="1.0" source="JAVA" basescreensize="1024x768">

    <PANEL name="EXAMPLE_PANEL">
        <TITLE>EXAMPLE_PANEL</TITLE>
        <SIZE>278,120</SIZE>
        <LABEL name="NAME_LABEL" disabled="no">
            <TITLE>NAME_LABEL</TITLE>
            <LOCATION>15,20</LOCATION>
            <SIZE>100,19</SIZE>
        </LABEL>
        <TEXTFIELD name="NAME" masked="no" editable="yes" disabled="no">
            <TITLE>NAME</TITLE>
            <LOCATION>161,14</LOCATION>
            <SIZE>100,26</SIZE>
        </TEXTFIELD>
        <BUTTON name="CLOSE_BTN" disabled="no">
            <TITLE>CLOSE_BTN</TITLE>
            <LOCATION>89,83</LOCATION>
            <SIZE>100,26</SIZE>
        </BUTTON>
    </PANEL>

</PDML>
```



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

Why PDML

IBM @server iSeries

- **PDML was created because:**
  - Create a representation of GUIs that is platform and UI technology independent
  - Java LayoutManager classes provide little control over actual sizes and positions of user interface components. Initial results are often disappointing and require repeated rework in order to achieve the desired results on a specific platform.
  - Each change to the panel layout requires recompilation of Java code
  - Java LayoutManager classes attempt to compensate for varying length text strings in different NLVs of an application by adjusting the sizes of the user interface components. In practice, this results in visually unappealing GUIs.
- **PDML provides the following benefits:**
  - GUIs are represented in a descriptive tag language
  - Developers have precise control over the sizes and positions of user interface components
  - Changes to the panel layout do not require any code to be recompiled
  - Unique versions of a given panel can be easily created for each national language language supported by an application developer.
  - National language translators do not need specialized skills to modify panel definitions

Notes: Java GUI Technology Highlights

IBM @server iSeries

- **Graphical User Interfaces**
  - Platform and technology independent representation of graphical user interfaces based on the Extensible Markup Language (XML) - Panel Definition Markup Language (PDML)
  - Pure-Java framework for interpreting the PDML language and constructing graphical user interfaces based on the Java Foundation Classes (JFC)
  - Set of tools for building GUIs in XML
- **AS/400 toolbox data objects**
  - XML language to assist in building AS/400 toolbox objects that represent AS/400 program call parameters - Program Call Markup Language (PCML)
  - Pure-Java framework for interpreting the PCML language and retrieving/storing information using the AS/400 Toolbox for Java
- **Operations Navigator Plug-ins**
  - Contains the capability of locating, loading, and executing GUI components and plug-ins developed in Java

Notes: Graphical User Interfaces using Java

IBM @server iSeries

- **Panel definition markup language (PDML) for building GUIs using XML and pure-Java technology for rendering these GUI definitions using the Java Foundation Classes (JFC)**
  - Based on Java Beans
  - Contains three elements:
  - Platform and technology independent representation of graphical user interfaces using XML
  - Resource file converter and WYSIWYG Java GUI panel building
  - Java GUI runtime package
- **Design points**
  - Platform and technology independent representation of panels using XML
  - Other screen representations can be converted to XML (i.e. RCs) and used by the runtime package
  - Can be run:
  - As an AS/400 Operations Navigator plug-in
  - In a browser as an applet
  - In a Java application

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

Notes: XML Panel Definition Markup Language

IBM @server iSeries

- **Panel Definition Markup Language (PDML) is a tag language defined using the eXtensible Markup Language standard (XML)**
- **Extensible Markup Language (XML) is a evolving industry standard defined by the World Wide Web Consortium (W3C)**
  - Subset of the Standardized Generalized Markup Language (SGML) that is targeted for use on the Web
- **XML defines the grammar for creating your own markup language for a specific purpose**
- **Define a set of tags similar to HTML**
- **Parsed into its constituent parts by the IBM XML parser (written in Java) for use by Java beans/applets/applications**
  - Available from IBM Alphaworks - www.alphaworks.ibm.com

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

PCML

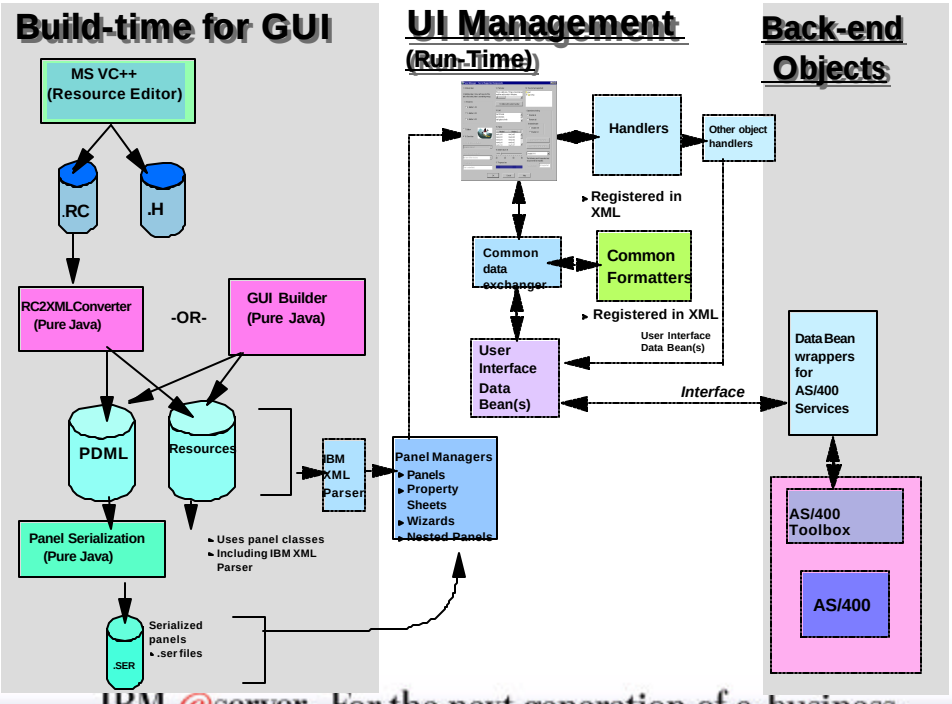
IBM @server iSeries

- **Program Call Markup Language (PCML)**
- **Used to access iSeries data**
- **Java runtime framework for supporting distributed iSeries program call to iSeries APIs using the iSeries Toolbox for Java**
  - Parsing of PCML (using the IBM XML parser integrated) or Serialized PCML
  - Java classes are supplied that use the PCML and iSeries Toolbox for Java classes
  - iSeries Distributed Program Call (DPC) server handles the remote program calls
  - For increased performance, a serialized version of the parsed PCML can be used to construct the remote program call
  - PCML program call support can run on the client or the iSeries JVM
  - ProgramCallDocument class is supplied with the following functions
    - ProgramCallDocument.setValue - sets an input parameter value
    - ProgramCallDocument.callProgram - calls a program on the AS/400
    - ProgramCallDocument.getValue - gets an output value

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

How It All Fits Together

iSeries



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

Notes: PDML GUI Build-Time Contents

IBM @server iSeries

- **3 ways to build panels in PDML:**
  - Windows tools can be used to create panels
    - Pure-Java tools supplied to convert .RC files to .PDML and Resource Bundle (Property/List)
    - PDML preserved for items directly edited (i.e. comments, supplemental tags (enable/disable), etc.)
  - Pure-Java WYSIWYG GUI builder can be used to create panels
    - Produces .PDML and Resource Bundle (Property/List)
    - PDML preserved for items directly edited (i.e. supplemental tags (enable/disable), etc.)
  - Manually type in PDML and Resource Bundle support
- **Panels can be serialized for better performance at run-time**
- **Translatable text separation for Internationalization via resource bundles (properties or list)**
- **HTML On-line help skeletons automatically generated for each panel**
- **Additional PDML tags supported for simpler GUI panel building**
  - PropertySheet
  - Wizard
  - Nested panels
  - Enable/Disable
  - Selected/Deselected
  - Data validation
  - Etc.

Notes: PDML GUI Run-Time Contents

IBM @server iSeries

- **Java runtime framework for rendering GUIs using the Java Foundation Classes (JFC)**
  - Parsing of PDML (using the IBM XML parser integrated) or Serialized PDML
  - Panel layout manager supplied that supports dialog units or pixels
  - Data exchanger provided for automatic data exchange between user interface panel and Java beans via introspection
  - GUI managers supplied for rendering:
    - Panels
    - PropertySheets (TabbedPanes)
    - Wizards
    - Multiple Panes
    - On-line help HTML information
  - Data formatter classes supplied for data validation (can be extended)
  - User interface component level event handlers supplied (can be extended)
  - Look and feel handled at runtime by GUI managers via Pluggable Look and Feel (PLAF) in JFC
    - "Windows looks like Windows"
    - "AIX looks like AIX"

Operations Navigator Java Plug-In Support

IBM @server iSeries

- **Java Runtime Environment (JRE) in Client Access**
- **Java Virtual Machine (JVM) is started by Operations Navigator**
- **AS/400 Toolbox for Java runtime support (including new PDML graphical user interface runtime and PCML data access) is included in Client Access**
  - PDML build-time tools will be included in the AS/400 Toolbox for Java
- **Operations Navigator Java enablement plug-in support includes:**
  - Plugging into the Operations Navigator function tree (code can be written in Java)
  - Plugging into the Operations Navigator lists (code can be written in Java)
  - Providing context menus for Operations Navigator (code can be written in Java)
  - Providing toolbars for Operations Navigator (code can be written in Java)
  - Graphical User Interface components can be written in Java
  - Remote AS/400 program calls can be built using the PCML
- **Note: Operations Navigator components that provide tree items or toolbars require a windows resource DLL for the icons, bitmaps and strings**

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

More Plug-in Information

IBM @server iSeries

ibm.com/eServer/iSeries/Oper_Nav

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

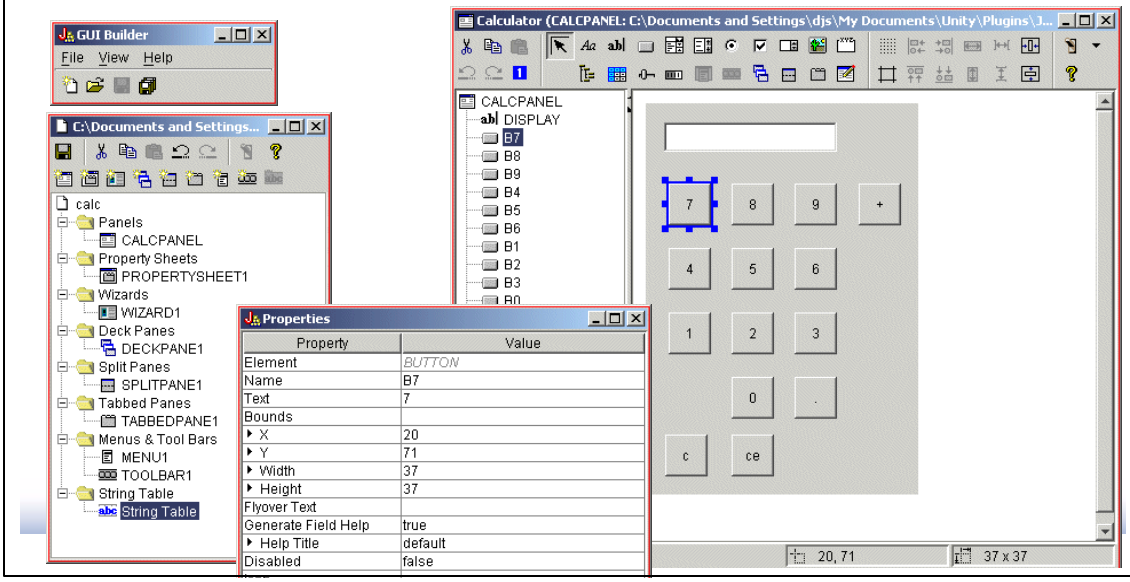
Getting to Know the GUI Builder

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

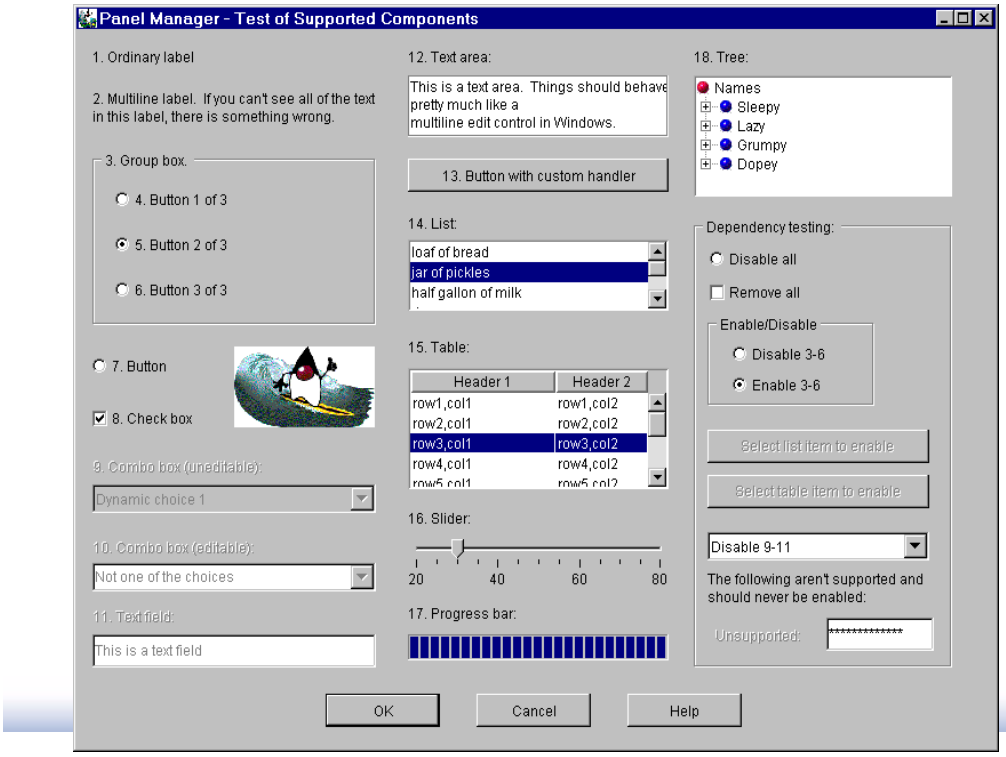
What is the GUI Builder?

IBM @server iSeries

- **The GUI Builder is...**
  - a PDML generator
  - a graphical tool used to build Java windows
  - packaged with the iSeries Toolbox for Java
  - the starting point for all Java development!!

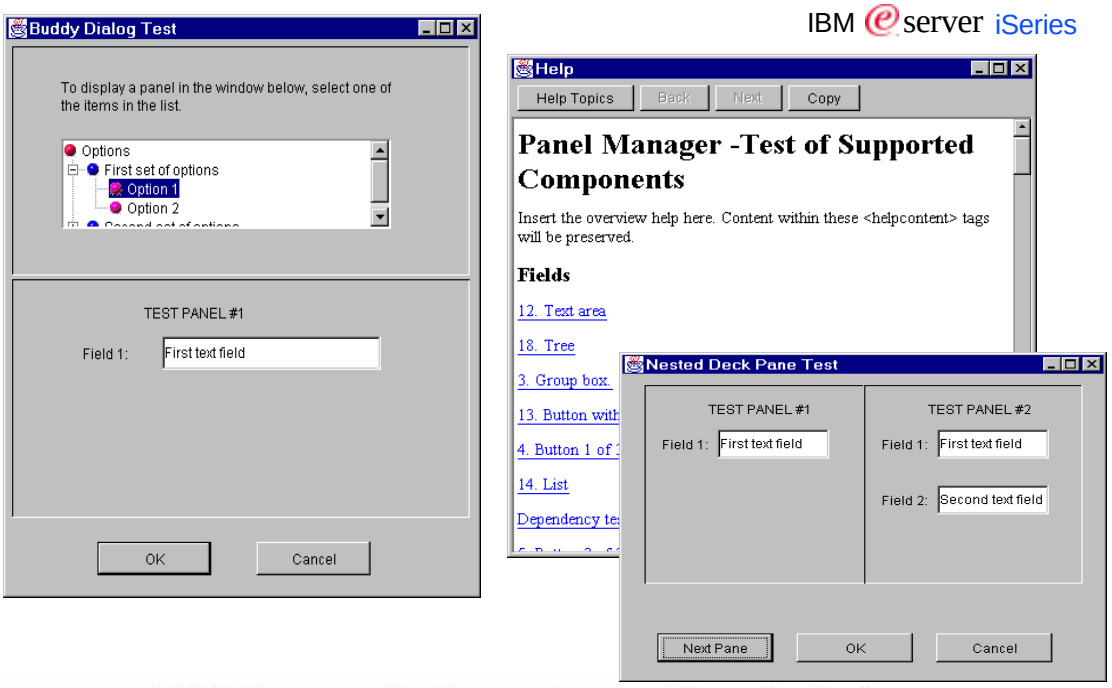


Java GUI Builder Examples



iSeries

Java GUI Builder Examples



IBM @server iSeries

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

GUI Builder and Java Development Tools

IBM @server iSeries

- **What's the difference between the Java GUI Builder and other Java GUI development tools?**
- **Java GUI Builder**
  - Builds the end-user interface -- the panels
  - Creates Java windows
  - Manages translatable text
  - Creates menus and toolbars
  - Generates help text
  - Generates data beans
  - Generates event handlers
- **Java development tools**
  - Build, compile, and manage the Java code
  - Do not use to build the end-user interface -- the panels (use the GUI Builder instead)
  - Examples:
    - VisualAge for Java
    - Symantec Visual Cafe

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

Why the Java GUI Builder

IBM @server iSeries

- **2 Other methods to create Java panels:**
  - Write raw AWT code (without a graphical tool)
    - (+) Panel components automatically resize for each NLV
    - (-) Panel layout is defined in the Java code
    - (-) Requires many edit, compile, and debug cycles before getting it right
  - Use graphical tools that generate AWT code
    - (-) Panel component locations and sizes are absolute
    - (-) Panel layout is defined in the Java code
    - (-) Translators must know Java to resize panels for each language
- **How the GUI Builder compares**
  - Use GUI Builder
    - (-) Panel component locations and sizes are absolute
    - (+) Panel layout is defined in PDML (no compilation required)
    - (+) Supports translation to other languages:
      - Structure is similar to the Windows .rc files
      - Translators don't change Java code (fewer bugs)
      - Translators don't need to know Java to resize panels
    - (+) Provides additional functionality, such as generating the help and data beans
    - (+) User interface definition is separated from the underlying technology. If the underlying technology ever changes, the UI definition doesn't need to change.

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

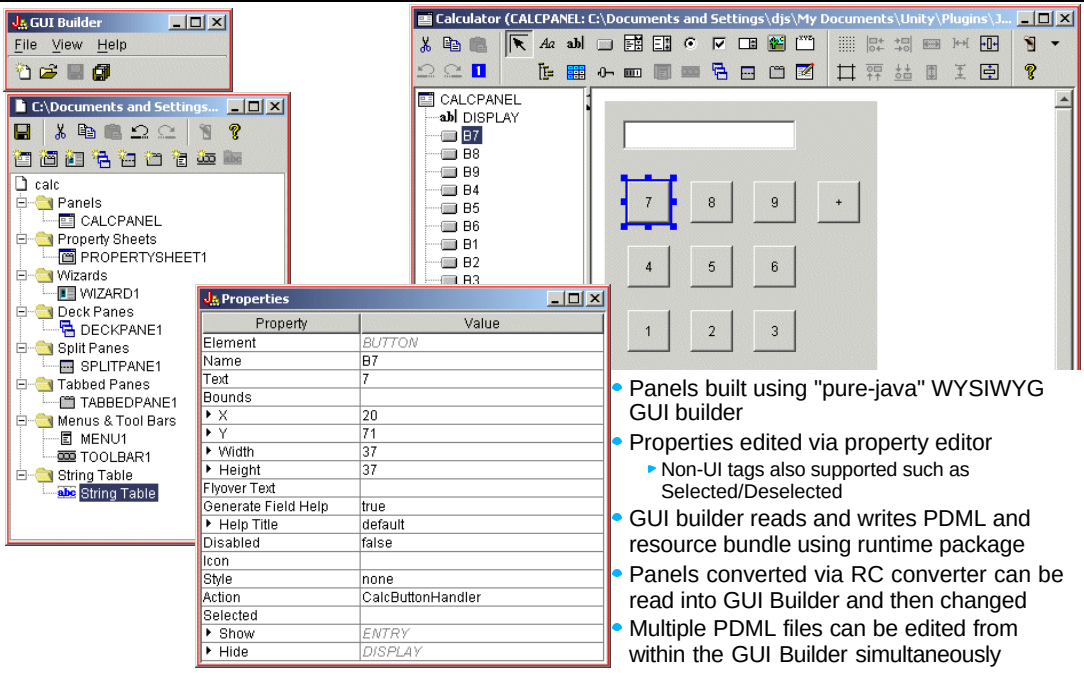
GUI Builder Requirements

IBM @server iSeries

- **Required software**
  - Java Developers Kit 1.1.8
  - Swing 1.1
  - iSeries Toolbox for Java
  - Client Access Express (with Operations Navigator installed)
- **CLASSPATH must include**
  - ClassPath = .;c;\java;c\jdk1.1.8\lib\classes.zip;C:\jdk1.1.8\lib\classes.zip;
    C:\progra~1\IBM\client~1\jt400\lib\uitools.jar;
    C:\progra~1\IBM\client~1\jre\lib\swingall.jar;
    C:\progra~1\IBM\client~1\classes\jopnav.jar;
    C:\progra~1\IBM\client~1\jt400\lib\jt400.zip;
    C:\progra~1\IBM\client~1\jt400\lib\jui400.jar;
    C:\progra~1\IBM\client~1\jt400\lib\data400.jar;
    C:\progra~1\IBM\client~1\jt400\lib\util400.jar;
    C:\progra~1\IBM\client~1\jt400\lib\x4j400.jar;
- **PATH must include**
  - Path = ...;c:\jdk1.1.8\ bin;
- **Start the GUI Builder with**
  - java com.ibm.as400.ui.tools.GUIBuilder

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

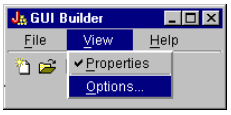
What's Where in the GUI Builder

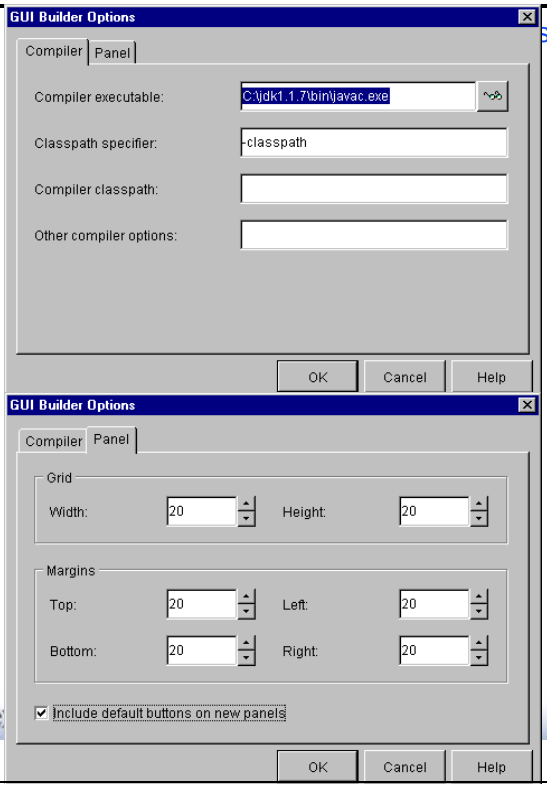


- Panels built using "pure-java" WYSIWYG GUI builder
- Properties edited via property editor
  - Non-UI tags also supported such as Selected/Deselected
- GUI builder reads and writes PDML and resource bundle using runtime package
- Panels converted via RC converter can be read into GUI Builder and then changed
- Multiple PDML files can be edited from within the GUI Builder simultaneously

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

Customizing the GUI Builder





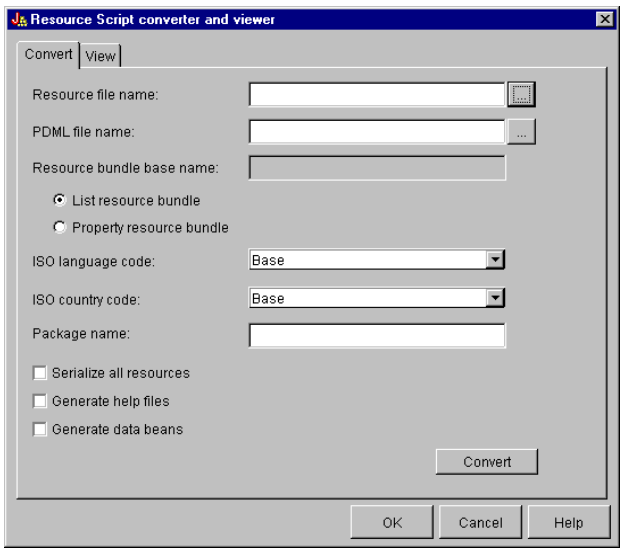
- **Compiler options**
  - Sets which compiler to use when compiling the List Resource Bundle, Data Beans, and Event Handlers
- **Panel options**
  - Sets defaults for panel creation and editing
    - Grid size
    - Margin size
    - Automatically adding OK, Cancel, Help buttons when creating a panel.

IBM @server. For the ne

Resource Converter

IBM @server iSeries

- **Resource Converter is...**
  - Tool to converts Windows resources (.rc files) to PDML
  - Can also be used to view the resources in the PDML.
  - .RC files converted to PDML using RC2XML converter/viewer "pure-java" tool (Microsoft and VisualAge for Windows)
  - Converter preserves PDML tags manually added
  - Resource Bundles can be created in any ISO language and country locale
  - Command line version also available for batch conversions



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

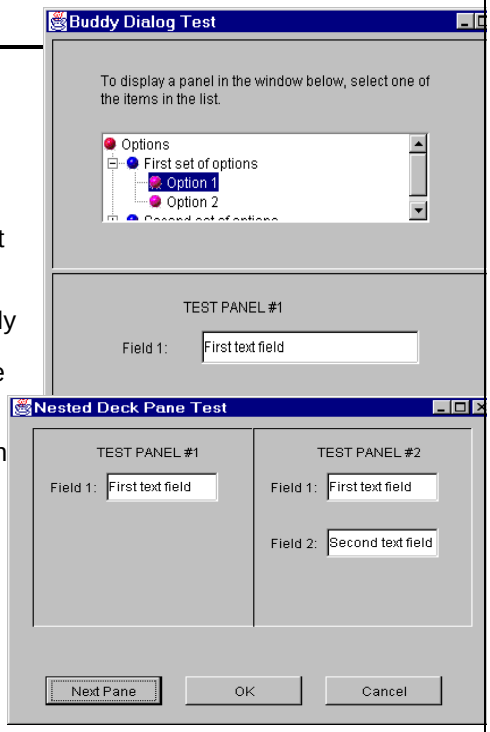
What can I do with the GUI Builder?

GUI Builder Capabilities

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

Types of Windows

- **GUI Builder creates different types of Java windows:**
  - **Panels** -- simple dialogs
  - **Property Sheets** -- Notebook pages with tabs at the top
  - **Wizards** -- multiple pages displayed sequentially
  - **Tabbed Panes** -- multiple pages with tabs at the top, bottom, right or left
  - **Split Panes** -- multiple pages displayed adjacen horizontally or vertically
  - **Deck Panes** -- multiple pages layed on top of each other

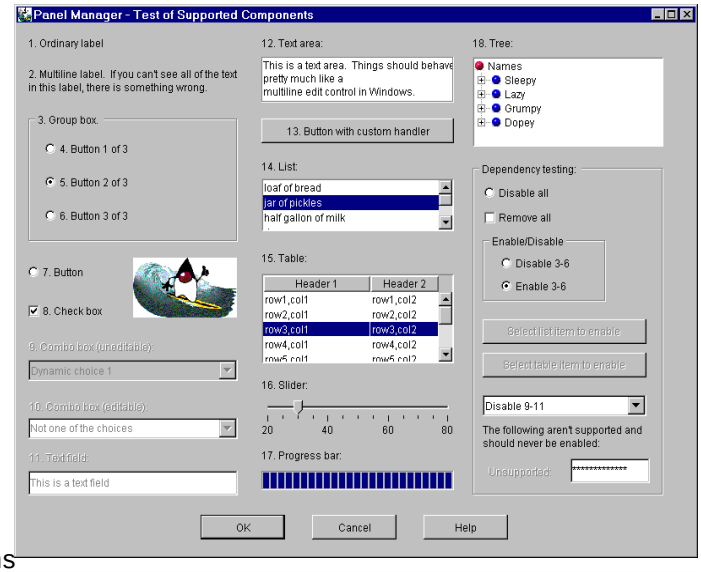


IBM @server. For the next generation of e-business.

Java Window Fields / Controls

IBM @server iSeries

- **Field / Controls that can be added to your panels:**
  - Label
  - Text box (Edit field)
  - Button
  - Combo box (Drop down list)
  - List
  - Radio button
  - Check box
  - Spinner
  - Image
  - Group box
  - Tree
  - Table
  - Slider
  - Progress bar
  - Menu bar
  - Toolbar
  - Deck Pane
  - Split Pane
  - Tabbed Pane
  - Custom
  - Buttongroup for radio buttons



Java Field Example: Text Box

IBM @server iSeries

- **GUI Builder Properties for Text Box**
  - Element -- Field Type (TEXTFIELD)
  - Name -- ID of the field (e.g. ID_NAME_EDIT)
  - Text -- Translated text that is displayed within the field by default
  - Bounds -- position and size of the fields (X, Y == position; Width, Height == Size)
  - Data Class -- Class name with the get and set methods for this field
  - Attribute -- attribute name used in the code to refer to this field
  - Flyover Text -- text that is displayed when the user hovers over this field
  - Generate Field Help -- specifies whether help should be generated for this field (or alias a link)
    - Help Title -- the label of the help index for this field (by default, it is the "Text" property)
  - Disabled -- specifies whether this field is grayed-out
  - Multiline -- specifies whether the edit field is a multiline or single-line edit box
  - Masked -- specifies whether the input/output to this field should be masked (e.g. passwords)
  - Editable -- specifies whether the user can edit the text in this field
  - Format -- the format of the input for this field (e.g. integer). This will enable format validation.

Steps to Create Java Windows

IBM @server iSeries

- **Steps to create Java Windows in the GUI Builder:**

1. **Create a "Panel"**
2. **Add the Fields to the Panels using the Toolbox**
3. **If creating a multi-page window:**
   - A. Create the additional "Panels" for each page in the multi-page window.
   - B. Create the Property Sheet, Wizard, Tabbed Pane, Split Pane, or Deck Pane.
   - C. Add the panels to the multi-page window.
4. **Click on the Preview button to verify it looks correct.**

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

Notes: Recommended Naming Conventions

IBM @server iSeries

- **Stand-alone panels:**
  - COMPONENT_PANEL
  - Example, Status Dialog in sv.guix: SV_STATUS
- **Panels within a parent window (Property Sheet, Wizard, Pane):**
  - COMPONENT_PARENT_PANEL
  - Example, General Tab on Auditing Properties: SV_AUDITING_GENERAL
- **Property Sheets, Wizards, Panes:**
  - COMPONENT_PARENT
  - Example, Auditing Properties: SV_AUDITING
- **Descriptive Labels:**
  - COMPONENT_FIELD
  - Example, "Name:" label: SV_NAME
- **Edit fields (text boxes, drop down lists, listboxes, etc.):**
  - COMPONENT_FIELD_EDIT
  - Example, textbox for "Name:": SV_NAME_EDIT
- **Radio buttons, Listbox items, Drop down list items:**
  - COMPONENT_FIELD_ITEM_ITEM
  - Example, "No maximum" radio button for Maximum jobs: SV_MAXJOB_NOMAX_ITEM

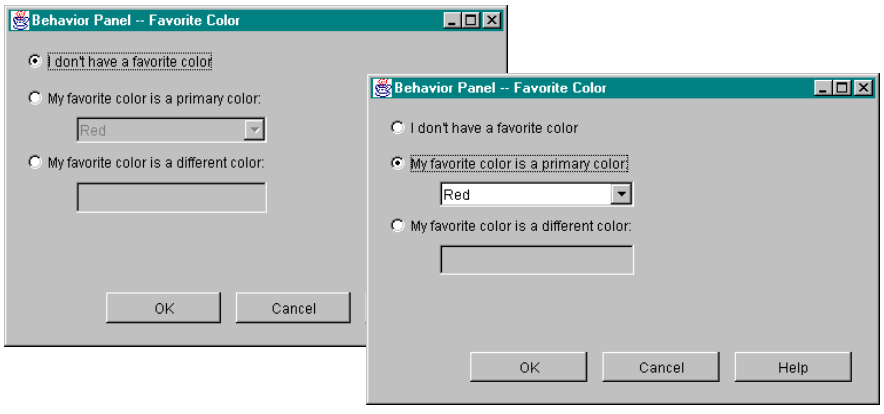
IBM @server. For the next generation of e-business.

Additional Functions

Selected & Deselected

IBM @server iSeries

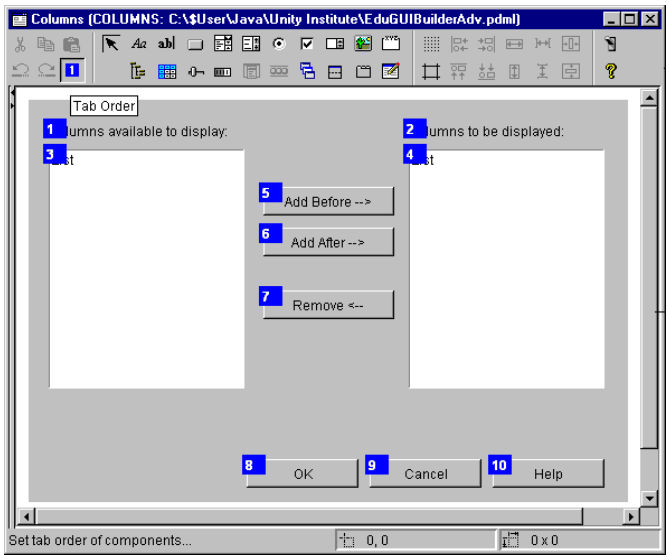
- **Selected & Deselected:**
  - You can add behaviors to these fields, such as disabling an edit box when it's associated radio button is not selected.



Tab Order

IBM eserver iSeries

- **Tab Order**
  - You can use the GUI Builder to set up the tabbing order through your windows. Tabbing order is important for accessibility.

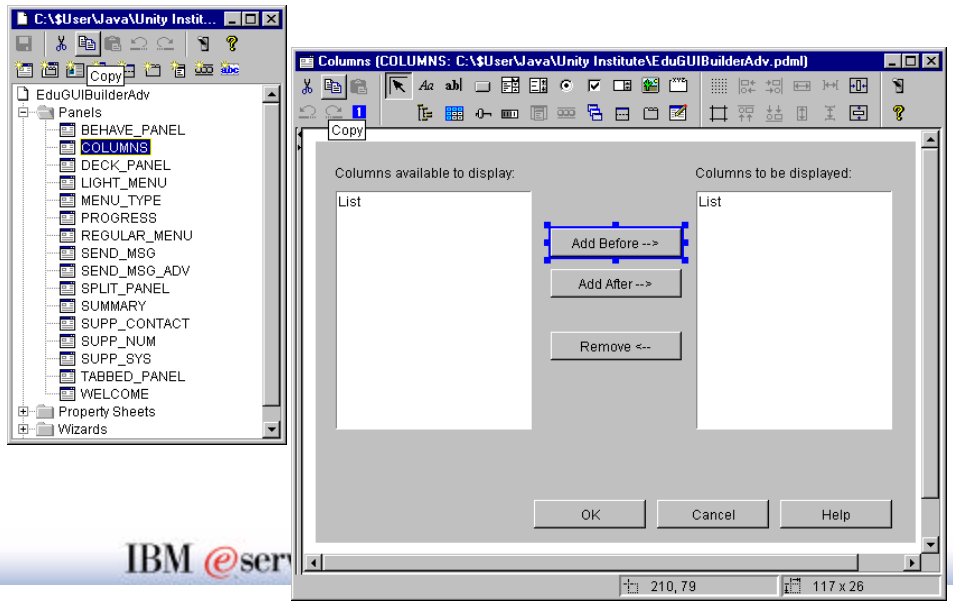


IBM eserver. For the next generation of e-business.

Cut, Copy, Paste

IBM eserver iSeries

- **Cut, Copy, Paste to copy**
  - Fields to other panels
  - Panels, Property Sheets, Wizards, Panes, Menus, or the String Table to another file

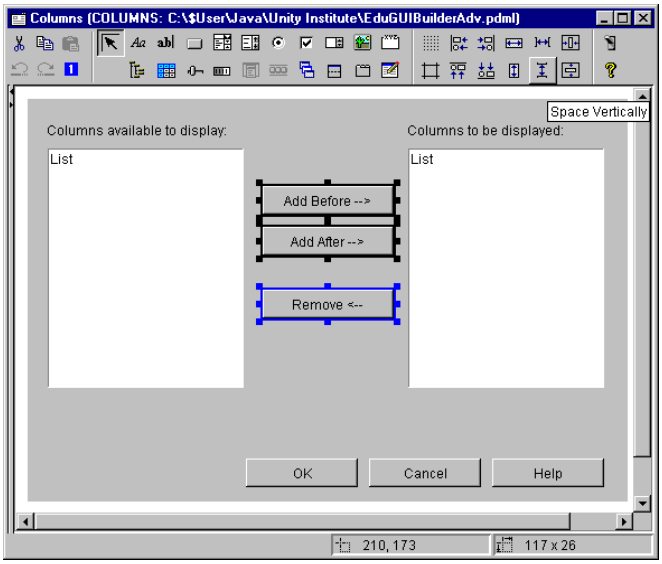


IBM eser

Alignment & Spacing

IBM @server iSeries

- **Alignment and Spacing options**
  - Left, Right, Top, or Bottom align fields
  - Equalize Width or Height of selected fields
  - Space Horizontally or Vertically
  - Center Horizontally or Vertically
- **Multi-select the fields to enable the options**
  - Use CTRL key with mouse
- **Blue selected field is the reference field**
  - Alignment and equalizing size

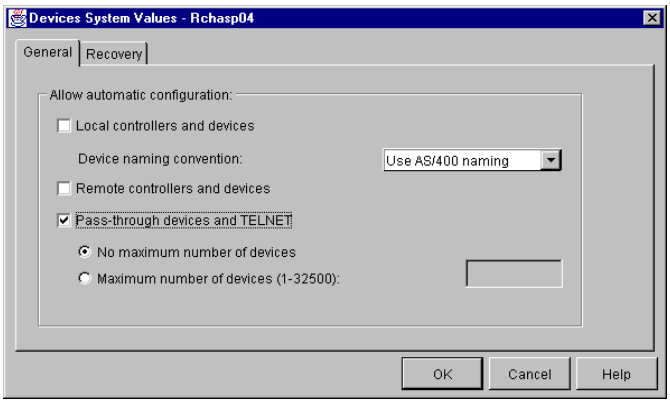


Types of Windows

Properties Sheets

IBM @server iSeries

- **Property Sheet contains Panels**
- **Each tab in the Properties sheet is a separate panel.**
  - Tab name is the panel title
- **OK, Cancel, and Help buttons are automatically added to the Property Sheet.**
- **To create a Property Sheet:**
  - Create a panel for each tab in the property sheet
  - Create a Property Sheet
  - Add the Panels to the Property Sheet

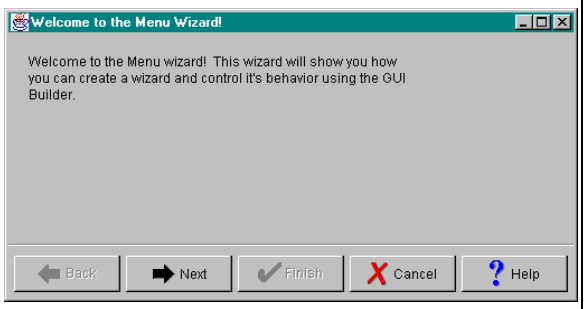


IBM @server. For the next generation of e-business.

Wizards

IBM @server iSeries

- **Wizards contain Panels**
  - Each wizard page is a separate panel
- **Back, Next, Finish, Cancel and Help buttons are added automatically**
- **You must write Java code (Wizard State Machine) to control the flow through the wizard.**
  - Use your Java development tool to do this part.
- **To create a Wizard:**
  - Create a panel for each page in the wizard
  - Create a Wizard
  - Add all of the Panels to the Wizard
  - Write a Wizard State Machine to control flow of wizard

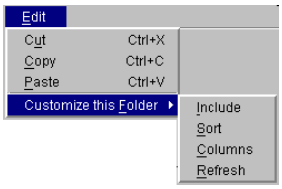


IBM @server. For the next generation of e-business.

Menus

IBM @server iSeries

- **Menus can be used for:**
  - Context menus
  - Within a Menu Bar
  - To build a Toolbar

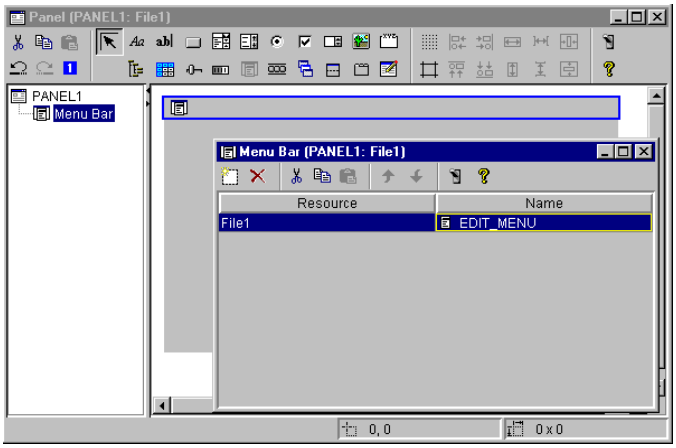


- **Types of menu items:**
  - menu item -- standard menu item
  - sub menu -- cascading menu item
  - check box -- toggling menu item (e.g. Status bar on View menu)
  - radio button -- selection menu item (e.g. Large Icon, Small Icon, List, Details on View menu)
  - Menu items can include icons on them (also used in toolbar)
- **To create a Menu, click on "Insert Menu" in the File window.**
  - Use "%" to add a mnemonic (underlined character) to a menu item
  - Click on "Insert Separator" to add separators to the menu

Menu Bars

IBM @server iSeries

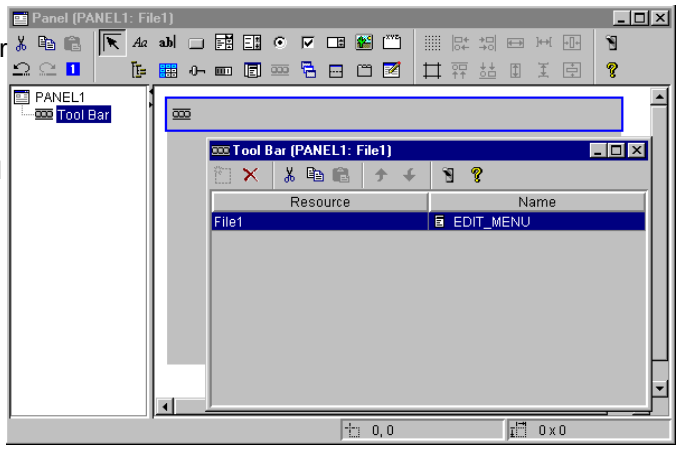
- **Menubars are built from menus**
- **Menubars can only be created on panels**
- **To create a Menu Bar:**
  - Create a menu for each drop down menu in the Menu Bar
  - Create the panel for the Menu Bar
  - Insert a Menu Bar on the Panel
  - Add all of the Menus to the Menu Bar



Toolbars

IBM @server iSeries

- **Toolbars are built from menu items**
- **Toolbars can only be created on panels**
- **The graphics on the toolbar is icon on the menu item**
- **To create a Tool Bar:**
  - Create a menu with all of your toolbar items on it
  - Create the panel for the Tool Bar
  - Insert a Tool Bar on the Panel
  - Add the Menu to the Tool Bar

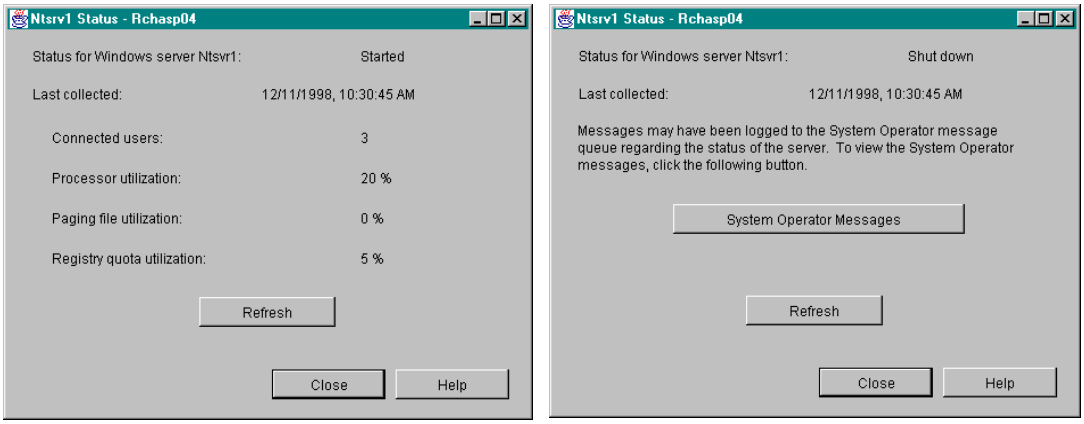


IBM @server. For the next generation of e-business.

Deck Panes

IBM @server iSeries

- **Deck Panes**
  - Allow you to conditionally show different panels within a panel. When you show the panel can either be determined by the state of a field or in your java code.
- **Example**
  - One status dialog with different information depending on the status. Use a deck pane to show one panel when started versus when stopped.

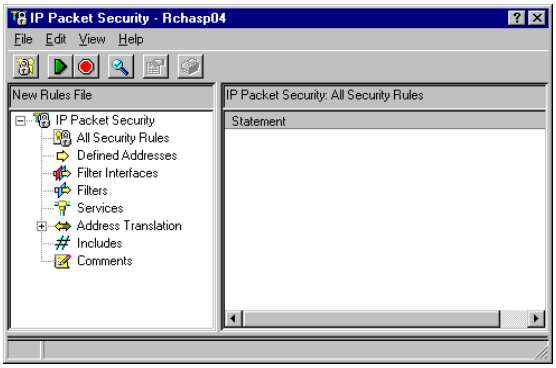


IBM @server. For the next generation of e-business.

Split Panes

IBM eserver iSeries

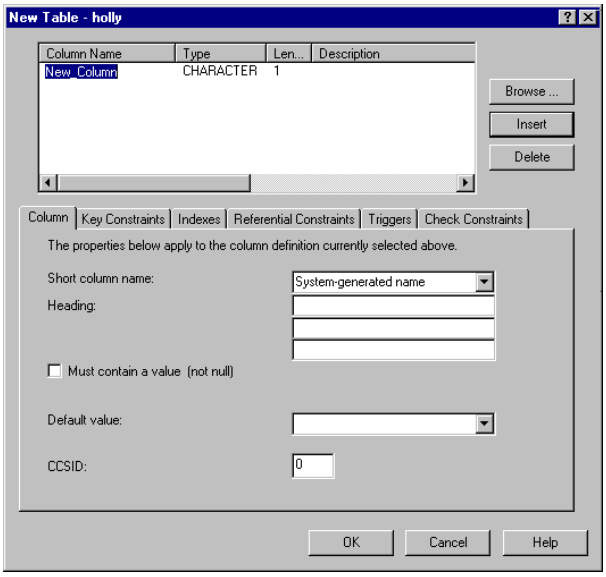
- **Split Panes**
  - Allow you to show different panels side by side.
- **Example**
  - A window similar to the Operations Navigator window would use a Split pane to show the left and right panes.



Tabbed Panes

IBM eserver iSeries

- **Tabbed Panes**
  - Allow you to imbed property sheets within panels.
- **Example**
  - A panel with a table of objects and then a set of tabbed properties below the table. (Similar to the C++ version of Create Table)



Creating Java Panes within a Panel

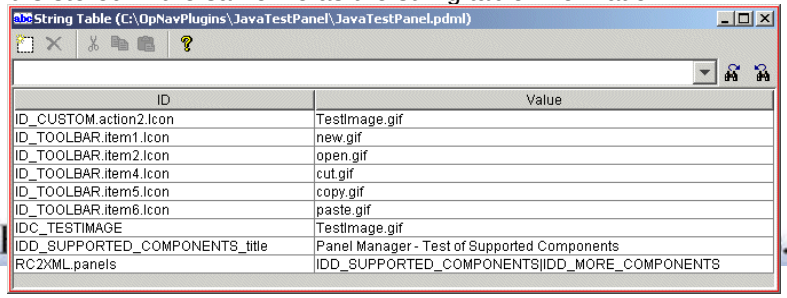
IBM eserver iSeries

- **There are two ways to create panes (deck, split, or tabbed):**
  - **Click on "Insert Deck Pane" in the File window**
    - Do this if you want to insert the pane into another pane.
    - For example, if you have a split pane with a tabbed pane inside of it, you would use this method to create the Tabbed Pane (not the split pane)
  - **Create a panel and click on "Deck Pane" to add a pane to the panel**
    - Do this when you are adding a pane to a panel.
    - For example, if you have a split pane, you would add the split pane to the panel and then specify which panels are in the split pane.

Managing Translatable Text

IBM eserver iSeries

- **GUI Builder provides the String Table function in order to manage translatable text**
- **Use the string table for:**
  - Messages
  - Operations Navigator Menu text and Status bar help for menu items
  - Additional status bar information (e.g. User Hlucke created)
- **What about the text on the panels?**
  - Field text on each panel can be translated
  - Field text does not need to be added to the string table
  - Field text is stored in the same file as the string table information



What types of files does it generate?

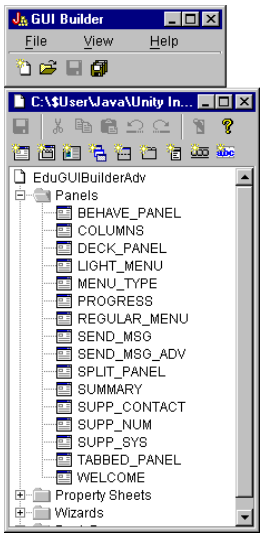
Behind the Scenes -- The Files

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

Putting Panels Into Files

IBM @server iSeries

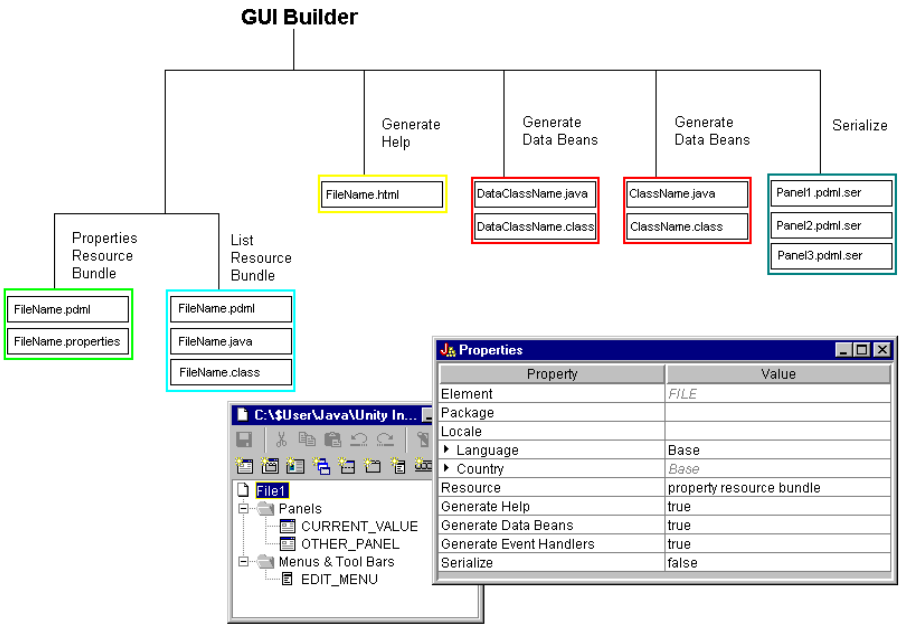
- **Should I split up my panels into multiple files?**
  - Split up the panels based on convenience:
    - n panels == 1 file:
      Easier to manage all of your panels if they are in one file.
    - n panels == 2-3 files:
      You may want to split them into separate files if there are multiple people working on the development. This avoids the circumstance where multiple people need to make changes to the PDML at the same time.
    - n panels == n files:
      It is not recommended that you place each panel in a separate file. You'll spend a lot of time managing the files themselves
  - Size of the PDML file does NOT affect run-time performance. Serialized panel files (not the PDML files) are used during run-time.
  - Each PDML file == HTML file for Help



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

Files generated by GUI Builder

IBM @server iSeries



Files generated by GUI Builder

IBM @server iSeries

- **Properties Resource Bundle**
  - By default, it uses the Properties Resource Bundle which will create two files for you:
  - FileName.pdml -- holds the panel definition information, such as the type of fields, their location, etc.
  - FileName.properties -- holds all of the translatable text
- **List Resource Bundle**
  - Used to enhance performance of the application.
  - Compiles the panel definitions and translatable text, so performance is better at run-time. However, it will slow down performance in the GUI Builder when previewing panels.
  - This bundle creates three files:
    - FileName.pdml -- holds the panel definition information, such as the type of fields, their location, etc.
    - FileName.java -- holds all of the translatable text
    - FileName.class -- compiled version of FileName.java

Files generated by GUI Builder

IBM @server iSeries

- **Generating Data Beans**
  - Generates data beans (get and set methods) for fields in the PDML.
  - Makes it easy for you to get started on the Java code for the panels.
  - Creates these files for you:
    - DataClassName.java (for each data class specified in the PDML file) -- file with get and set methods for each attribute in the data class
    - DataClassName.class (for each data class specified in the PDML file) -- compiled version of the corresponding .java file
- **Generating Help**
  - Generates ONE help skeleton for the entire PDML file. This help skeleton will be used to write the help.
  - Does not overwrite any existing Help in the Help skeleton.
  - Includes topics for all of the fields specified by the developer to generate help for
  - Creates this file for you:
    - FileName.html

Files generated by GUI Builder

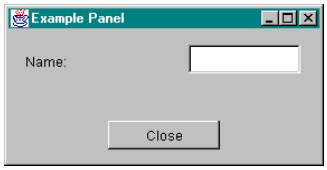
IBM @server iSeries

- **Generates Event Handlers for:**
  - Button Handlers
  - Activation Handlers
  - Double-click Handlers
  - Slider Handlers
- **Context Menu Handlers**
  - Makes it easy for you to get started on the Java code for the panels.
  - Creates these files for you:
    - ClassName.java (for each class specified in the PDML file) -- file with EventHandler code
    - ClassName.class (for each class specified in the PDML file) -- compiled version of the corresponding .java file
- **Serialization**
  - Compresses the information for each window in the PDML file.
  - Enhances performance of your Java panels.
  - Creates these files for you:
    - PanelName.pdml.ser (for each panel in the PDML file)

Data Bean Generation

IBM @server iSeries

- **Creates data bean skeletons (get and set methods) for each data class and attribute specified in the PDML file.**
- **Change the File Properties to Generate Data Beans (true/false)**

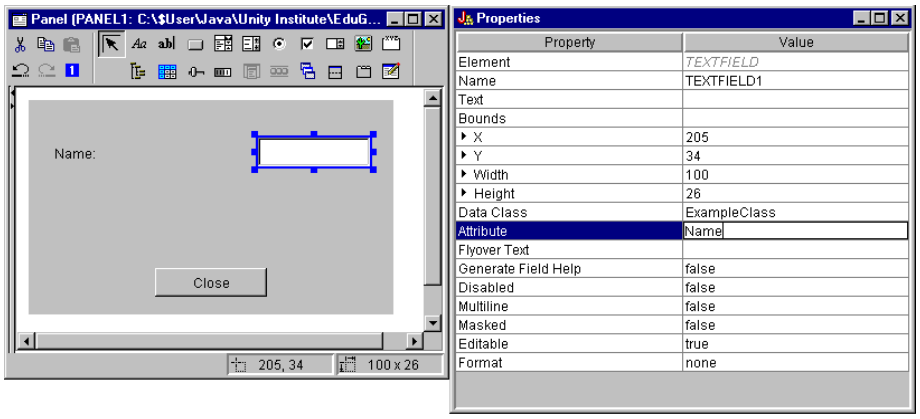



```
import com.ibm.as400.ui.framework.java.*;

public class ExampleClass extends Object implements DataBean
{
   private String m_sName;

  public String getName()
   {      return m_sName;    }

   public void setName(String s)
   {      m_sName = s;   }

   public Capabilities getCapabilities()
   {      return null;   }

   public void verifyChanges()
   {  }

   public void save()
   {  }

   public void load()
   {      m_sName = "";   }
}
```


Data Bean Generation

IBM @server iSeries

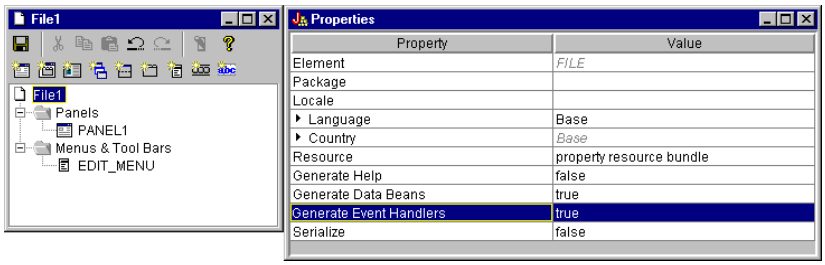
- **Data Class and Attributes**
  - Sends and receives information to and from individual fields on the panels
  - Specify the Data Class and Attribute in the Properties for each field



Event Handler Generation

IBM eserver iSeries

- **Creates event handler skeletons for each type of event handler specified in the PDML file.**
- **Change the File Properties to Generate Event Handlers (true/false)**

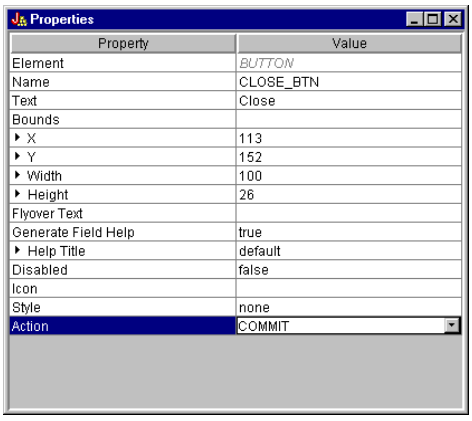


IBM eserver. For the next generation of e-business.

Event Handler: Button

IBM eserver iSeries

- **Button Actions**
  - Specify each button's Action handler in the Properties (by specifying the class name)
  - Three Button Action's are available without having to write a button handler (must specify in Properties):
    - COMMIT -- For OK buttons
    - CANCEL -- For Cancel buttons
    - HELP -- For Help buttons

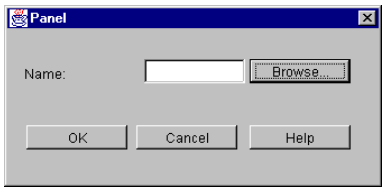


IBM eserver. For the next generation of e-business.

Event Handler: Button

IBM eserver iSeries

- **Button Handler**
  - Extends Event Handler
  - Implements ActionListener

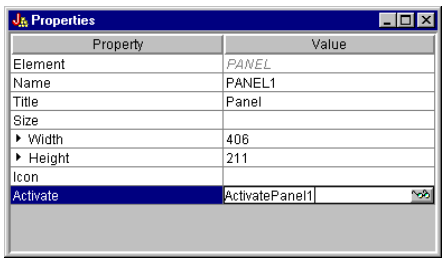



```
import com.ibm.as400.ui.framework.java.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class BrowseName extends EventHandler
   implements ActionListener
{
   public BrowseName(PanelManager pm)
   {
      super(pm);
   }
   public void actionPerformed(ActionEvent e)
   {
      System.out.println(e);
   }
}
```


Event Handler: Activation

IBM eserver iSeries

- **Activation Handler**
  - Used to specify what you want to do when the panel appears (hide fields, resize the panel, etc.)
  - Specify the class name in the Activate property for the panel
  - Extends Event Handler
  - Implements ActionListener

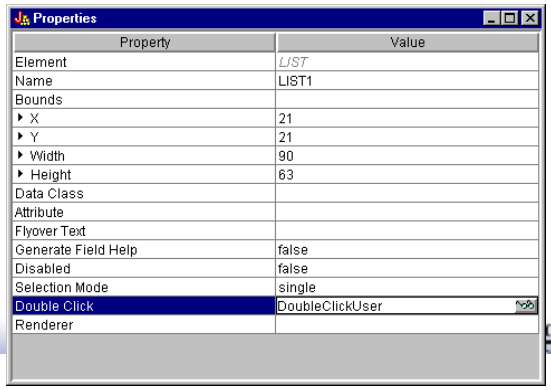



```
import com.ibm.as400.ui.framework.java.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class ActivatePanel1 extends EventHandler
   implements ActionListener
{
   public ActivatePanel1(PanelManager pm)
   {
      super(pm);
   }
   public void actionPerformed(ActionEvent e)
   {
      System.out.println(e);
   }
}
```

Event Handler: Double-click

IBM @server iSeries

- **Double-Click Handler**
  - Used to specify what you want to do when the user double-clicks on an item. For example, launch an edit panel when the user double clicks on a user in a list.
  - Specify the class name in the Double-Click property for a List, Table, or Tree
  - Extends Event Handler
  - Implements ActionListener

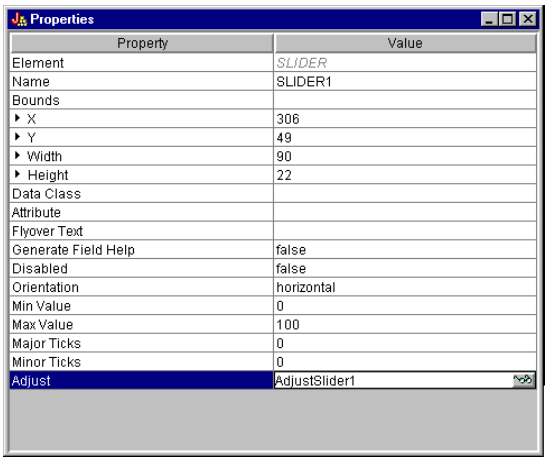



```
import com.ibm.as400.ui.framework.java.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class DoubleClickUser extends EventHandle
   implements ActionListener
{
   public DoubleClickUser(PanelManager pm)
   {
      super(pm);
   }
   public void actionPerformed(ActionEvent e)
   {
      System.out.println(e);
   }
}
```


Event Handler: Slider

IBM @server iSeries

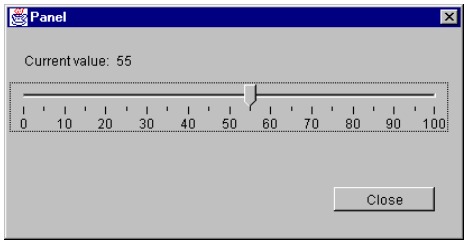
- **Slider Handler**
  - Used to specify what you want to do when the user changes the value in the slider control on a panel.
  - Specify the class name in the Adjust property for a Slider



Event Handler: Slider

IBM eserver iSeries

- **Slider Handler**
  - Extends Event Handler
  - Implements ChangeListener



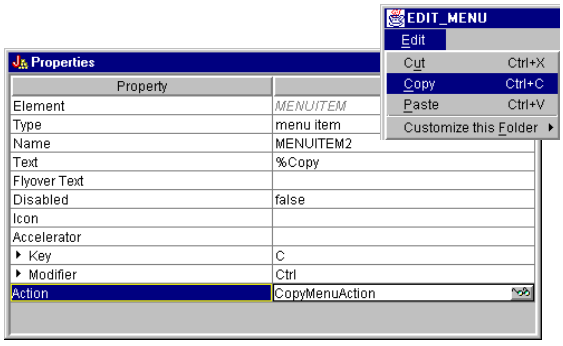

```
import com.ibm.as400.ui.framework.java.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class AdjustSlider1 extends
EventHandler
   implements ChangeListener
{
   public AdjustSlider1(PanelManager pm)
   {
      super(pm);
   }
   public void stateChanged(ChangeEvent e)
   {
      System.out.println(e);
   }
}
```

IBM eserver. For the next generation of e-business.

Event Handler: Context Menu

IBM eserver iSeries

- **Context Menu Handler**
  - Used to specify what you want to do when the user selects the menu item.
  - Specify the class name in the Action property for each menu item.
  - Extends Event Handler
  - Implements ActionListener



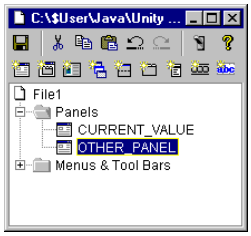

```
import com.ibm.as400.ui.framework.java.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class CopyMenuAction extends
ActionHandler
   implements ItemListener
{
   public CopyMenuAction(MenuManager mm)
   {
      super(mm);
   }
   public void actionPerformed(ActionEvent e)
   {
      System.out.println(e);
   }
   public void itemStateChanged(ItemEvent e)
   {
      System.out.println(e);
   }
}
```

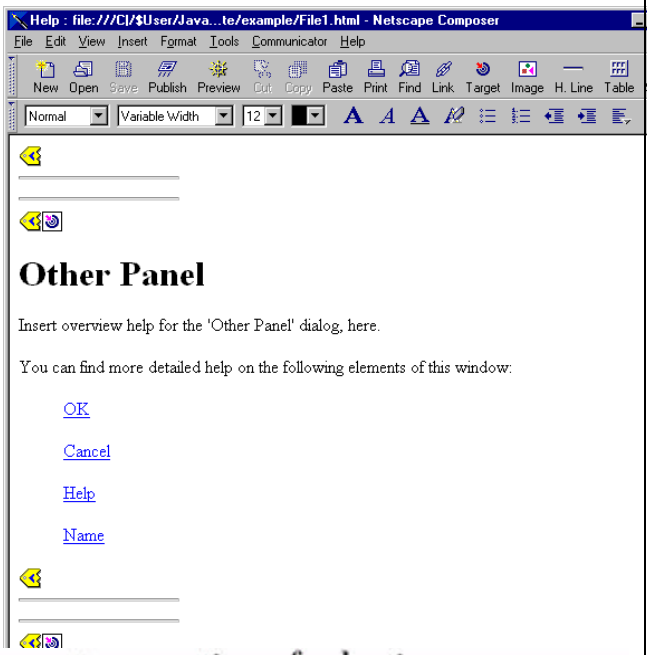
IBM eserver. For the next generation of e-business.

Help Generation

IBM @server iSeries

- **Creates ONE help skeleton for all of the panels in the PDML file**
- **Help skeletons can be used by the help writer (not necessarily the developer) to build the help for each panel**





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

Help Generation

IBM @server iSeries

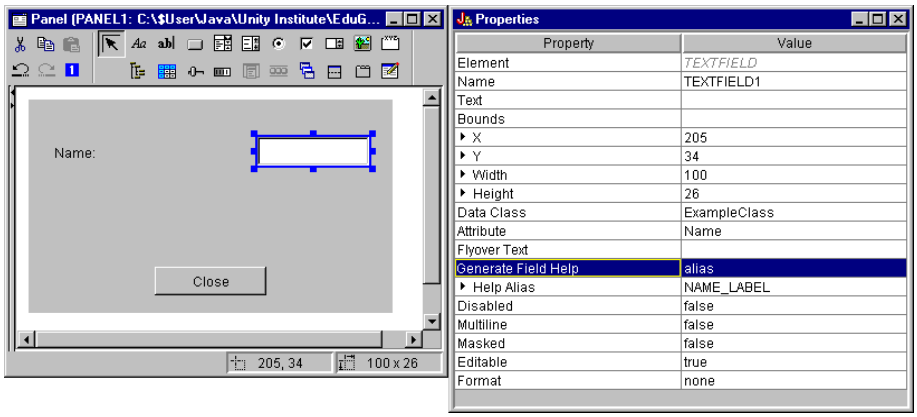
- **Generate Help & Help Title**
  - Help Generation is now developer-specified in the properties for each field.
  - You can generate help for any field. The Help title specifies what text will be used as the topic ID in the help file. "DEFAULT" indicates that it will use the Label text.
  - The Generate Help condition is set to TRUE by default for the following fields:
    - radio buttons
    - checkboxes
    - group boxes
    - buttons
    - label

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

Help Generation

IBM @server iSeries

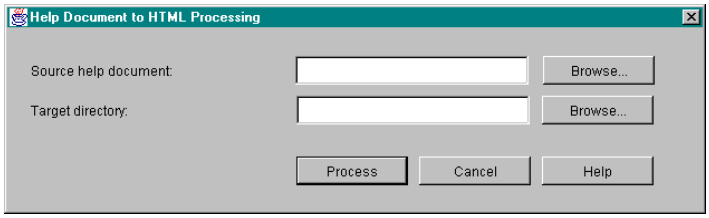
- **Help Alias**
  - Points to another field which already has help, such as a label that ends with a colon
  - Specify the Help alias in the Properties



Help Generation

IBM @server iSeries

- **hdoc2htmViewer tool**
  - Located in uitools.jar: com.ibm.as400.ui.tools.hdoc2htmViewer
  - Use this tool after the help is written to separate the help file into multiple files
  - This tool must be run for Help to appear during run-time.



Summary

IBM @server iSeries

- **Overview**
  - XML & PDML overview
  - PCML
  - How it all fits together
  - Operations Navigator plug-in support
  - Where to get more information
- **What is the Java GUI Builder**
  - Advantages
  - Java development tools
  - GUI Builder requirements
  - Customizing the GUI Builder
- **Creating & editing panels**
  - Window types
  - Control / field types
  - Field properties
  - Recommended Naming conventions
  - Selected & Deselected
  - Tab order
  - Cut, copy, paste
  - Alignment and spacing
- **Window Types**
  - Property sheets
  - Wizard
  - Menus & Menu bars
  - Toolbars
  - Deck panes
  - Split panes
  - Tabbed panes
- **Generating Files**
  - Generated files
  - Translatable text
  - Databean generation
  - Event handler generation
    - Buttons
    - Activation
    - Double-click
    - Slider
    - Context menu
  - Help generation
- **Exercises**

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

Summary

IBM @server iSeries

Any questions?

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

Exercises: Using the GUI Builder

Hands-on practice with the GUI Builder

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

Exercises

IBM @server iSeries

- **These exercises are intended to give you some experience with the GUI Builder so that you are familiar with how it works**
- **The Objectives of these Exercises:**
  - Install the GUI Builder (at least V4R5) and get it up and running
  - Build a simple panel
  - Try different types of fields
  - Add conditional behaviors to the fields
  - Update the String Table

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

Exercises

Simple Panels & String Table

Exercises -- Simple Panels & String Table

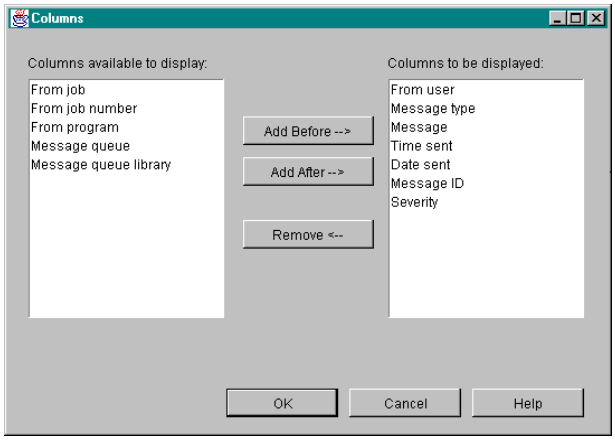
IBM @server iSeries

- **Build the following panels. After building each panel, preview the panel and verify that you can answer all of the questions for each panel.**
  - Tips to keep in mind while creating the panels:
    - Right click -- there are context menus in many places
    - Double click to edit a field
    - Use the Preview function to verify the panel is working as expected
    - F1 can be used to bring up help

Exercises -- Panels

IBM @server iSeries

- **Panel: Columns**



Preview questions

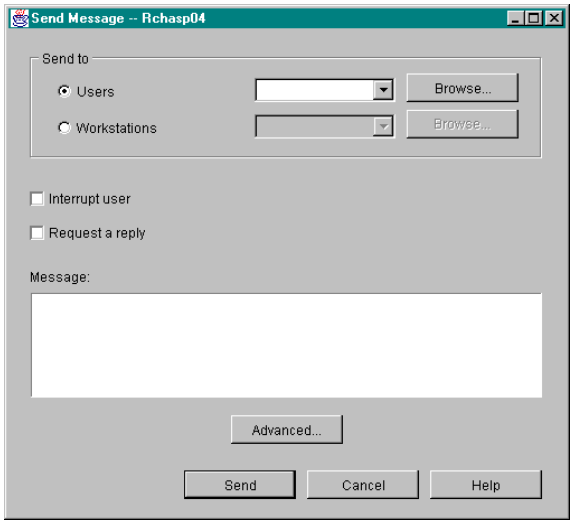
- ✔ Can you select multiple items in the listboxes?
- ✔ Is OK the "default" button (extra black outline around it)?
- ✔ Is the tab order correct (see tabbing order slide)?

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

Exercises -- Panels

IBM @server iSeries

- **Panel: Send Message**



Preview questions

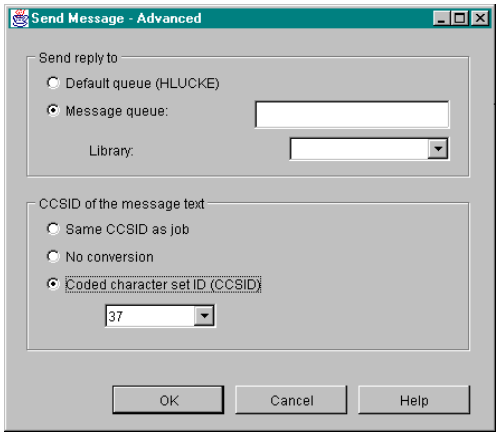
- ✔ Can you only select one radio button at a time?
- ✔ Can you check both checkboxes?
- ✔ Is Advanced centered in the panel?
- ✔ Can you type multiple lines in the Message area?
- ✔ Can you type a user name or workstation name into the drop down boxes?

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

Exercises -- Panels

IBM @server iSeries

- **Panel: Send Message - Advanced**



Preview questions

- ✔Can you only select one radio button for each group box?
- ✔Can you type in a library name?
- ✔Do both comboboxes contain the right choices?
- ✔Can you type in a Message queue?

For Library, use the following choices in the drop down:

- Library list, Current library

For CCSID, add a choice in the drop down for each of the following numbers:

- 37, 256, 273, 277, 278, 280, 284

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

Exercises -- String Table

IBM @server iSeries

- **String table -- Add the following strings to the string table**

  - ▶Are you sure you want to delete user DSchmidt?
  - ▶P&roperties
  - ▶Displays the properties of the selected items.
  - ▶{0} of {1} files copied to Clipboard

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

Exercises

Advanced Panels & Menus

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

Exercises -- Advanced Panels & Menus

IBM @server iSeries

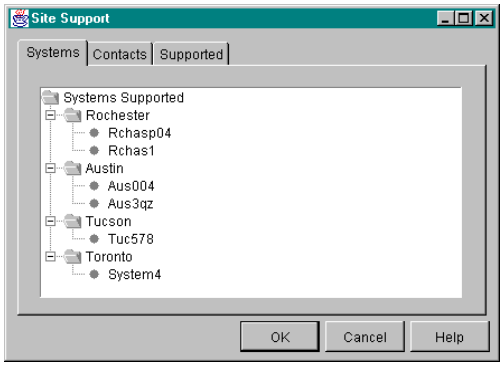
- **The following exercises demonstrate some of the advanced features of the GUI Builder. These exercises are recommended, but are not required.**
- **The following exercises demonstrate how to create and use a**
  - Wizard
  - Tabbed Pane
  - Split Pane
  - Deck Pane
  - Menu
- **Keep in mind that throughout the next few exercises, you will be able to reuse many of the panels for each different type of advanced panel.**

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

Exercises -- Properties

IBM @server iSeries

- **Panel: Site Support -- Systems**



Preview questions

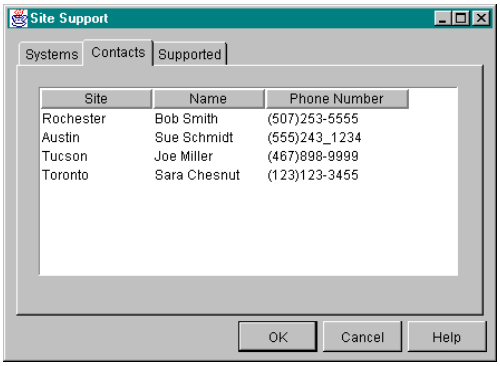
- ✔ Does the panel title say "Site Support"?
- ✔ Does the tab title say "Systems"?

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

Exercises -- Properties

IBM @server iSeries

- **Panel: Site Support -- Contacts**



Make all columns, except "Site" editable.

Preview questions

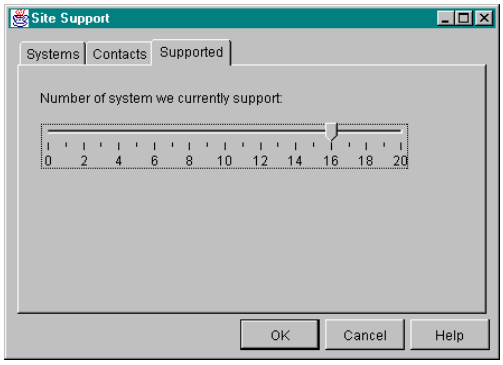
- ✔ Can you change the information about the contact?
- ✔ Is the column for the site read-only?

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

Exercises -- Properties

IBM @server iSeries

- **Panel: Site Support -- Support**



Preview questions

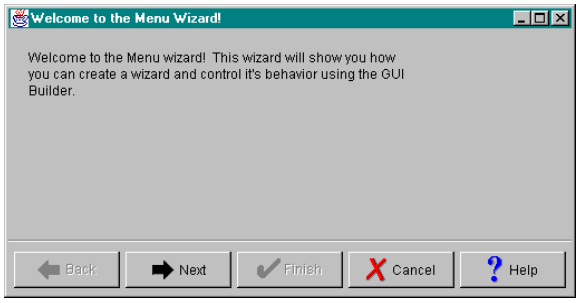
- ✔ Do the tick marks show up correctly?

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

Exercises -- Wizard

IBM @server iSeries

- **Wizard: Restaurant -- Welcome Panel**



Preview questions

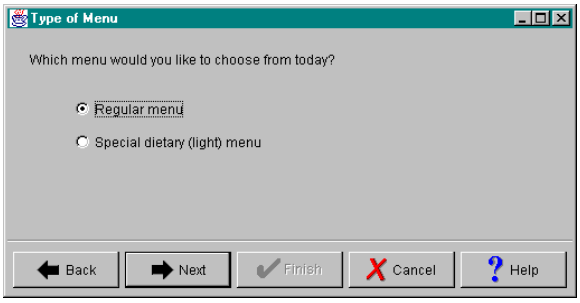
- ✔ Have you told the wizard not to show the steps?

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

Exercises -- Wizard

IBM @server iSeries

- **Wizard: Restaurant -- Menu Type Panel**



Preview questions

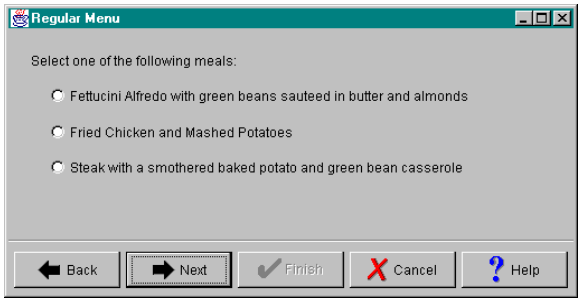
- ✔ Can you only select one radio button?

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

Exercises -- Wizard

IBM @server iSeries

- **Wizard: Restaurant -- Regular Menu Panel**



Preview questions

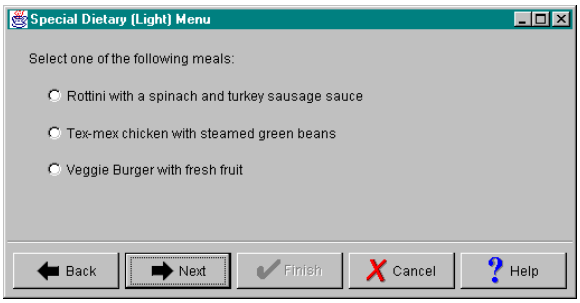
- ✔ Can you only select one radio button?

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

Exercises -- Wizard

IBM @server iSeries

- **Wizard: Restaurant -- Light Menu Panel**



Preview questions

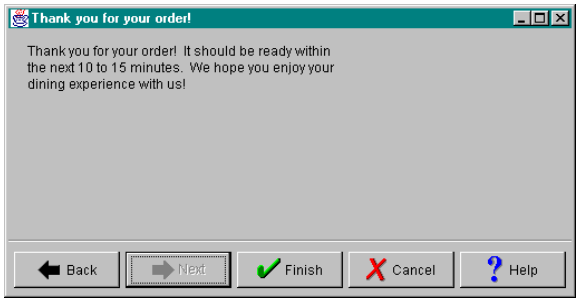
- ✔ Can you only select one radio button?

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

Exercises -- Wizard

IBM @server iSeries

- **Wizard: Restaurant -- Summary Panel**

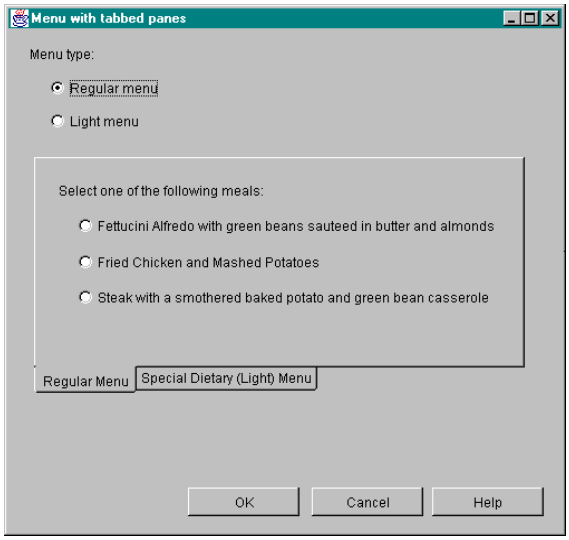


IBM @server. For the next generation of e-business.

Exercises --Tabbed Pane

IBM @server iSeries

- **Tabbed Panes: Restaurant**



Preview questions

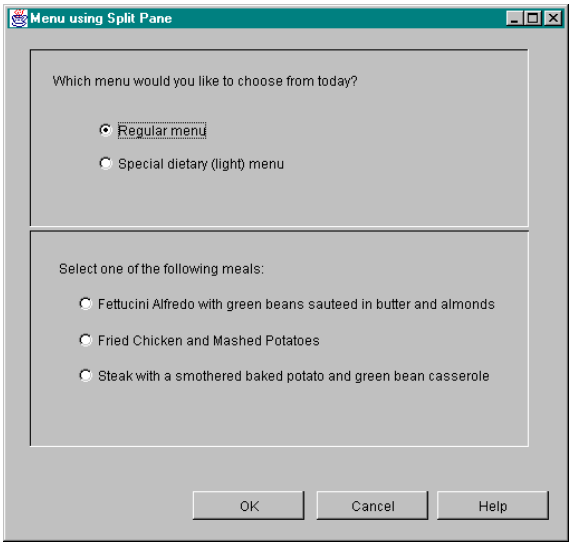
- ✔ Are the tabs displayed at the bottom left corner?
- ✔ Are the OK, Cancel, and Help buttons displayed?
- ✔ Do the tabs change when you switch between the menu type radio buttons?

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

Exercises --Split Pane

IBM @server iSeries

- **Split Pane: Restaurant**



Preview questions

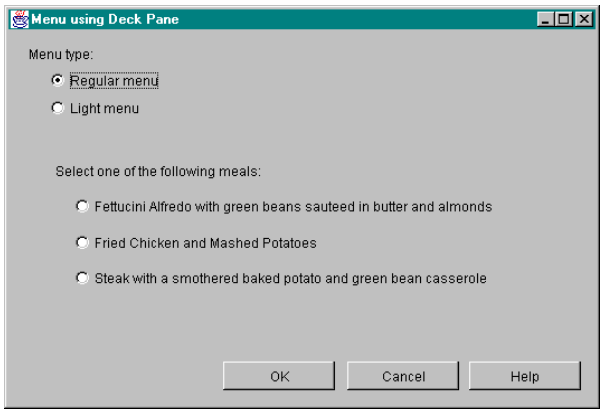
- ✔ Are the OK, Cancel, and Help buttons displayed?
- ✔ Does the menu in the bottom pane change when you switch between the menu type radio buttons? (Hint: the bottom pane needs to be a deck pane in order for it to change)

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

Exercises --Deck Pane

IBM @server iSeries

- **Deck Pane: Restaurant**



Preview questions

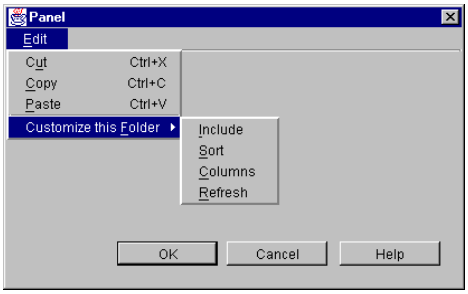
- ✔ Are the OK, Cancel, and Help buttons displayed?
- ✔ Does the menu change when you switch between the menu type radio buttons? (Hint: you need to create a deck pane on the panel in order for it to switch)

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

Exercises -- Menu

IBM @server iSeries

- **Panel with a menubar**



Preview questions

- ✔ Does the menu bar item say "Edit"?
- ✔ Do all of the menu options have mnemonics (underlined characters)?
- ✔ Do Cut, Copy, and Paste have accelerators?
- ✔ Is "Customize this Folder" a cascading menu?

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

Exercises

IBM eserver iSeries

Congratulations!

You have completed the GUI Builder exercises.

IBM eserver. For the next generation of e-business.

Trademarks

IBM eserver iSeries

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 the IBM Corporation in the United States or other countries or both:

ADSTAR	DataGuide	NetFinity
Advanced Function Printing	DataPropagator	NetView
AFP	DB2	OfficeVision
AIX	IBM	OS/2
AnyNet	IBM Network Station	OS/400
Application Development	Information Warehouse	PowerPC
APPN	Integrated Language Environment	PowerPC AS
AS/400	Intelligent Printer Data Stream	Print Services Facility
AT	IPDS	PSF
BrioQuery	JustMail	SanFrancisco
BRMS	Net.Commerce	SmoothStart
Client Series	Net.Data	SystemView

cc:Mail, Lotus, Lotus Notes, Lotus Domino, Domino.Action, and Domino.Merchant are trademarks or registered trademarks of Lotus Development Corporation in the United States or other countries or both.
Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.
UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.
IBM's VisualAge products and services are not associated with or sponsored by Visual Edge Software, Ltd.
Intel and Pentium are trademarks of Intel Corporation in the United States and other countries.
Tivoli is a registered trademark of Tivoli Systems Inc. in the United States or other countries or both.
Other company, product, and service names may be trademarks or service marks of others.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources. 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.

Any performance data contained in this document was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurement quoted in this presentation may have been made on development level systems. There is no guarantee these measurements will be the same on generally available systems. Some measurements quoted in this presentation may have been estimated through extrapolation. Actual results may vary. Users of this presentation should verify the applicable data for their specific environment. Customer examples sited are examples of how the referenced customers use IBM and other products. Results vary by environment and may not be realized in all situations.

THIS MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS MATERIAL INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Functions and availability dates may change after this presentation was completed.

IBM eserver. For the next generation of e-business.