IBM ILOG Views

Studio V5.3

User's Manual

June 2009

C O N T E N T S

Table of Contents

About This Manual

This *User's Manual* explains how to use IBM® ILOG® Views Studio, a GUI builder tool used to design IBM ILOG Views applications.

What You Need to Know

This manual assumes that you are familiar with the PC or UNIX® environment in which you are going to use IBM ILOG Views Studio, including its particular windowing system. Since IBM ILOG Views Studio is written for C++ developers, the documentation also assumes that you can write C++ code and that you are familiar with your C++ development environment so as to manipulate files and directories, use a text editor, and compile and run C++ programs.

Manual Organization

This manual contains the following chapters:

◆ Chapter 1, *Introducing IBM ILOG Views Studio* provides set-up information, explains how to launch IBM® ILOG® Views Studio, and briefly presents the panels that appear at start-up.

◆ Chapter 2, *Using ILOG Studio* helps you get acquainted with the basic components of IBM ILOG Views Studio and explains how to perform the basic tasks you will need to create graphic objects. You will learn how to edit these objects using various tools and commands available in IBM ILOG Views Studio.

◆ Chapter 3, *The IBM ILOG Views Studio Interface* provides a detailed description of the IBM ILOG Views Studio interface.

◆ Chapter 4, *Using IBM ILOG Script* introduces you to an extension of IBM ILOG Views Studio designed to facilitate the writing and editing of IBM ILOG Script-based callbacks that can be attached to IBM ILOG Views objects.

◆ Chapter 5, *Commands* provides a description of the commands available in IBM ILOG Views Studio.

◆ Chapter 6, *Panels* describes the various panels and dialog boxes that you will encounter when usingIBM ILOG Views Studio.

◆ Chapter 7, *Customizing IBM ILOG Views Studio* provides information on how to customize IBM ILOG Views Studio using configuration files.

Notation

Typographic Conventions

The following typographic conventions apply throughout this manual:

◆ Code extracts and file names are written in `courier` typeface.

◆ Entries to be made by the user are written in `courier` typeface.

Naming Conventions

Throughout this manual, the following naming conventions apply to the API.

◆ The names of types, classes, functions, and macros defined in the ILOG Views Foundation library begin with `Ilv`.

◆ The names of classes as well as global functions are written as concatenated words with each initial letter capitalized.

```
class IlvDrawingView;
```

◆ The names of virtual and regular methods begin with a lowercase letter; the names of static methods start with an uppercase letter. For example:

```
virtual IlvClassInfo* getClassInfo() const;
```

```
static IlvClassInfo* ClassInfo*() const;
```

Related Documentation

The IBM ILOG Views *Studio Reference Manual* provides a description of the IBM ILOG Views Studio C++ classes.

The following IBM ILOG Views manuals may be helpful to you when using the Foundation package of ILOG Views:

◆ For a description of the IBM ILOG Views C++ classes, global functions, type definitions, macros, and error message, see the IBM ILOG Views *Foundation Reference Manual*.

◆ The IBM ILOG Views *Foundation User's Manual* provides helpful information and numerous

1

Introducing IBM ILOG Views Studio

This chapter introduces you to IBM® ILOG® Views Foundation Studio. You can find information on the following topics:

◆ *What is Foundation Studio?*

◆ *Installation Directory*

◆ *Setting Up and Running IBM ILOG Views Studio on UNIX Systems*

◆ *Setting Up and Running IBM ILOG Views Studio on Microsoft Windows Systems*

◆ *A Quick Look at the Interface*

What is Foundation Studio?

IBM® ILOG® Views Foundation Studio is a customizable and extendable WYSIWYG editor that lets you interactively draw IBM ILOG Views 2D graphics. IBM ILOG Views Foundation Studio provides the basic tools that you will need to draw the 2D graphic objects that will become the contents of container or manager objects of IBM ILOG Views. The Foundation Studio also provides support for the IBM ILOG Script language.

Other IBM ILOG Views packages come with their own IBM ILOG Views Studio extensions that let you handle the additional features of the package.

With the IBM ILOG Views Gadgets package, you can use the GUI Application Studio Extension to edit your GUI using the comprehensive set of IBM ILOG Views Gadgets. You can use the GUI Generation Studio Extension to generate the C++ code for the GUI part of your application.

Installation Directory

Before running IBM® ILOG® Views, you must install it in the appropriate directory and configure the ILOG License Manager (ILM) provided with the product as explained in its documentation.

Depending on the platform you are using, install IBM ILOG Views in the following directory:

On UNIX®: `/usr/ilog/viewsXX`

On Microsoft® Windows®: `C:\ILOG\VIEWSXX`

where `XX` stands for the version number.

If you decide to install IBM ILOG Views in another directory, replace subsequent references to the default directory with your actual installation directory

For the latest list of platforms on which you can use IBM ILOG Views, see the table in the `README` file delivered with the product.

Setting Up and Running IBM ILOG Views Studio on UNIX Systems

Use the information in this section to set up and run IBM® ILOG® Views Studio on your UNIX® system.

Setting the ILVHOME Environment Variable

To launch IBM ILOG Views Studio on UNIX for the first time, you must set the environment variable `ILVHOME` to the root directory whereIBM ILOG Views was installed. See the previous section "Installation Directory" for information.

Running IBM ILOG Views Foundation Studio

IBM ILOG Views Foundation Studio, a generic version of IBM ILOG Views Studio using shared libraries, is provided in the following directory:

`$ILVHOME/studio/<system>/<subsystem>`

To run this editor, set the environment variable `LD_LIBRARY_PATH` to the directory:

```
$ILVHOME/lib/<system>/<subsystem>
```

and

```
$ILVHOME/studio/<system>/<subsystem>
```

where `<system>` indicates the name of the platform you are using, such as `sparc_5_5.0`, `i86_linux2_glic2.1_egcs1.1`, and so on. `<subsystem>` indicates the name of the subplatform you are using, such as `static_pic`, `shared`, and so on.

Loading Plug-Ins

A plug-in is an IBM ILOG Views dynamic module that creates an IBM ILOG Views Studio extension.

Selecting Plugs-Ins

When you use `ivfstudio` for IBM ILOG Views for the first time, `ivfstudio` prompts you to select the plug-ins it has detected.

Figure 1.1   *IBM ILOG Views Studio Plug-Ins Dialog Box (UNIX® systems)*

Select the plug-ins you want to load by checking the corresponding toggle buttons. Then click OK. The plug-ins you use in this session will be loaded again the next time you launch `ivfstudio`.

If you click Cancel, `ivfstudio` starts without loading any plug-ins.

Using the -selectPlugIns Command Line Argument

If you want to select the plug-ins before `ivfstudio` starts running, you can launch `ivfstudio` with the `-selectPlugIns` argument. The IBM ILOG Views Studio Plug-Ins dialog box will appear and you can specify again the plug-ins you want to load.

Using the SelectPlugIns Command

Studio provides you with the `SelectPlugIns` command that you can use to select plug-ins. To use this command, choose SelectPlugIns from the Tools menu in the Main window. Studio displays the IBM ILOG Views Studio Plug-Ins dialog box for you to make your selections. You will then have to exit Studio and launch it again to load the plug-ins you just selected.

Specifying Plug-Ins Using the Environment Variables

You can also specify either a list of the modules you want to load or a directory containing all the modules you want to load. When you do this, the plug-ins loaded in the previous session are ignored.

Before initializing itself, Studio loads all the extension modules contained in a plug-in directory. The plug-in directory defaults to `<ILVHOME>/studio/plugins`. If you want to put your plug-ins in another directory, you can use the `ILVSTPLUGINSDIR` environment variable to indicate the directory. After loading the modules found in that plug-in directory, Studio looks for a subdirectory named `<system>`.

You can use the `ILVSTPLUGINS` environment variable to indicate a list of the modules you want to load. You can specify the full paths or the paths relative to the working directory, separated by a ";" character (a semicolon character).

Common Plug-Ins for IBM ILOG Views Studio

Some IBM ILOG Views packages come with their own plug-ins for IBM ILOG Views Studio. For more information, see the documentation of the IBM ILOG Views packages you are using.

The following is a list of some common plug-ins located in the `$ILVHOME/studio/<system>/<subsystem>` directory:

◆ If you have the IBM ILOG Views Gadgets package, you can use the IBM ILOG Views Studio GUI Application plug-in to edit GUI applications with gadgets panels. The corresponding dynamic package is `libsmguiapp.<ext>`, where `<ext>` is `.so`, `.sl`, and so on, according to your platform. You can also use the IBM ILOG Views Studio

GUI Generation plug-in to generate the C++ code for the GUI part of your application. The corresponding dynamic package is `libsmguigen.<ext>`, where `<ext>` is `.so`, `.sl`, and so on, according to your platform.

◆ If you have the IBM ILOG Views Grapher package, you can use the IBM ILOG Views Studio Grapher plug-in to edit your graphers, nodes, and links. The corresponding dynamic package is `libsmgrapher.<ext>`, where `<ext>` is `.so`, `.sl`, and so on, according to your platform.

◆ If you have the IBM ILOG Views Prototypes package, you can use the IBM ILOG Views Studio Prototypes plug-in to edit your business graphic objects. The corresponding dynamic package is `libsmproto.<ext>`, where `<ext>` is `.so`, `.sl`, and so on, according to your platform.

Building Other Executables of IBM ILOG Views Studio

If for any reason you have to build an executable of IBM ILOG Views Studio, go to the `$ILVHOME/studio/<system>/<subsystem>` directory and run the `make` utility passing the name of the executable you want to build as the only argument.

The provided `ivfstudio` can be used with the IBM ILOG Views Foundation product. However, you need the Gadgets and Manager packages to build any version of IBM ILOG Views Studio.

Setting Up and Running IBM ILOG Views Studio on Microsoft Windows Systems

Use the information in this section to set up and run IBM ILOG Views Studio on your Microsoft® Windows® system.

Setting the ILVHOME Environment Variable

To launch ILOG Studio on Microsoft Windows for the first time, you must:

◆ Set the environment variable `ILVHOME` to the root directory where IBM ILOG Views was installed.

or

◆ Put the line `IlvHome=<ViewsDir>` in the `VIEWS.INI` file, where `<ViewsDir>` is the root directory where IBM ILOG Views was installed.

Running IBM ILOG Views Foundation Studio

IBM ILOG Views Foundation Studio, a generic version of IBM ILOG Views Studio using dynamic libraries (DLL), is provided in the following directory:

```
%ILVHOME%\studio\<system>\<subsystem>
```

To run this editor, set the environment variable PATH to the directory:

```
%ILVHOME%\lib\<system>\<subsystem>
```

and

```
%ILVHOME%\studio\<system>\<subsystem>
```

where `<system>` indicates the name of the platform you are using (such as `x86_.net2005_8.0`) and `<subsystem>` indicates the name of the subplatform you are using (such as `dll_mda`).

Loading Plug-Ins

A plug-in is an IBM ILOG Views dynamic module that creates an IBM ILOG Views Studio extension.

Selecting Plug-Ins

When you use `ivfstudio` for IBM ILOG Views for the first time, `ivfstudio` prompts you to select the plug-ins it has detected.
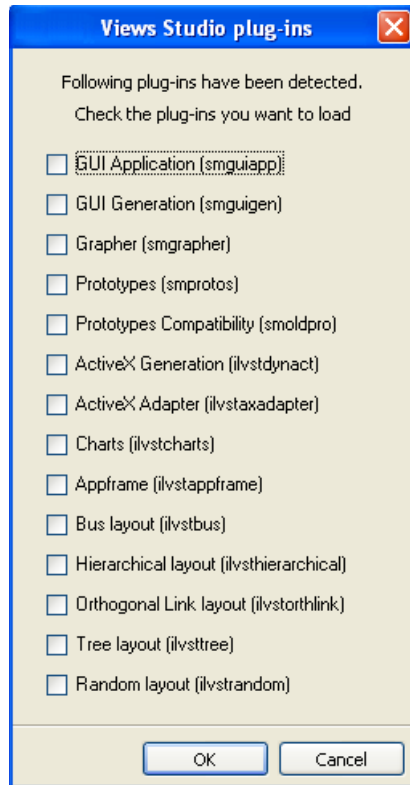
Figure 1.2   IBM ILOG Views Studio Plug-Ins Dialog Box (Microsoft® Windows® Systems)

Select the plug-ins you want to load by checking the corresponding toggle buttons. Then click OK. The plug-ins you use in this session will be loaded again the next time you launch `ivfstudio`.

If you click Cancel, `ivfstudio` starts without loading any plug-ins.

Using the -selectPlugIns Command Line Argument

If you want to select the plug-ins before `ivfstudio` starts running, you can launch `ivfstudio` with the `-selectPlugIns` argument. The IBM ILOG Views Studio Plug-Ins dialog box will appear and you can specify again the plug-ins you want to load.

Using the SelectPlugIns Command

Studio provides you with the `SelectPlugIns` command that you can use to select plug-ins. To use this command, choose SelectPlugIns from the Tools menu in the Main window. Studio displays the IBM ILOG Views Studio Plug-Ins dialog box for you to make your selections. You will then have to exit Studio and launch it again to load the plug-ins you just selected.

Specifying Plug-Ins Using the Environment Variables

You can also specify either a list of the modules you want to load or a directory containing all the modules you want to load. When you do this, the plug-ins loaded in the previous session are ignored.

Before initializing itself, Studio loads all the extension modules contained in a plug-in directory. The plug-in directory defaults to `%ILVHOME%/studio/plugins`. If you want to put your plug-ins in another directory, you can use the `ILVSTPLUGINSDIR` environment variable to indicate the directory. After loading the modules found in that plug-in directory, Studio looks for a subdirectory named `<system>`.

You can use the `ILVSTPLUGINS` environment variable to indicate a list of the modules you want to load. You can specify the full paths or the paths relative to the working directory, separated by a ";" character (a semicolon character).

Common Plug-Ins for IBM ILOG Views Studio

Some IBM ILOG Views packages come with their own plug-ins for IBM ILOG Views Studio. For more information, see the documentation of the IBM ILOG Views packages you use.

The following is a list of some common plug-ins located in the `%ILVHOME%\studio\<system>\<subsystem>` directory:

◆ If you have the IBM ILOG Views Gadgets package, you can use the IBM ILOG Views Studio GUI Application plug-in to edit GUI applications with gadgets panels. The corresponding dynamic package is `smguiapp.dll`. You can also use the IBM ILOG Views Studio GUI Generation plug-in to generate the C++ code for the GUI part of your application. The corresponding dynamic package is `smguigen.dll`.

◆ If you use the ILOG Views Grapher package, you can use the ILOG Studio Grapher plug-in to edit your graphers, nodes, and links. The corresponding dynamic package is `smgrapher.dll`.

◆ If you use the IBM ILOG Views Prototypes package, you can use the IBM ILOG Views Studio Prototypes plug-in to edit your business graphic objects. The corresponding dynamic package is `smproto.dll`.

Building Other Executables of IBM ILOG Views Studio

If for any reason you have to build an executable of IBM ILOG Views Studio, go to the `%ILVHOME%\studio\<system>\<subsystem>` directory and run the `make` utility passing the name of the executable you want to build as the only argument.

The provided `ivfstudio` can be used with the IBM ILOG Views Foundation product. However, you need the Gadgets and Manager packages to build any version of IBM ILOG Views Studio.

A Quick Look at the Interface

This section provides a quick look at the IBM ILOG Views Foundation Studio interface. You will see how the interface appears for Foundation Studio and learn about the various elements that make up the interface.

When you launch IBM ILOG Views Studio, the Main window appears. The Main window is where you will perform the tasks required to create the interface of your application.

The Main window includes the following elements:

◆ The work space containing the buffer windows in which you work. A Graphics window is displayed by default.

◆ The Palettes panel that lets you add graphic objects to the buffer windows.

◆ A menu bar that provides access to all the functions of the IBM ILOG Views Studio interface.

◆ An action bar underneath the menu bar that provides access to frequently used menu commands.

◆ An editing modes toolbar that provides access to the various modes that will define the behavior of your mouse when you are creating your graphic objects (selection, navigation, linking, and creating).

◆ The inspector area where you can edit some general properties of the graphic objects.

◆ The message area that displays system messages.

Note: If you are using Studio extensions provided with other packages of ILOG Views, the Studio interface may look different. For instance, the type of default buffers available in the work space depends on the extensions you are using. Also, Studio extensions may add items to menus and toolbars in the Main window and additional palettes in the Palettes panel.
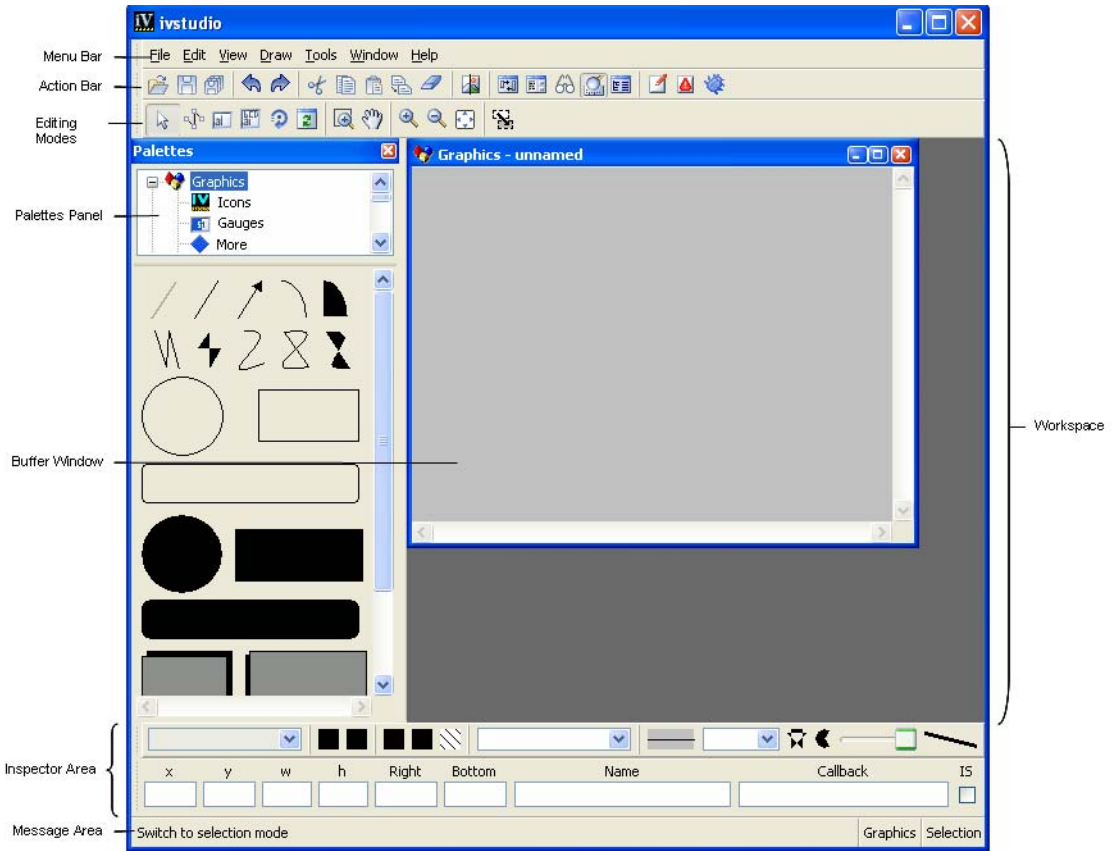
Figure 1.3 *Foundation Main Window at Start-up Time*

Buffer Windows

You will create your graphic objects in the buffer windows that are displayed within the
IBM ILOG Views Studio Main window. When using the IBM ILOG Views Foundation
Studio, you have access to only one type of buffer window, a 2D Graphics window. By
default, an empty buffer window is displayed when IBM ILOG Views Studio is launched.

The current buffer window type is shown at the bottom of the Main window in the message area.

Note: As you use IBM ILOG Views Studio extensions provided with other ILOG Views packages, you will have access to other types of buffer windows (Gadgets, Grapher, Application, and Prototypes). Each buffer type provides access to additional features of IBM ILOG Views Studio and has its own set of editing modes. In addition, the behavior of certain commands varies depending on the current buffer. For example, the Test command tests just the panel if you are editing a Gadgets buffer, but it tests all the panel instances in an application if you are editing an Application buffer.

Palettes Panel

You will use the Palettes panel in conjunction with the buffer windows to create your graphic objects. The palettes available through this panel provide predefined graphic objects to make your drawing tasks easier. You can use the drag-and-drop feature or the creation mode to add the various graphic objects to your buffer windows.

The Palettes panel opens automatically when IBM ILOG Views Studio is launched. It is divided into two panes. The upper pane displays a tree gadget with various items, each corresponding to a particular graphic palette. The lower pane displays the objects contained in the palette selected in the tree. To access the palette, you are interested in, click the appropriate item in the tree. For example, to display gauge-related items, click Gauges in the tree in the upper pane of the Palettes panel.

Click on a tree item to display the corresponding palette

Gauges Palette
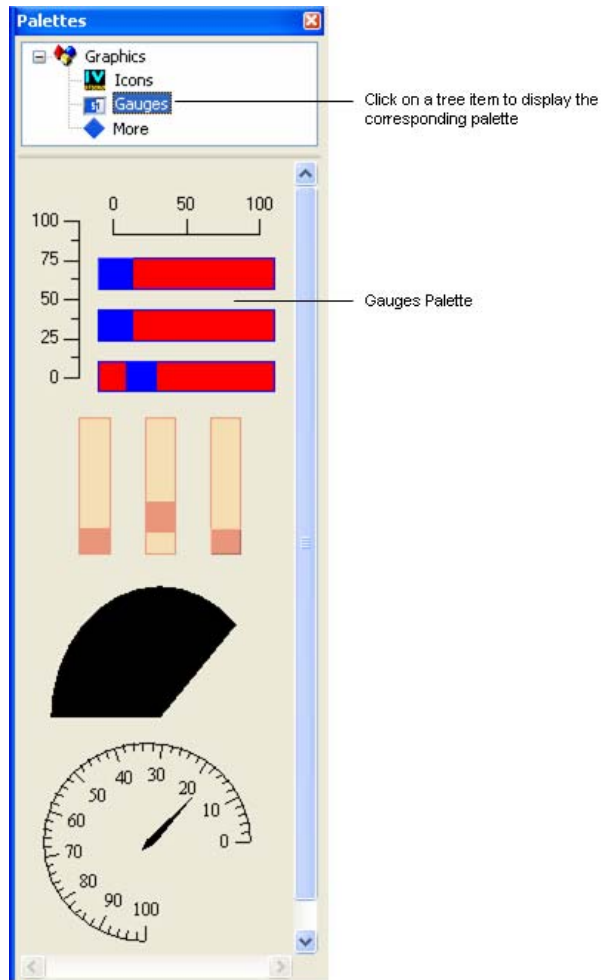
Figure 1.4 *The Palettes Panel*

2

Using ILOG Studio

This chapter tells you how to perform basic tasks in ILOG Studio. You can find information on the following topics:

◆ *Creating Objects*

◆ *Selecting Objects*

◆ *Specifying Resources for Objects*

◆ *Inspecting an Object*

◆ *Aligning Objects*

◆ *Using the Editing Modes of ILOG Studio*

◆ *Using Marking Menus*

◆ *Working with Description Files*

◆ *Printing*

Creating Objects

The Palettes panel provides various predefined graphic objects from which you will create your own objects. ILOG Studio gives you two ways to create these objects in the buffer window. You can use either a drag-and-drop operation or the creation mode feature.

Using the Drag-and-Drop Operation

When you use the drag-and-drop operation for creating your objects, the object that is added to the buffer window is an exact copy of the object as it is found in the Palettes panel. The object has the same shape and dimensions of the object in the Palettes panel.

To create an object using the drag-and-drop operation:

1. In the upper pane of the Palettes panel, click the item in the tree corresponding to the type of object you want to create.

   The related palette appears in the lower pane.

2. Click the object you are interested in and drag it to the Graphics buffer window.

   When you release the mouse button, you are in Selection mode. The object remains selected in the buffer window and you can modify it as you want.

Example - Drag and Drop

For example, to create a simple vertical rectangular scale:

1. In the upper pane of Palettes panel, click Gauges in the tree.

2. In the lower section of the Palettes panel, click the vertical `IlvRectangularScale` object.

3. Drag it to the Graphics buffer window.
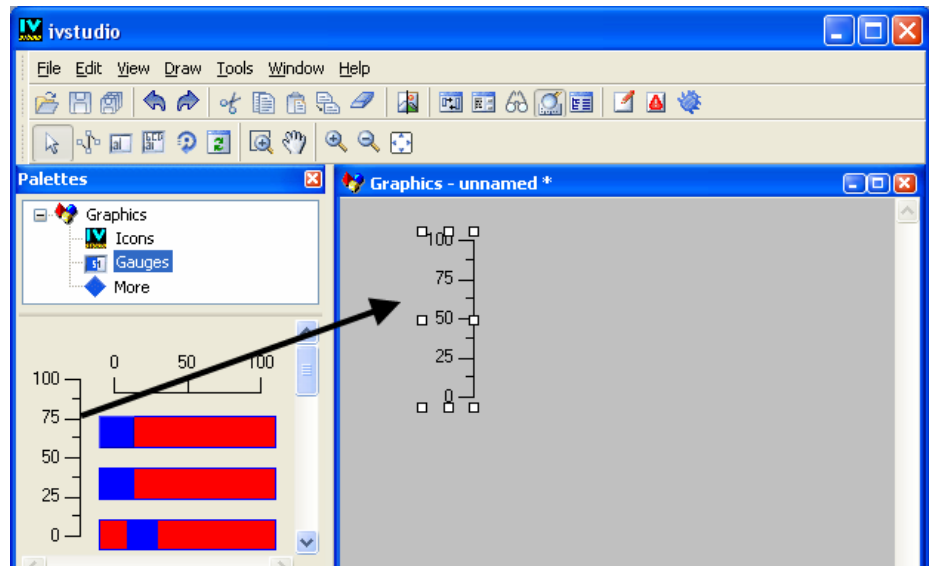
Using the Creation Mode

When you use the creation mode, you are essentially drawing the object in the buffer window. You determine for yourself the size and shape of the object. Creation mode also allows you to create multiple objects once you have selected the kind of object you want to create in the Palettes panel.

To create an object using the creation mode:

1. In the upper pane of the Palettes panel, click the item in the tree corresponding to the kind of object you want to create.

The related palette appears in the lower pane.

2. In the lower pane of the Palettes panel, click the object you are interested in. A bounding box appears around the object to indicate that creation mode is active.

If you want to add only one object to the buffer window, click the object in the Palettes window once. (This puts you in transient creation mode. After you have drawn the object in the buffer window, you will leave creation mode automatically.)

If you want to add multiple objects of the same kind, hold down the Shift key and click the object in the Palettes panel. (This puts you in permanent creation mode. You will remain in creation mode and you can draw as many objects as you like. To leave creation mode, you must click the Selection mode icon in the Editing Modes toolbar.)

3. Move the pointer to the buffer window.

4. Click in the buffer window where you want your object positioned and drag the mouse until the object is the size and shape you want.

Example - Creating a Single Object

For example, to create a single ellipse:

1. In the upper pane of Palettes panel, click Graphics in the tree.

2. In the lower pane of the Palettes panel, click the `IlvEllipse` icon once. Notice the bounding box that appears around the `IlvEllipse` icon indicating you are in creation mode.

3. Click in the Graphics window at the position where you want to start drawing the ellipse.

4. Drag the mouse until the ellipse is the size and shape you want it to be. As you drag the mouse, you see a bounding box that shows the shape and size of your object.

5. Release the mouse button. The ellipse appears with dimensions of the bounding box you have just drawn.
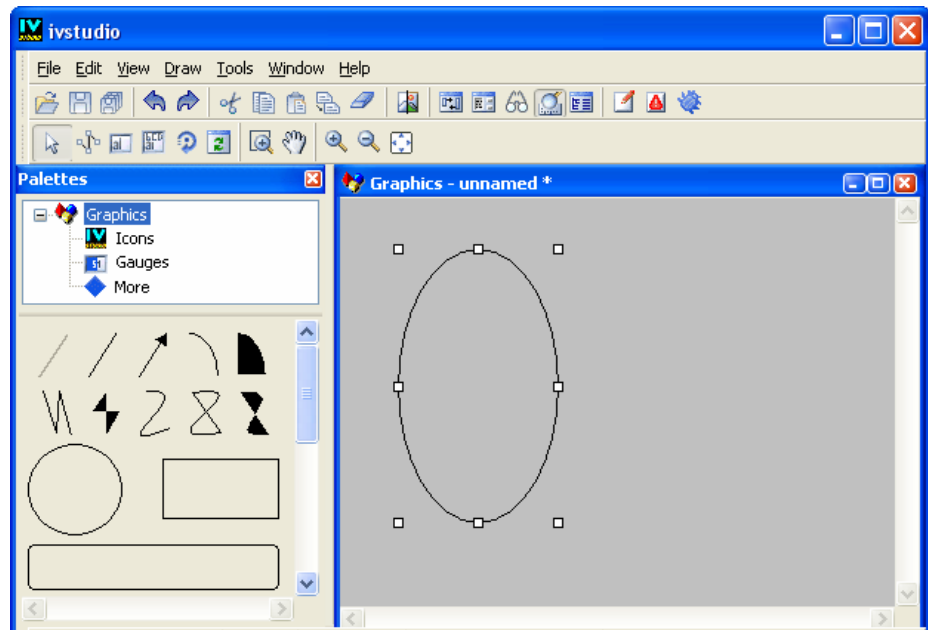
When you release the mouse button, you automatically leave creation mode and are put into Selection mode. Notice that the `IlvEllipse` icon in the Palettes panel is no longer selected and that the Selection mode icon ⬚ is selected in the Editing Modes toolbar.

You can reshape, resize, move, or modify the ellipse as you want.

Example - Creating Multiple Drawings

The following is an example of adding multiple drawings of the same kind of object using permanent creation mode. To create three filled rectangles:

1. In the upper pane of Palettes panel, click Graphics in the tree.

2. In the lower section of the Palettes panel, *Hold down the Shift key and click the* `IlvFilledRectangle` *icon*. Notice the bounding box that appears around the icon indicating you are in creation mode.

3. Click in the Graphics window at the position where you want to start drawing the rectangle.

4. Drag the mouse until the rectangle has the dimensions you want it to have. As you drag the mouse, you see a bounding box that shows the shape and size of your object.
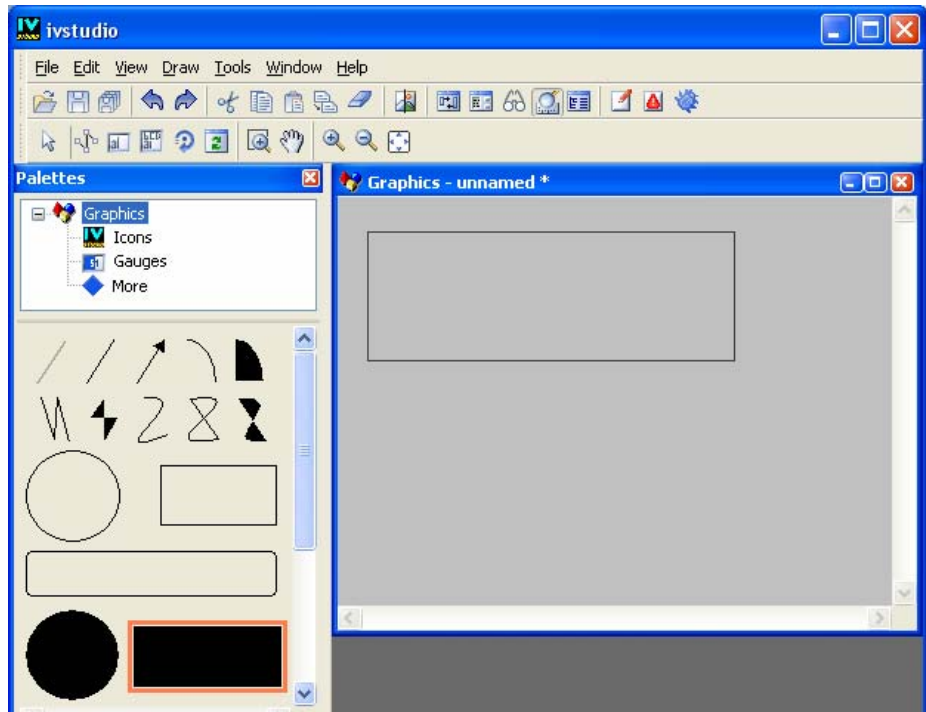
5. Release the mouse button. The filled rectangle appears with dimensions of the bounding box you have just drawn.
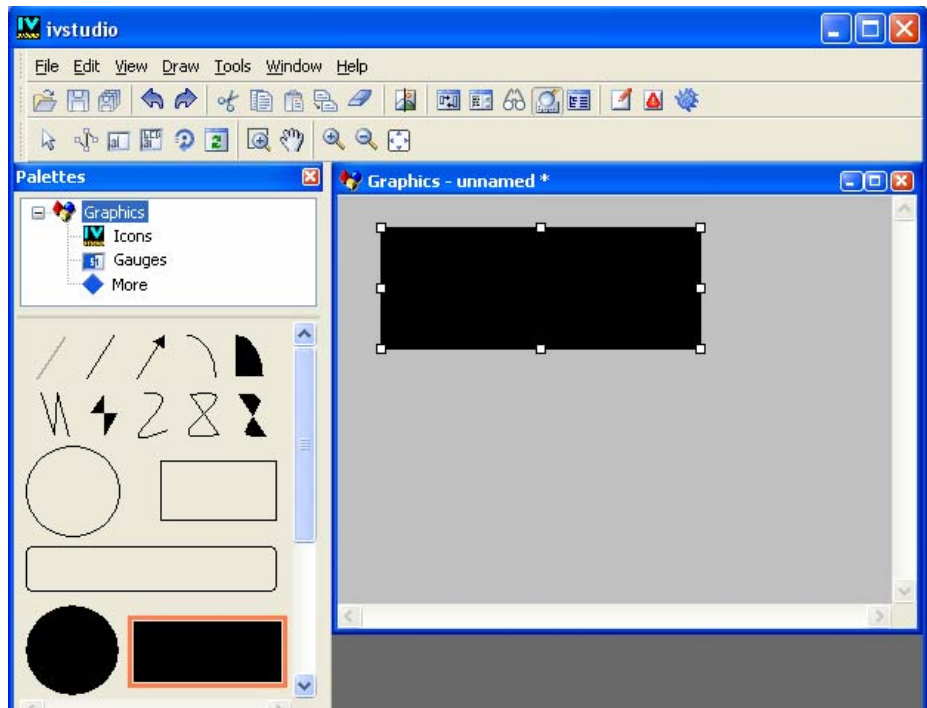
Notice that the `IlvFilledRectangle` icon remains selected in the Palettes panel indicating that you are still in creation mode. You can continue drawing the other two rectangles in the buffer window.

6. Click in the buffer window where you want to begin drawing the second rectangle. Once again, drag the mouse until the bounding box is the desired size.
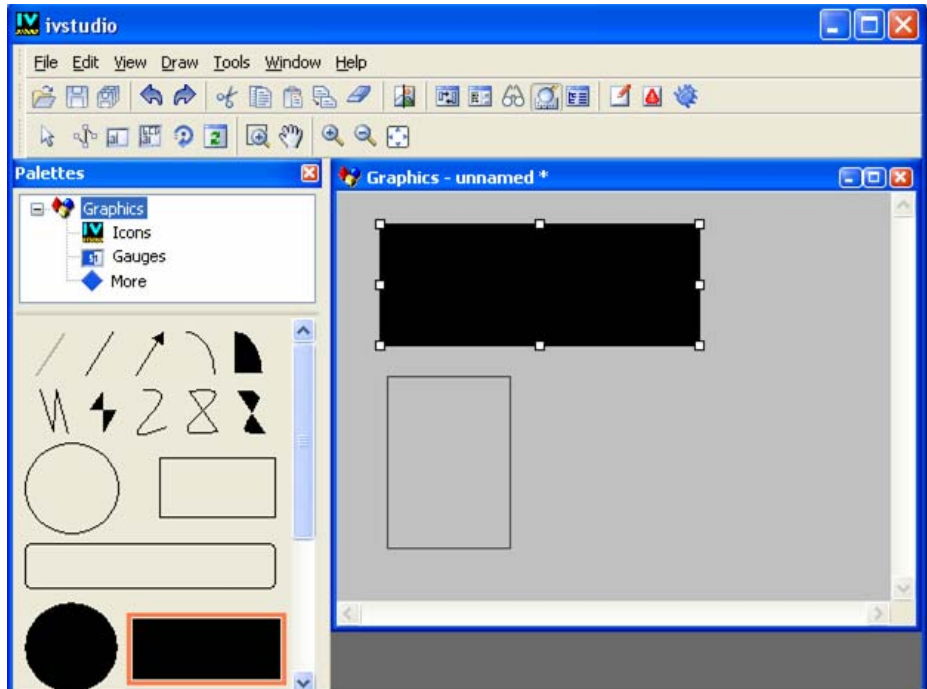
7. Release the mouse button and then draw the third rectangle. As long as you are in creation mode, you can draw as many objects as you like.

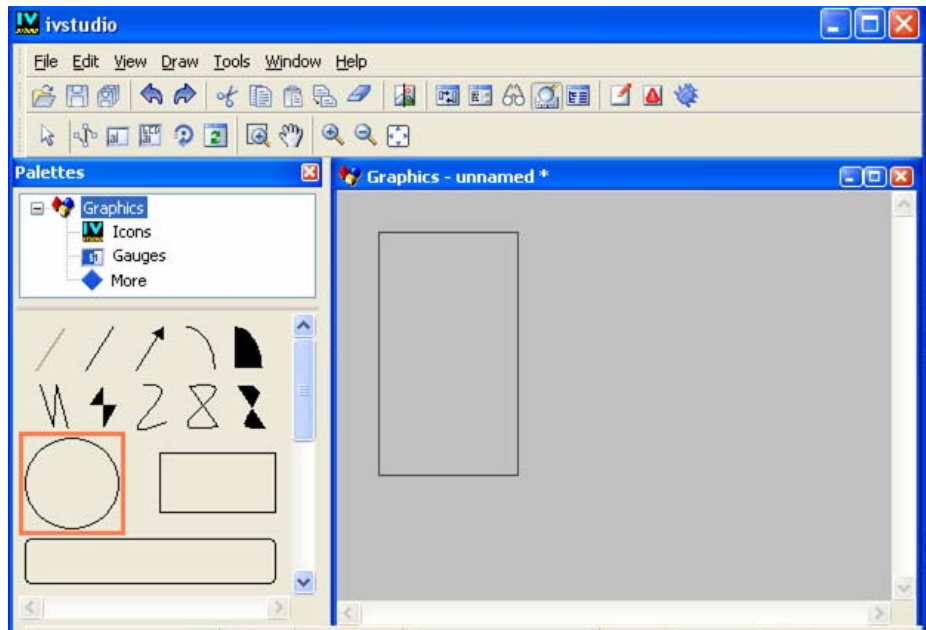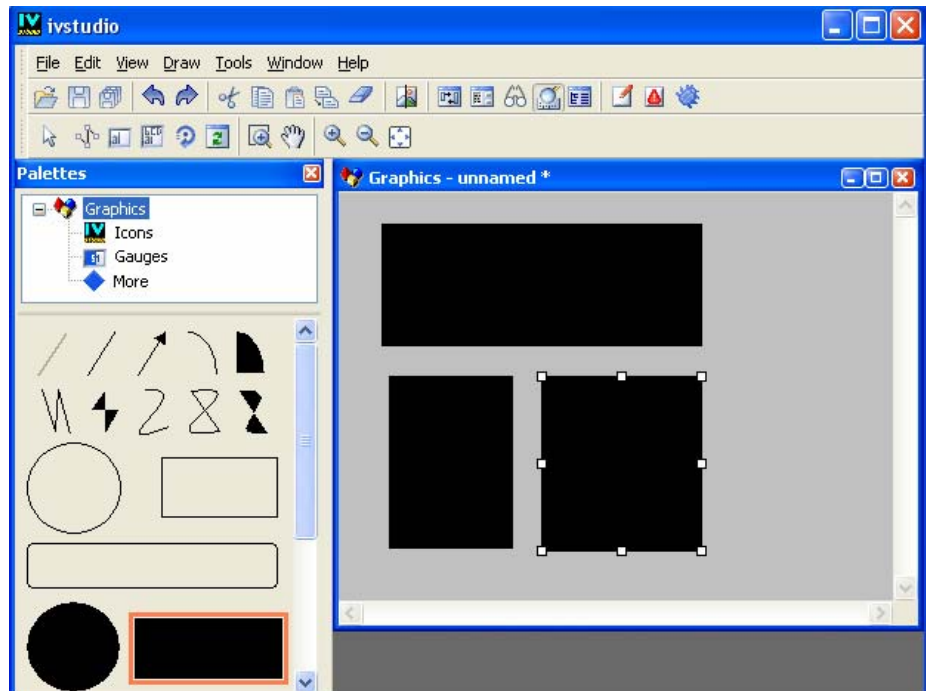**8.** To leave creation mode, click the Selection mode icon ⟨⟩ in the Editing Modes toolbar.

Notice that the `IlvFilledRectangle` icon in the Palettes panel is no longer selected. Now that Selection mode is active, you can select an object and reshape, resize, move, or modify it as you want.

Note: Some of the other types of objects in the Palettes panel have customized creation modes that better fit the needs for drawing their shape. For example, polyline, arc, and spline modes have additional steps to create the object. See Graphics Palette on page 83 for details on creating these objects.

Selecting Objects

The Selection mode lets you select, move, and resize objects and do other common editing operations. This mode is activated by default when ILOG Studio is launched.

To activate Selection mode, click the arrow icon in the Editing Modes toolbar. To select an object for editing, click the object in the work space. To select more than one object, hold down the Shift key when clicking.

Selection Mode Icon
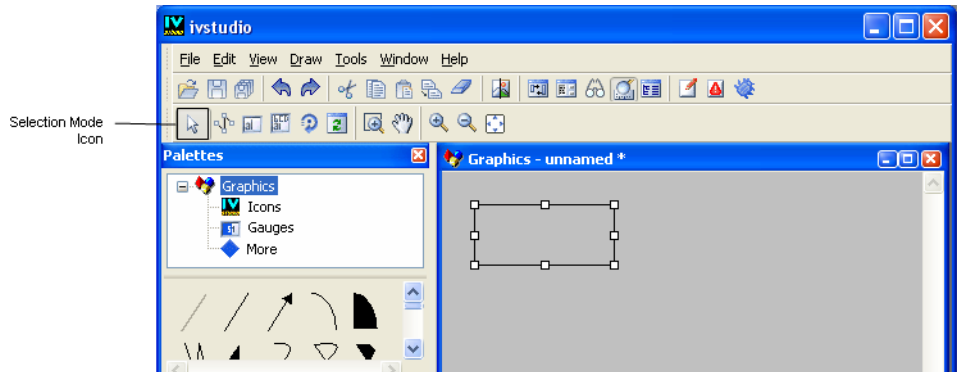
Figure 2.1 *Selection Mode Icon*

When an object is added to the work space, it is selected. When the Selection mode is active, you can move or resize the selected objects contained in the work space. You can select and move more than one object simultaneously by pressing down the Shift key when clicking. Only one object, however, can be resized at a time.

Editing the Basic Properties of an Object

When only one object is selected, its basic properties are displayed in the generic inspector at the bottom of the Main window. You can use the generic inspector to edit the basic properties of the selected graphic object. These properties can be applied to any graphic object class.
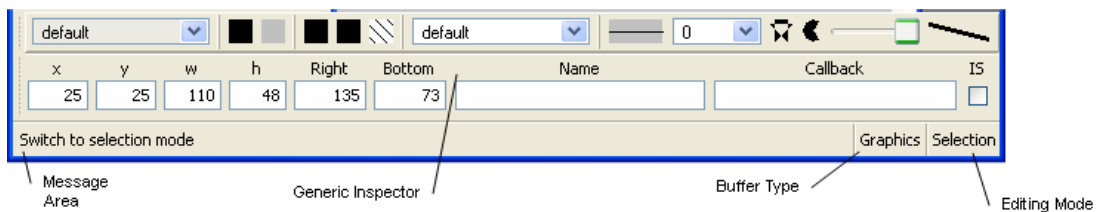
Figure 2.2 *Generic Inspector and Message Area*

The message area displays the class name of the selected object and any ILOG Studio messages. The area to the right of the message area displays the type of the current buffer window (here Graphics) and the current editing mode (here Selection).

For details about the fields contained in the generic inspector, see *The Generic Inspector* on page 78.

Specifying Resources for Objects

The resource editor lets you apply various graphic resources to the objects selected in the work space. The resource editor is displayed by default when ILOG Studio is launched.
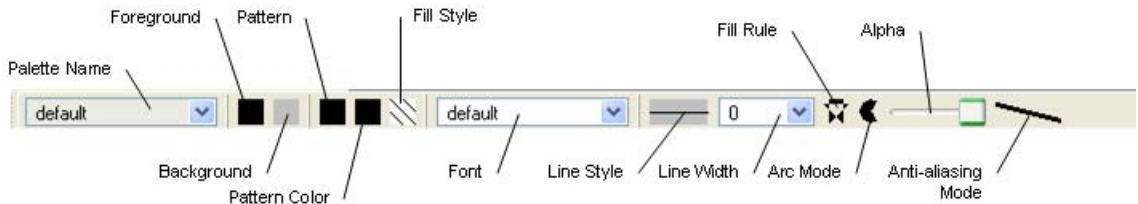
Figure 2.3 *Resource Editor in the Main Window*

When an object is selected, the resource editor is automatically updated and displays the current resources of the selected object. When more than one object is selected, the panel keeps the value of the first selected object, but any change affects all the selected objects.

For details about the fields contained in the resource editor, see *The Resource Editor* on page 79.

Inspecting an Object

To inspect the specific properties of an object, double-click that object. You can also click the Inspect icon in the Action toolbar of the Main window.

Figure 2.4 *Inspect Icon in the Action Toolbar*

If the class of the object has an associated inspector panel, you can use it to edit the specific properties of the object class. The contents of the inspector depend on the related object class. Each inspector contains a General page and a Callbacks page, plus additional notebook pages as needed depending on the class.

For example, you double-click an `IlvRectangle` object, the following inspector panel appears:
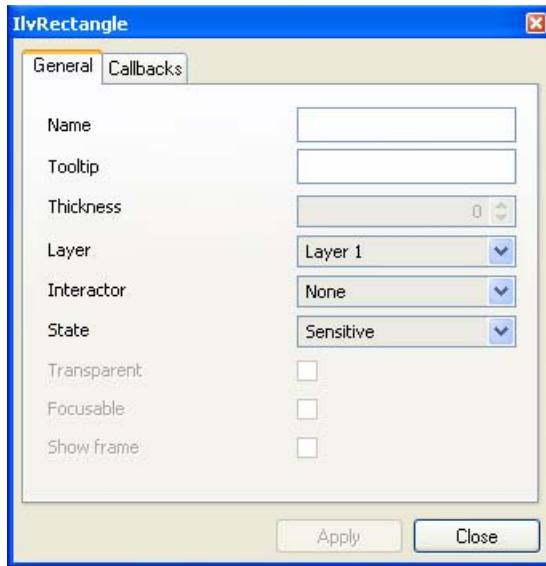
Figure 2.5 *Rectangle Inspector Panel (General Page)*

If you double-click an `IlvShadowRectangle` object, the following inspector panel appears:
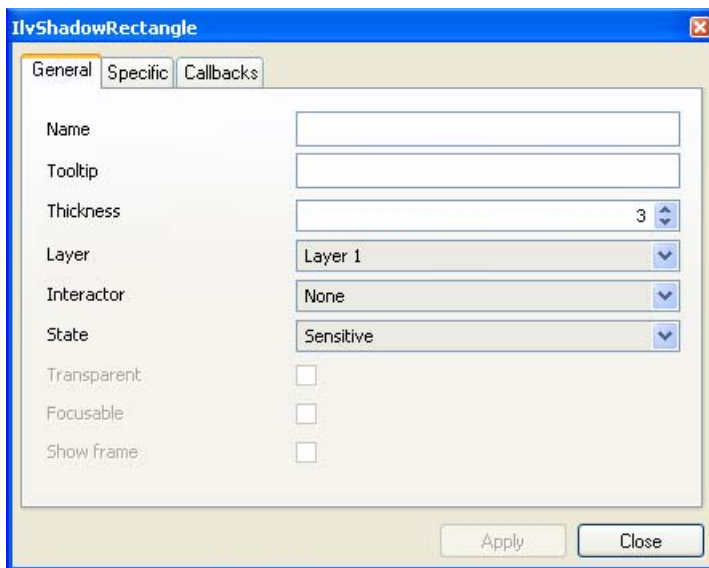
Figure 2.6 *Shadow Rectangle Inspector Panel (General Page)*

To validate the changes made to the object properties, click Apply. To close the panel, click Close.

Only one object can be inspected at a time. If you select another object of the same type while the first is being inspected, the properties of the newly selected object appear in the inspector panel. If another type of object is selected, its associated inspector replaces the one that is displayed.

Aligning Objects

You can align objects in your buffer window using the Align/Distribute from the Draw menu. The alignment operations apply to objects that are already selected. When at least two objects are selected, the first selected object is used as the reference for the other objects. If only one object is selected, it is aligned with either the whole panel or with any guides to which the object may be attached.
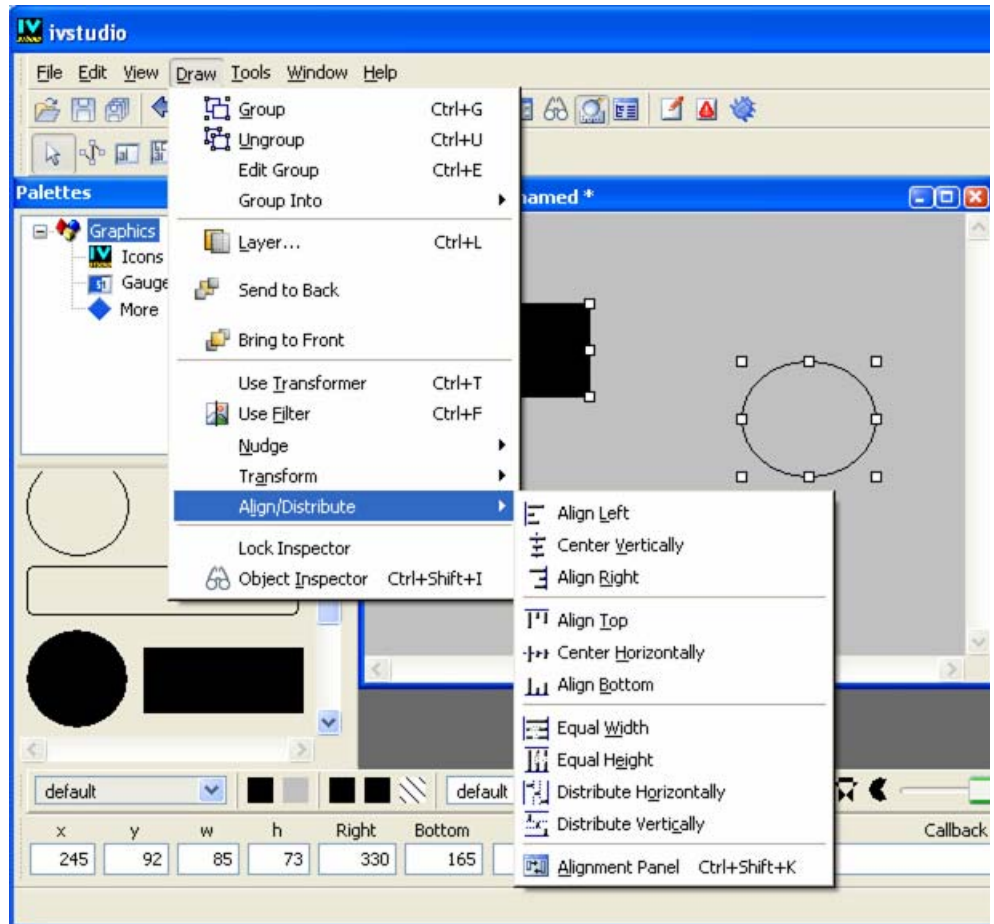
Figure 2.7   Align/Distribute Submenu of the Draw Menu

Use the following selections from the Align/Distribute submenu to align objects in your buffer windows:

Align Left Moves the selected objects horizontally so their left borders are vertically aligned. Object dimensions are not changed. If only one object is selected, it is moved horizontally to the left panel border (or to its left guide).

Center Vertically Moves selected objects horizontally so their centers are vertically aligned. Object dimensions are not changed. If only one object is selected, it is moved horizontally so its center is vertically aligned with the center of the panel (or between guides).

Align Right Moves the selected objects horizontally so their right borders are vertically aligned. Object dimensions are not changed. If only one object is selected, it is moved horizontally to the right panel border (or to its right guide).

Align Top Moves the selected objects vertically so their top borders are horizontally aligned. Object dimensions are not changed. If only one object is selected, it is moved vertically to the top panel border (or to its guide).

Center Horizontally Moves selected objects vertically so their centers are horizontally aligned. Object dimensions are not changed. If only one object is selected, it is moved vertically so its center is horizontally aligned with the center of the panel (or between guides).

Align Bottom Moves the selected objects vertically so their bottom borders are horizontally aligned. Object dimensions are not changed. If only one object is selected, it is moved vertically to the bottom panel border (or to its guide).

Equal Width Resizes the selected objects so they have the same width. Objects are not moved. If only one object is selected, it is moved and resized so it has the same width as the panel (or the distance between the guides to which it may be attached).

Equal Height Resizes the selected objects so they have the same height. Objects are not moved. If only one object is selected, it is moved and resized so it has the same height as the panel (or the distance between the guides to which it may be attached).

Distribute Horizontally Creates an equal amount of horizontal space between the selected objects. The space between two objects is at least equal to a predefined value. Object dimensions are not changed. You must select at least two objects.

Distribute Vertically Creates an equal amount of vertical space between the selected objects. The space between two objects is at least equal to a predefined value. Object dimensions are not changed. You must select at least two objects.

Using the Alignment Panel

The Alignment panel can also be used to align the objects in your buffer windows. It provides access to the same alignment commands as are found in the Align/Distribute submenu.

Select ▦ Alignment Panel from Align/Distribute of the Draw menu to display the panel.
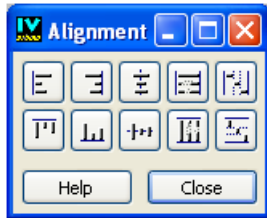
Figure 2.8   *The Alignment Panel*

Using the Editing Modes of ILOG Studio

Once you have created an object in the buffer window, you can use the tools available in the editing modes toolbar to change or modify the object.

Figure 2.9   *The Editing Modes Toolbar*

When you click an icon in the toolbar, you will enter the editing mode and will be able to perform the following tasks:

◆ Select an object for editing.

◆ Reshape or resize a polyline or polygon object.

◆ Add or edit a single-line label or multiple-line text to the buffer window.

◆ Rotate an object in the buffer window.

◆ Test the behavior of your objects and edit some of their properties.

◆ Zoom in on a selected portion of an object.

◆ Pan an object in the buffer window.

◆ Zoom in or zoom out a selected object in the buffer window.

◆ Fit all the contents of the buffer within the window.

For details about each editing mode, see *The Editing Modes Toolbar* on page 75.

Note: As you use ILOG Studio extensions provided with other ILOG Views packages, you will have access to other types of buffer windows (Gadgets, Grapher, and Application). Each buffer type provides access to additional features of ILOG Studio and has its own set of editing modes. The editing modes vary depending on the type of the current buffer window.

Using Marking Menus

The marking menu feature of ILOG Studio provides fast access to certain items of the Main window menu bar.

You trigger the marking menu feature by pressing the right mouse button in the buffer window. The marking menu then appears showing the menu items in a circle around the current location of the pointer. To choose a menu item, you move up, down, left, right, or diagonally to the appropriate region of the menu and release the mouse button.

Figure 2.10 shows the marking menu of ILOG Studio. The Views, Tools, Inspect, Edit, Layers, and Draw menu items are arranged around the center circle.
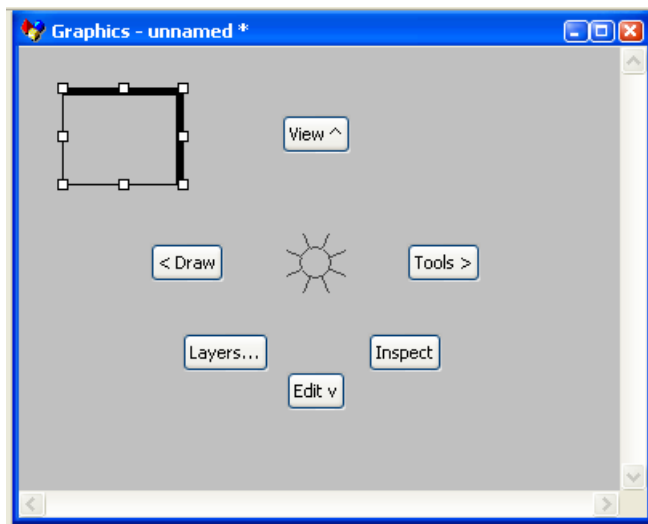
Figure 2.10   ILOG Studio Marking Menu

To activate one of the menu items (for example, Inspect), press the right mouse button, move the mouse quickly in the direction of the Inspect item, and release the button. When you do this, the inspector panel for the object selected in the buffer window is displayed.

Some of the marking menu items contain submenus. For example, when you move the mouse over Edit (Figure 2.11), a submenu appears with additional items (Figure 2.12). You could then choose Paste, for example, by moving the mouse in that direction. You can go back to the previous menu by moving the mouse back to the center of the previous menu. In Figure 2.12, the previous menu center is the circle just above Redo in the Edit submenu.
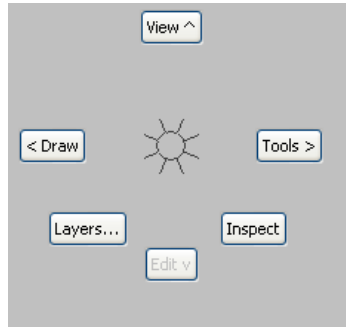
Figure 2.11   *The Edit Menu Selected in the Marking Menu*
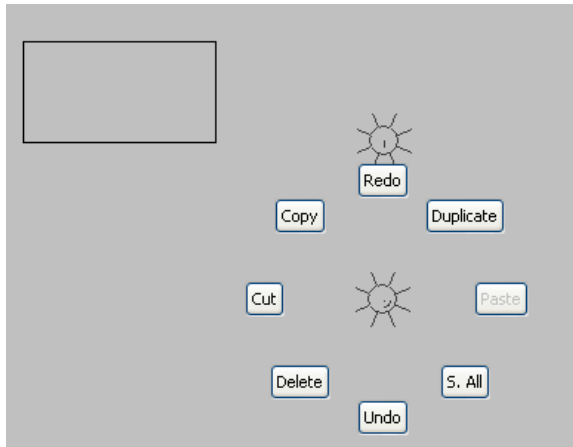
Figure 2.12   *The Paste Selection from the Edit Submenu*

You can also trigger the items in a submenu by using the "fast" method. This consists of moving the mouse quickly in the directions needed to trigger the item without having to wait for the menu to appear. For example, the Edit/Paste menu selection can be made by making an L-shaped mark in the buffer window while holding down the right mouse button. Move the mouse down and then to right (Figure 2.13). The term "marking menu" comes from this capability to select submenu items as if you are tracing marks with a pencil.
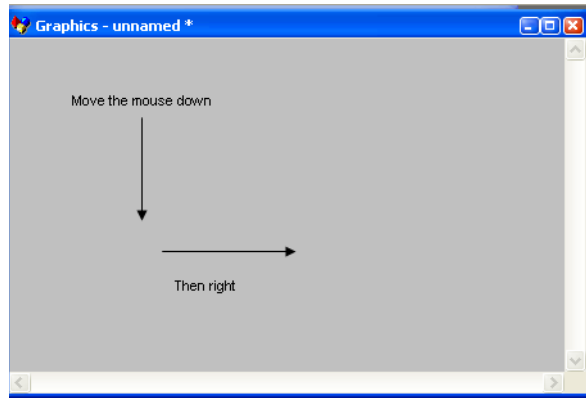
Figure 2.13   *The "Marking" Action of Marking Menus*

Note: If you practice using the Marking Menus feature for a short period of time, you will find that you can work in a much more fluid and fast way. Therefore, we encourage you to consider practicing with the marking menus to get the benefits of the improved workflow.

ILOG Studio Default Marking Menu

The ILOG Studio default marking menu consists of the Views, Tools, Inspect, Edit, Layers, and Draw menu items are arranged around the center circle. These items contain the following selections:

View Menu

Menu Item	Description	Command
Grid Snap	Snaps movements of object to the grid.	ToggleActiveGrid
Zoom Out	Zooms out the current view.	ZoomOut
Fit	Resizes the buffer window so that all the contents of the buffer fit within the window.	FitViewToContents
Grid	Displays a grid in the current buffer window.	ToggleGrid

Menu Item	Description	Command
Reset	Returns the object to its original size.	ResetView
Zoom In	Zooms in the current view.	ZoomIn

Tools Menu

Menu Item	Description	Command
Points	Activates PolySelection mode.	SelectPolySelectionMode
Select	Activates Selection mode.	SelectSelectionMode
Text	Activates Label List mode.	SelectLabelListMode
Last	Returns to the last creation mode used in the buffer window.	
Active	Activates Active mode.	SelectActiveMode
Rotate	Activates Rotate mode.	SelectRotateMode

Inspect Menu Item

Menu Item	Description	Command
Inspect	Displays the Inspector panel for the object selected in the work space.	

Edit Menu

Menu Item	Description	Command
Redo	Repeats an editing action that was previously undone.	Redo
Duplicate	Duplicates the selected objects.	Duplicate
Paste	Copies the objects in the clipboard to the buffer window.	ClipboardPaste

Menu Item	Description	Command
S. All	Selects all objects in the buffer window.	SelectAll
Undo	Undoes the previous action.	UnDo
Delete	Deletes the selected objects.	Delete
Cut	Cuts the selected objects to the clipboard.	ClipboardCut
Copy	Copies the selected objects to the clipboard.	ClipboardCopy

Layers Menu Item

Menu Item	Description	Command
Layers	Displays the Layers Inspector panel.	ShowLayerPanel

Draw Menu

Menu Item	Description	Command
Front	Brings the selected object to the front of the work space.	Pop
Layer ...	Lets you change the layer number of the selected object.	SetLayer
Ungroup	Ungroups the selected object.	UnGroup
Use T.	Places the selected object in transformer mode.	ToggleTransformed
Back	Sends the selected objects to the back of the work space.	Push
Group	Groups the selected objects.	Group

Working with Description Files

The contents of each buffer window (that is, the description of your panel) can be saved in a data file that typically has an `.ilv` extension. This data file can then be loaded and edited in ILOG Studio or used by a container or a manager in an application. See the description of the container and manager in the *ILOG Views User's Manual* for more information.

The following commands in the menu bar of the Main window let you manipulate data files:

Saving Files

To save the buffer for the first time, choose Save As… from the File menu. This command opens a dialog box where you can specify the name of the file and its directory.

If the current buffer has already been saved with the Save as… command or has been loaded from the disk, choose Save from the File menu or click the Save icon from the toolbar to save it.

Opening Files

To load a file previously saved by ILOG Studio, choose Open… from the File menu or click the Open icon from the toolbar at the top of the Main window. This command opens a file selector that lets you choose a file. The selected file is loaded in a new buffer, which becomes the current buffer.
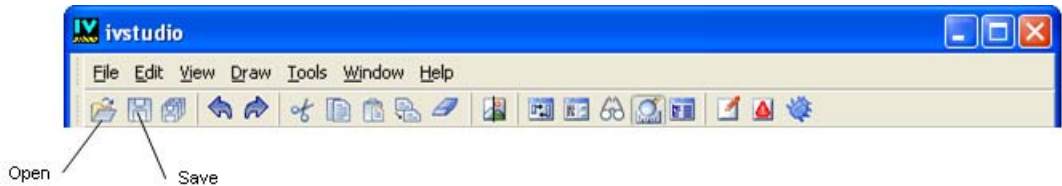
Figure 2.14   Open and Save Icons

Switching between Open Buffer Windows

You can switch between the buffers that are currently loaded using the Window menu. The name of the current buffer is displayed just after the menu separator (in Figure 2.15, it is unnamed), followed by the names of the other opened buffers.

Although you can open several buffers simultaneously, you can only edit one buffer at a time. You can, however, switch from one buffer window to another without having to save the file. To do so, select the buffer window you want to display from the Window menu.
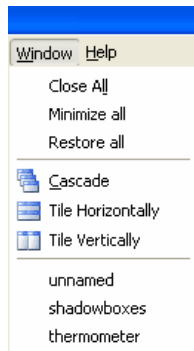
Figure 2.15 *The Window Menu with Multiple Buffers*

You can edit different types of buffer windows. In the Foundation Studio, you have access to only 2D Graphics buffer windows. However, as you use the ILOG Studio extensions provided with other ILOG Views packages, you will have access to other types of buffer windows (Gadgets, Graphics, and Application windows). Each buffer type provides access to additional features of ILOG Studio and has its own editing modes.

Printing

To do a printing you can select one of the following items from the File menu:

◆ Print the current buffer

◆ Preview the current buffer

◆ Print...

◆ Preview...

The first two items, Print the current buffer and Preview the current buffer, allow you to print or preview the current buffer. The layout of the current buffer is defined by means of the `IlvPrintableLayoutIdentity` class. The header associated with this layout contains the *file name*, while the footer contains *the index of the page/the total number of pages*. You cannot modify this layout.

The last two items, Print and Preview, allow you to print or preview a list of buffers or texts. You can also choose the layout, and several other properties.

How to Edit a Document

When you select Print... or Preview... from the File menu , you obtain a dialog box that contains a hierarchical tree and four buttons, as shown in the following figure:
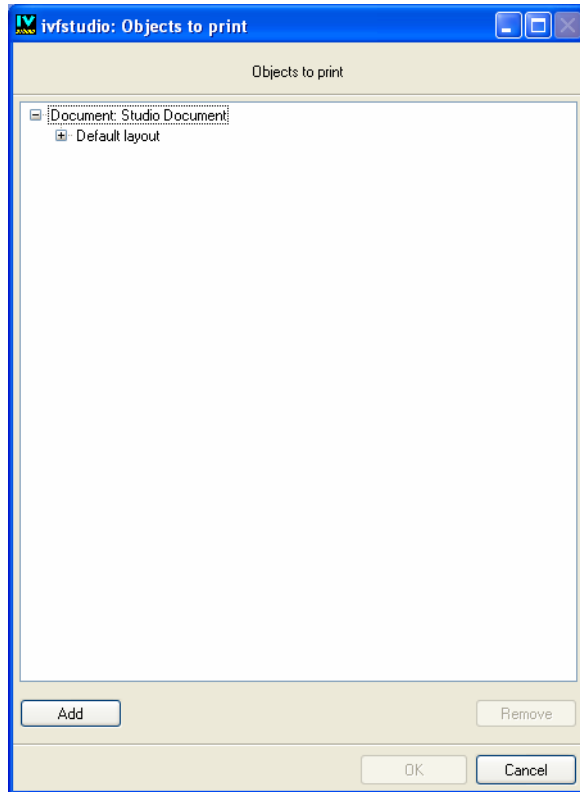
Figure 2.16   Objects to Print

◆ The Add button allows you to select a new chapter (that is, a new object to print which may be on several pages.)

◆ The Remove button allows you to remove the selected chapter.

◆ The OK button allows you to preview or print your document.

◆ The Cancel button allows you to cancel without printing or previewing.

Note: To make active the Remove and OK buttons, you must select at least one chapter.

Note: ivfstudio *keeps the choices you made for this document, even if you select Cancel, as long as you do not exit* ivfstudio.

The Main Dialog

The hierarchical tree (see Figure 2.16) shows you the structure of your document. You can edit the properties by right-clicking them.

The hierarchical tree is composed of the following items:

◆ The root item, that is, the name of your document. You can edit the name by right-clicking it. This item always exists.

◆ The default layout used when you do not specify a layout for a chapter. You can change it by right-clicking it. This item always exists.

◆ One item for each chapter to print. These items may be added or removed interactively by using the Add and Remove buttons.

Adding a New Chapter

By clicking the Add button, the following dialog box appears:
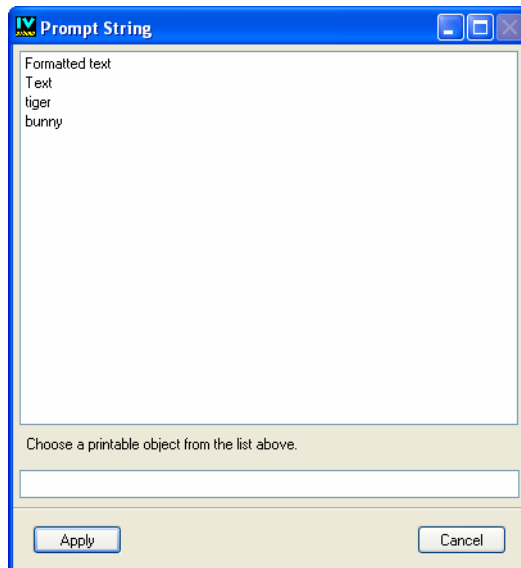
The first two items that appear in the dialog box are pure text. See the `IlvPrintableText` class and the `IlvPrintableFormattedText` class in the Reference Manual.

You can choose a printable object associated with the new chapter. You may print all the `.ilv` files you loaded in Studio and the new buffers you are editing.

Removing a Chapter

To remove a chapter, select the chapter in the hierarchical tree and click the Remove button.

Modifying the Properties of a Printable Object

You can modify most of the properties of a printable object, for example the properties associated with a chapter, or those associated with the background, the header, the footer, or the foreground of a layout. Some properties are common to all the printable objects, and some others are specific. If you want to see all the properties of a printable object, you may need to expand the relevant item.

Common Properties of Printable Objects

The common properties of a printable object are the following:

◆ type

   Identifies the type of the printable object. It can be *ivfstudio buffer*, *Text*, or *Formatted text*. This item is not editable.

◆ name

   It is the name of the printable object. This item is editable. If you want to edit this item, right-click it and enter the new name in the Prompt String dialog box.

Specific Properties of the Text and Formatted Text Printable Objects
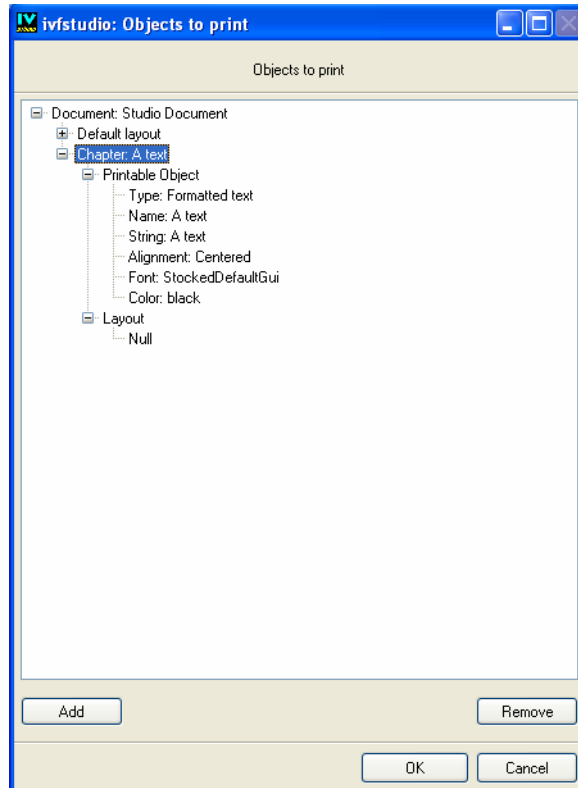
All the specific properties are editable.

The specific properties are the following:

◆ String

The text to be printed. Right-click this item if you want to edit it, and enter the new text in the Prompt String dialog box.

◆ Alignment

Right-click this item if you want to choose a new alignment, by using the following dialog box:
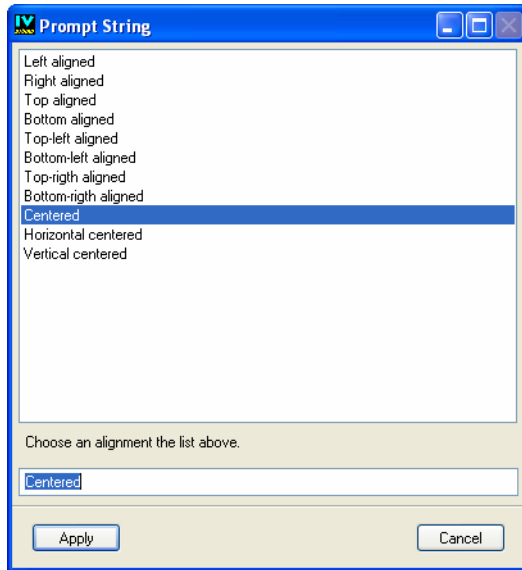
◆ font

Right-click this item if you want to choose a new font, by using the Font Chooser dialog box.

◆ color

Right-click this item if you want to choose a new color, by means of the Color Chooser dialog box.

Specific Properties of the ivfstudio buffer Printable Object
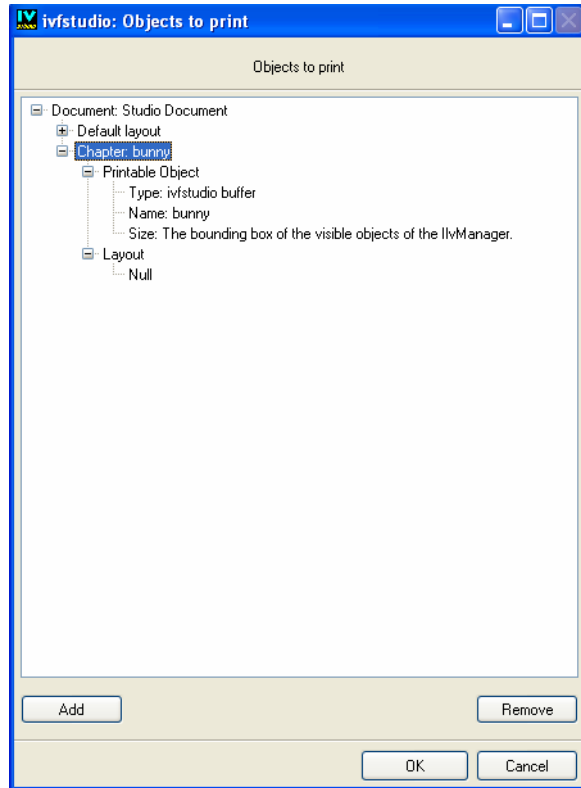
◆ size

This specific property identifies the size of the area to be printed. Right-click this item, if you want to edit it.

The following dialog box appears:
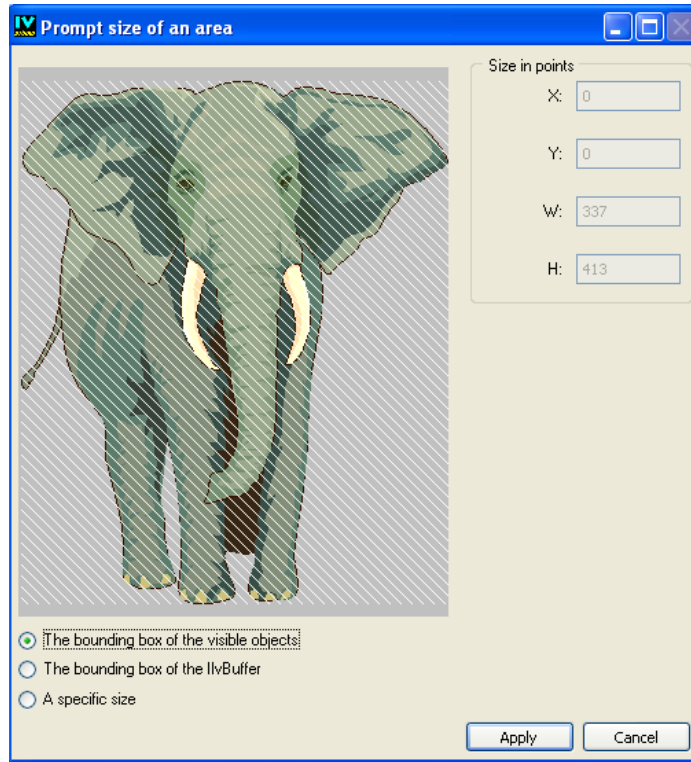
You have three choices:

◆ The bounding box of the visible objects in the buffer.

◆ The bounding box of the view of the buffer.

◆ A size you may specify either by number fields, or by an interactor (which is an instance
   of the `IlvMoveReshapeInteractor` class) in the small view of the buffer inserted in
   the dialog.

Replacing the Default Layout

To replace the default layout, right-click the item *Default layout*. You will get a dialog box
that allows you to choose the layout to apply as default.
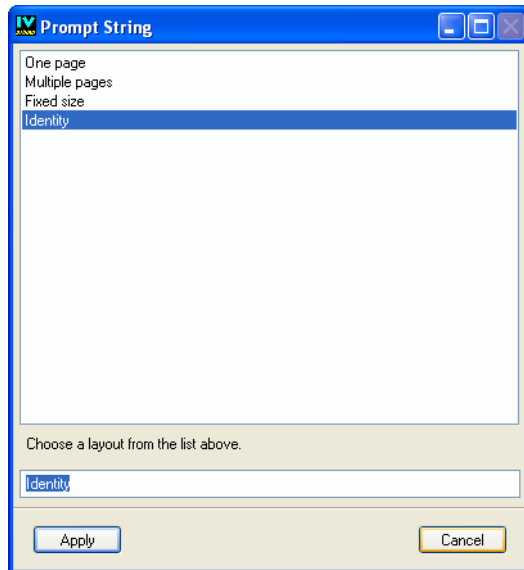
You can select one of the following layouts:

◆ Identity

   The chapter is printed roughly as it would appear on the screen.
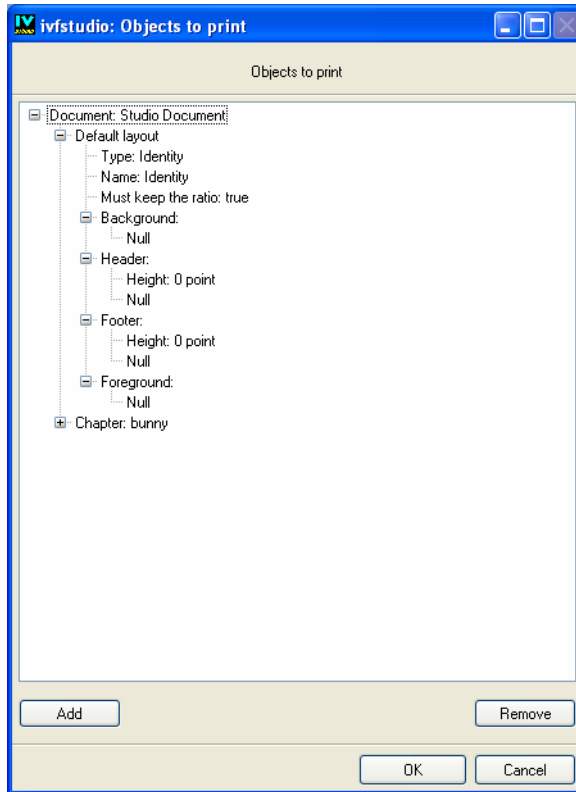
◆ One page

The chapter is printed on a single page.
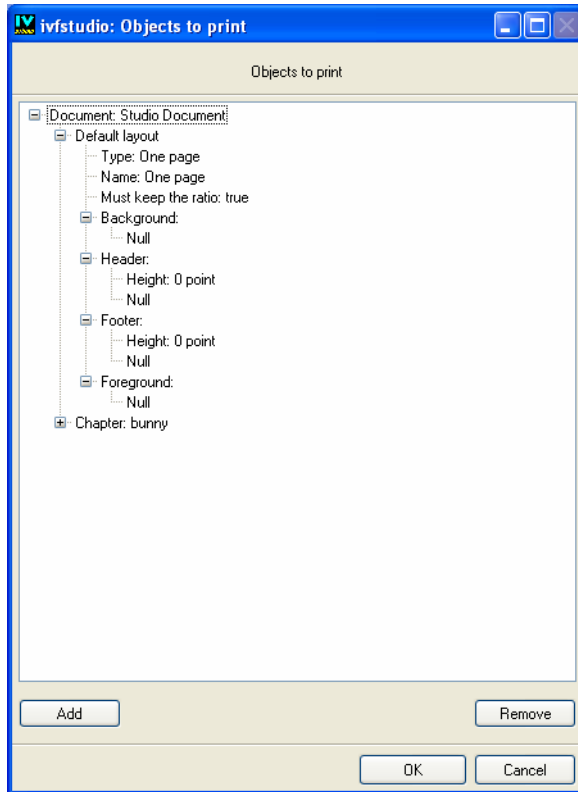
◆ Multiple pages

The chapter is printed on the number of pages specified in the layout.
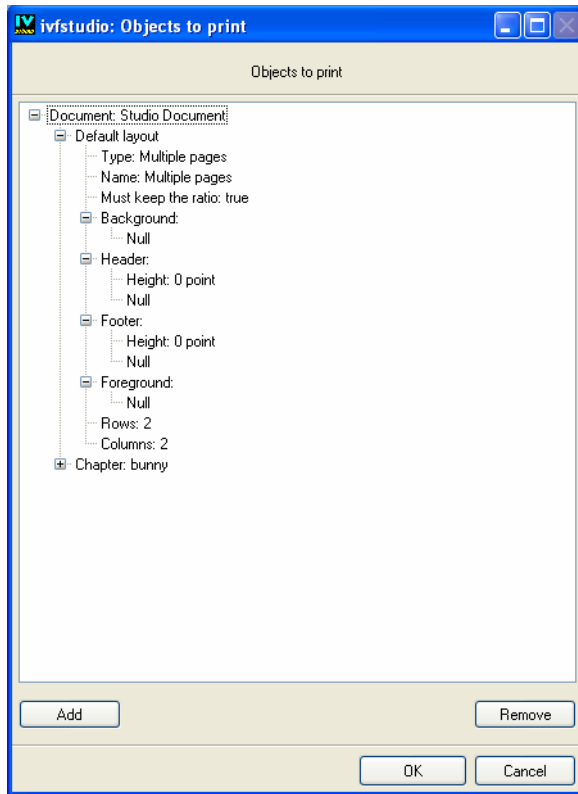
◆ Fixed size

The chapter is printed in an area whose size is specified in the layout.
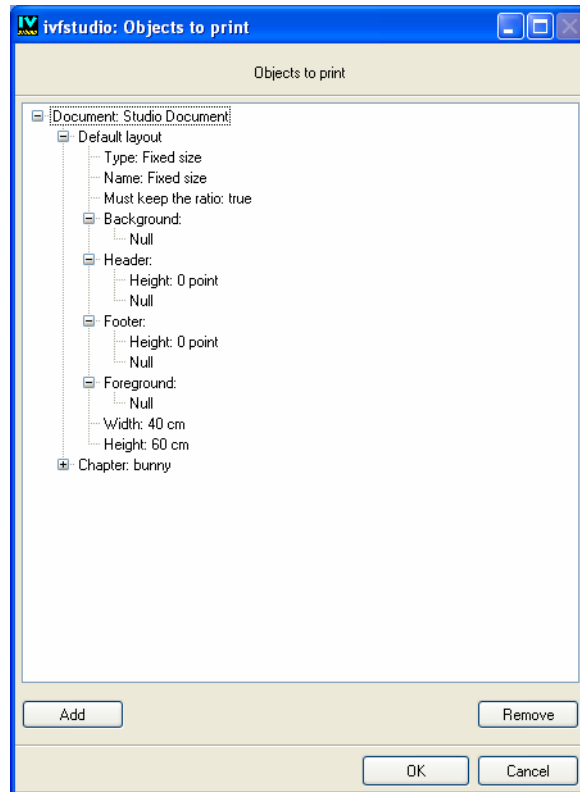
Replacing or Specifying the Layout of a Chapter

To replace or specify the layout of a chapter, right-click the item *Layout*.

Modifying the Properties of a Layout

You can modify most of the properties of a layout (the default one, or one associated with a chapter). Expand the item of the layout, if needed. Your tree now appears as in the following picture:
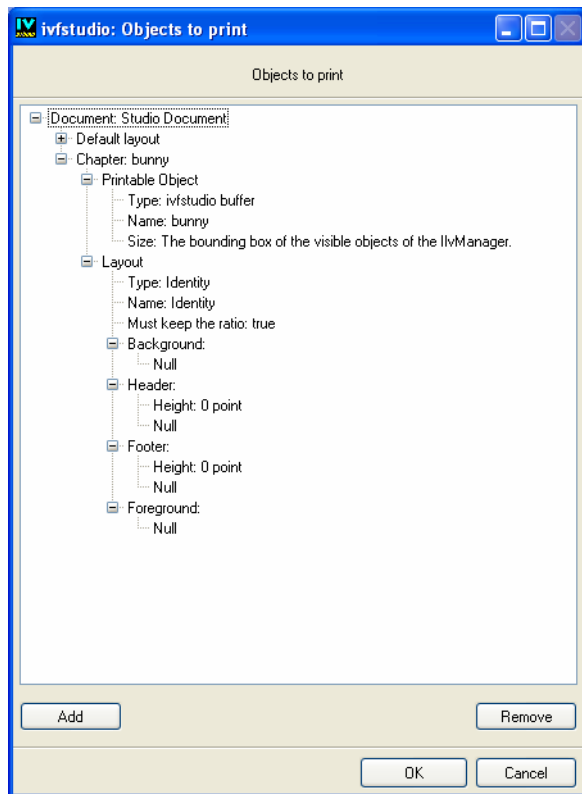
Most of the properties are common to all the layouts, but some layouts may have specific properties.

Common Properties of Layouts

The common properties are all editable, except the property *type*.

The common properties are the following:

◆ type

This property is not editable.

◆ name

Right-click this item if you want to edit it, and enter the new name in the Prompt String dialog box.

◆ must keep the ratio

Right-click this item if you want to choose the Boolean value, by using the following dialog box :
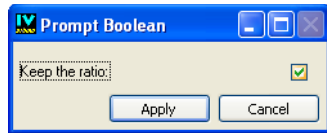
The properties of the selected printable object (if any) may be edited in the same way as the printable object associated with a chapter.

◆ background

Choose null if you do not want a background.

◆ header

Choose null if you do not want a header.

◆ footer

Choose null if you do not want a footer.

◆ foreground

Choose null if you do not want a foreground.

Specific Properties of the Layout Multiple Pages
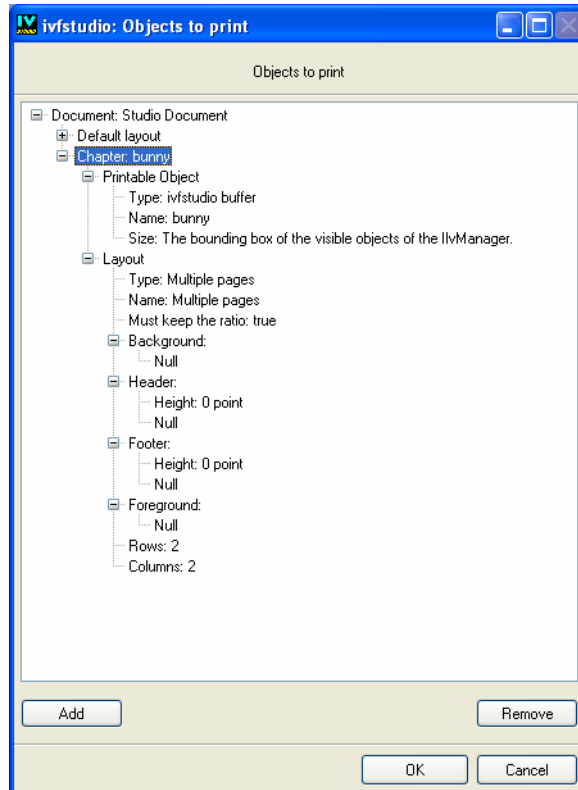
The specific properties of the layout multiple pages are the following:

◆ rows

   Identifies the number of rows of pages.

◆ columns

   Identifies the number of columns of pages.

Right-click these items if you want to edit them, and enter the new value in the following dialog box:
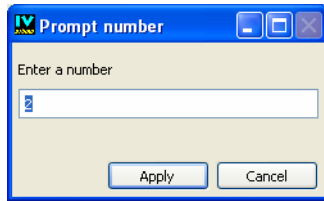
Specific Properties of the Layout Fixed Size
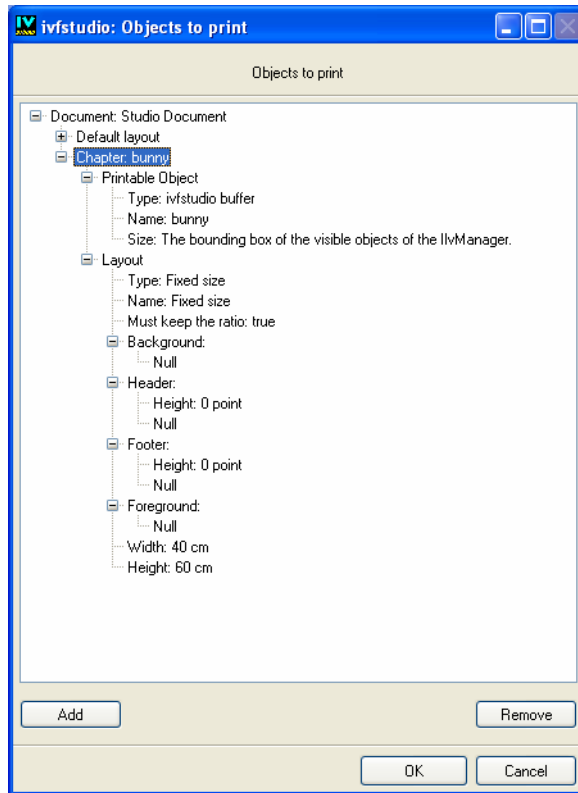
The specific properties of the layout fixed size are the following:

◆ width

Width of the area.

◆ height

Height of the area.

Right-click these items if you want to edit them, and enter the new value in the following dialog box:
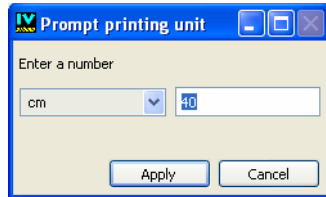
Print or Preview the Document

Once you have finished and you want to print or preview your document, click the OK button.

Cancel the Print or the Preview

If you want to stop (for example to modify a buffer, or to load another file), click the Cancel button. You will be able to get the document in the same state as you left it (except if one or several buffers were closed) and to keep on editing it later.

3

The IBM ILOG Views Studio Interface

This chapter provides information about the various components of the
IBM® ILOG® Views Studio interface.

◆ The *Main Window* section provides detailed descriptions of the elements that make up
the window.

◆ The *Palettes Panel* section describes the palettes available in the Foundation Studio for
creating graphic objects in your buffer windows.

Main Window

Menu Bar
Action Bar
Editing Modes
Palettes Panel
2D Graphics Window
Resource Editor
Generic Inspector
Message Area
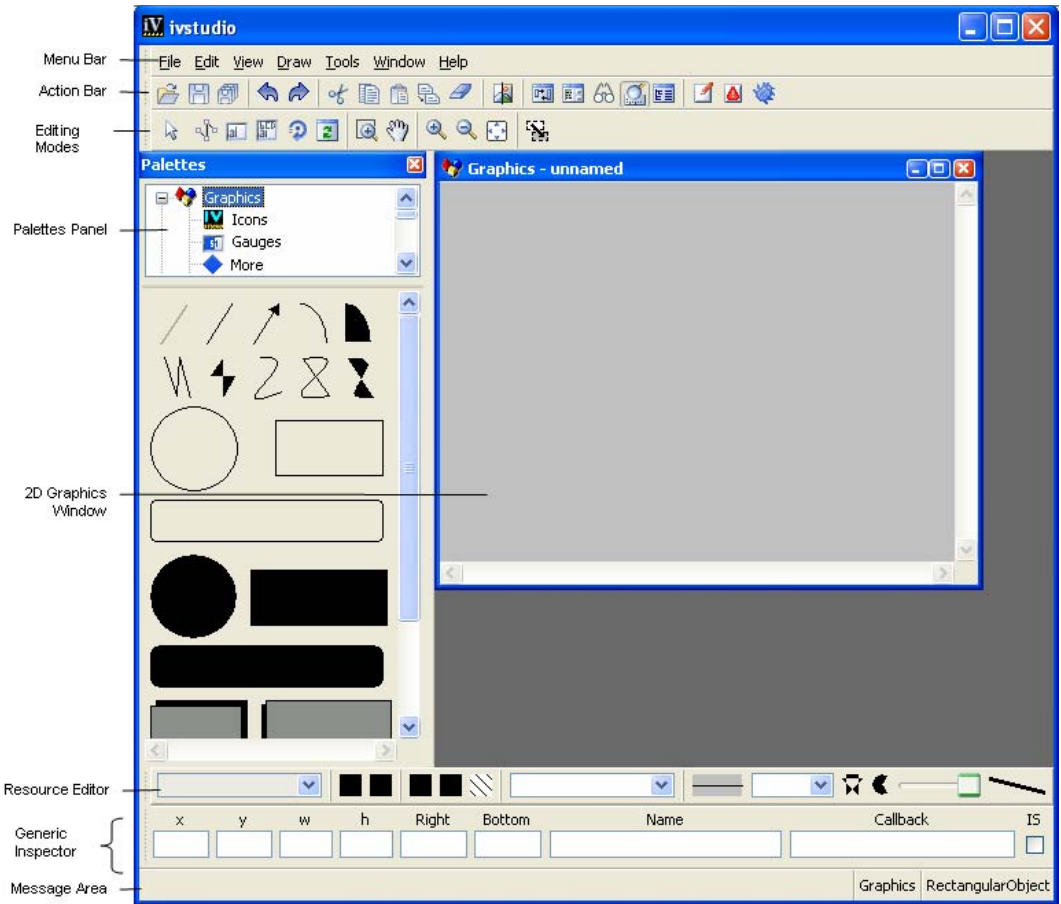
Figure 3.1   Main Window

The Main window appears when IBM ILOG Views Studio is launched and contains the following areas:

◆ *The Menu Bar*

◆ *The Action Toolbar*

◆ *The Editing Modes Toolbar*

◆ *The Buffer Windows*

◆ *The Resource Editor*

◆ *The Generic Inspector*

◆ *The Message Area*

The Menu Bar

The menu bar in the Main window provides you with general IBM® ILOG® Views Studio commands. When you highlight a command item, the Message area gives you a short description of the command. The following tables summarize the commands you can execute through the menu bar. For details on these commands, see Chapter 5, *Commands*, where they are listed in alphabetical order.

File Menu Commands

Menu Item	Command
New > 2D Graphics	NewGraphicBuffer
Open…	OpenDocument
Revert	RevertDocument
Close	CloseDocument
Save	SaveDocument
Save As…	SaveDocumentAs
Save All	SaveAll
Print the Current Buffer	PrintCurrentBuffer
Preview the Current Buffer...	PrintPreviewCurrentBuffer
Print...	Print
Preview...	PrinterPreview
Quit	Quit

Edit Menu

Menu Item	Command Name
Undo	UnDo
Redo	Redo

Menu Item	Command Name
Repeat	Repeat
Cut	ClipboardCut
Copy	ClipboardCopy
Paste	ClipboardPaste
Duplicate	Duplicate
Delete	Delete
Select All	SelectAll
Fit to Contents…	FitToContents

View Menu

Menu Item	Command Name
Grid	ToggleGrid
Snap to Grid	ToggleActiveGrid
Foreground Grid	ToggleForegroundGrid
Grid Options...	ShowGridOptionsPanel
Zoom In	ZoomIn
Zoom Out	ZoomOut
Original Size	ResetView
Fit to View	FitViewToContents
Options > Double Buffering	DoubleBuffering
Options > Opaque Move	OpaqueMoving
Options > Windows Look	Windows Look
Options > Windows 95 Look	Windows95Look
Options > Windows XP Look	WindowsXPLook
Options > Motif Look	MotifLook

Menu Item	Command Name
Options > English (US)	SetLanguage_en_US
Options > French (France)	SetLanguage_fr_FR

Draw Menu

Menu Item	Command Name
Group	Group
Ungroup	UnGroup
Edit Group	EditGroup
Group Into > IlvGraphicSet	GroupIntoGraphicSet
Group Into > IlvSelector	GroupIntoSelector
Group Into > IlvSmartSet	GroupIntoSmartSet
Layer...	SetLayer
Send to Back	Push
Bring to Front	Pop
Use Transformer	ToggleTransformed
Use Filter	MakeFilteredGraphic
Nudge > Up	NudgeUp
Nudge > Left	NudgeLeft
Nudge > Down	NudgeDown
Nudge > Right	NudgeRight
Transform > Flip Horizontal	FlipHorizontal
Transform > Flip Vertical	FlipVertical
Transform > Rotate Right	Rotate90
Transform > Rotate Left	Rotatem90
Transform > Rotate	SelectRotateMode
Align/Distribute > Align Left	AlignLeft

Menu Item	Command Name
Align/Distribute > Center Vertically	AlignVertical
Align/Distribute > Align Right	AlignRight
Align/Distribute > Align Top	AlignTop
Align/Distribute > Center Horizontally	AlignHorizontal
Align/Distribute > Align Bottom	AlignBottom
Align/Distribute > Equal Width	SameWidth
Align/Distribute > Equal Height	SameHeight
Align/Distribute > Distribute Horizontally	HorizontalSpaces
Align/Distribute > Distribute Vertically	VerticalSpaces
Align/Distribute > Alignment Panel	ShowAlignPanel
Lock Inspector	LockInspector
Object Inspector	ShowInspector

Tools Menu

Menu Item	Command Name
View Inspector	ShowGridPanel
Layer Editor	ShowLayerPanel
Script Editor	ShowScriptEditor
Script Error List	ShowScriptErrorList
Script Debugger	ShowScriptDebugger
Palettes	ShowPalettePanel
Generic Inspector	ShowGenericInspector
Property Sheet	ShowPropertiesPanel
Messages Editor	ShowMessagePanel
Commands	ShowCommandPanel
Errors	ShowErrorPanel

Menu Item	Command Name
Preferences...	EditOptions
Select Plug-Ins...	SelectPlugIns

Window Menu

Menu Item	Command Name
Close All	CloseAllBuffers
Minimize All	MinimizeAllWindows
Restore All	RestoreAllWindows
Cascade	CascadeWindows
Tile Horizontally	TileHorizontally
Tile Vertically	TileVertically

In addition to the options listed above, the Window menu contains the name of all opened buffer windows. This list is automatically modified when you open, create, close or rename a buffer.

Help Menu

Menu Item	Command Name
Info...	ShowInfoPanel
Main Panel ...	HelpMainPanel

The Action Toolbar

Use the Action toolbar, located underneath the menu bar, to quickly access commands. If the Main window is not large enough to display all the buttons, two arrow buttons are displayed for scrolling the toolbars.

Figure 3.2   *Action Toolbar*

The Action toolbar has the following icons, each representing a command:

Open Displays a file selector dialog box to specify an existing `.ilv` file to open in a buffer window. See *OpenBuffer* on page 111.

Save Saves the files modified in the current buffer. See *SaveDocument* on page 119.

Save All Saves all the files modified in the current buffer.

Undo Undoes the last action performed in the buffer window. See *UnDo* on page 137.

Redo Undoes the last action performed in the buffer window. See *Redo* on page 115.

Cut Removes the currently selected object or objects from the buffer window and places them in the clipboard. See *ClipboardCut* on page 102.

Copy Places a copy of the currently selected object or objects to the clipboard. See *ClipboardCopy* on page 102.

Paste Pastes the current clipboard object or objects in the buffer window. When you click the paste icon, the pointer changes to a crosshair pointer so that you can indicate where in the buffer window you want the object to be placed. See *ClipboardPaste* on page 102.

Duplicate Creates an exact copy of the currently selected object and places it in the buffer window. See *Duplicate* on page 104.

Delete Deletes the object or objects currently selected in the buffer window. See *Delete* on page 103.

Enable/Disable Enables/Disables the graphic effects for a graphic object (Ctrl+F).

Alignment Displays the Alignment panel. See *ShowAlignPanel* on page 130.

Commands Displays the Commands panel. See *ShowCommandPanel* on page 130.

Inspect Displays the object inspector for the currently selected object in the buffer window. See *ShowInspector* on page 132.

Generic Inspector Shows or hides the generic inspector in the Main window. See *ShowGenericInspector* on page 131.

Property Sheet Shows or hides the property sheet in the Main window. See *ShowPropertiesPanel* on page 134 .

Script Editor Shows or hides the Script Editor at the bottom of the Main window. See *ShowScriptEditor* on page 96.

Script Error List Shows or hides the Script error list at the bottom of the Main window that displays syntax errors. See *ShowScriptErrorList* on page 96.

Script Debugger Shows or hides the Script Debugger panel that lets you debug your scripts. See *ShowScriptDebugger* on page 96.

The Editing Modes Toolbar

Use the Editing Modes toolbar to select an editing mode for the current buffer window. If the Main window is not large enough to display all the buttons, two arrow buttons are displayed for scrolling the toolbars.

Figure 3.3   Editing Modes Toolbar

The Editing Modes toolbar contains the following icons:

Selection Mode Use the Selection mode for selecting, creating, deleting, moving, resizing and performing other common editing operations. This mode is selected when IBM® ILOG® Views Studio is launched.

The Selection mode lets you:

◆ Select objects:

- Click an object to make that object the unique selected object.

- Use Shift + Click to add (or remove) objects one by one to (or from) the current selection to select multiple objects.

- Click the background and draw a rectangle with the pointer to select all objects that are contained in the drawn rectangle.

◆ Move objects:

- Click a selection and drag the pointer to move all the selected objects.

◆ Resize an object:

- Click one of the corners or side handles of an object and drag the pointer to resize it. Use the Shift key whilst resizing to maintain the aspect ratio of this object. This is particularly useful when resizing images or objects that must maintain a fixed aspect ratio. Note that a few objects cannot be resized, such as labels, for instance.

◆ Edit individual points:

- Click once on a selected object, such as a line, a polygon or a Spline curve to move individual points independently.

PolySelection Mode Use this mode to move or rotate the different points of your `IlvPolyline`, `IlvPolygon`, `IlvSpline`, `IlvFilledSpline`, and `IlvClosedSpline` objects. To complete the operation, double-click the work space or select another mode.

Each object type has its own mode of operation:

◆ Editing polylines and polygons:

- Click and drag a point to move it.

- Click on a segment to move it

- Click on a point then press delete or backspace to remove it

- Shift-click to select multiple points and move them altogether

- Ctrl-click on a segment to insert a point at that position

◆ Editing Bezier splines (closed or not):

- Click and drag on a point to select it or move it. The tangent and tension handles for this point appear when you select a point.

- Click and drag on a tangent/tension handle to change the tangent and tension of the curve at that point.

- Press the Ctrl key while dragging a tangent handle to create/remove a cusp at the Bezier curve's position by dissociating the left and right tangents and tensions at that point.

- Click on a point then press delete or backspace to remove it

- Shift-click to select multiple points and move them altogether

- Ctrl-click on the curve outside of a point to insert a point at that position

◆ Editing arcs (see *Arc Mode* in the Palette section).

Label Mode Use this mode to create and edit an `IlvLabel` object. After selecting this interactor, click the work space to indicate the label position and type the string you want. Press Enter to complete the operation.

To edit an existing `IlvLabel` object, select this mode and click the `IlvLabel` object you want to edit.

Label List Mode Use this mode to create and edit a multiline label (`IlvListLabel`) object. After selecting this interactor, click the work space to indicate the label position and type the string you want. You can go to a new line by pressing Enter. Double-click the work space (outside this object) to complete the operation.

To edit an existing `IlvListLabel`, select this mode and click the `IlvListLabel` you want to edit.

Rotate Mode Use this mode to rotate an object in the buffer window. First, select the object you want to rotate in the buffer window. Click the Rotate Mode icon in the Editing Modes toolbar. Then click the left mouse button in the buffer window. An arrow appears in the buffer window. Drag the mouse to indicate the angle of rotation. When you release the mouse button, the object will rotate the specified amount.

Active Mode Use the Active mode to test the behavior of your objects and edit some of their properties. In the Active mode, the objects in the work space can respond to mouse and keyboard events. You can thus change text field labels, toggle the state of a toggle button, and so on.

Zoom Mode Use this mode to enlarge a portion of the buffer window so you can see it in greater detail.

Pan Mode Use the Pan mode to pan an area of the buffer window. This mode allows you to move around within a buffer window to see all the contents of the window.

Zoom In Mode Use this mode to zoom in an object in the buffer window.

Zoom Out Mode Use this mode to zoom out an object in the buffer window.

Fit to View Mode Use this mode to fit all the contents of the buffer within the window.

In the course of interaction, all editing operations can be cancelled by pressing the Escape key. Once an operation has been performed, it can be undone using the Edit > Undo command.

The Buffer Windows

You will create your graphic objects in the buffer windows that are displayed within the IBM® ILOG® Views Studio Main window. When using the Foundation Studio, you have access to only one type of buffer window, a 2D Graphics window. By default, an empty buffer window is displayed when IBM ILOG Views Studio is launched. The current buffer window type is shown at the bottom of the Main window in the message area.

A 2D Graphics buffer window allows you to edit the contents of an `IlvManager` or an `IlvContainer`. It uses an `IlvManager` object to load, edit, and save objects. You can have multiple buffer windows open in the work space.

To create a new 2D Graphics buffer window, choose New from the File menu, then 2D Graphics from the submenu.

To open a 2D Graphics buffer, you can also execute the `NewGraphicBuffer` command from the Commands panel, which you can display by choosing Commands from the Tools menu.

When you open a `.ilv` file that was generated by an `IlvManager`, a 2D Graphics buffer window is automatically opened.

Using the small boxes in the top-right corner, you can minimize or maximize the buffer windows in the work space.

> *Note: As you use IBM ILOG Views Studio extensions provided with other IBM ILOG Views packages, you will have access to other types of buffer windows (Gadgets, Grapher, Application, and Prototypes). Each buffer type provides access to additional features of IBM ILOG Views Studio and has its own set of editing modes. In addition, the behavior of certain commands varies depending on the current buffer. For example, the Test command tests just the panel if you are editing a Gadgets buffer, but it tests all the panel instances in an application if you are editing an Application buffer.*

The Generic Inspector

Use the Generic inspector area to display or edit the general properties of a selected object. You can use it to name the currently selected object, and edit its position, size, and callback, if required. Only one object can be inspected at a time. If you select more than one object,

the last object selected is inspected. If you deselect an object while at least two objects remain selected, the Generic inspector does not inspect any object.

Note: All graphic objects and panel names should have a valid C++ name, that is, a continuous string (no spaces) that starts with an ASCII character from a to z or A to Z. This restriction is mandatory if you wish to use the corresponding object in the C++ generated code and/or with the States package.

Figure 3.4 *The Generic Inspector*

The Generic inspector contains the following fields:

◆ **x** The distance between the left side of the object and the left side of the buffer window.

◆ **y** The distance between the top of the object and the top of the buffer window.

◆ **w** The width of the object.

◆ **h** The height of the object.

◆ **Right** The distance between the right side of the object and the left side of the buffer window.

◆ **Bottom** The distance between the bottom of the object and the top of the buffer window.

◆ **Name** The name of the object.

◆ **Callback** The callback name for the object.

◆ **IS** indicates whether the callback is an IBM ILOG Script callback or not.

The Resource Editor

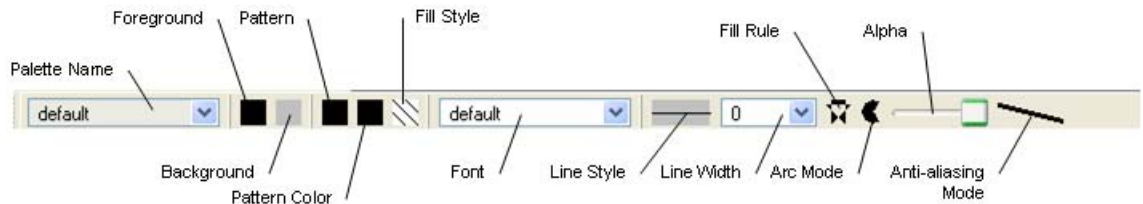Use the resource editor to edit the various graphic resources of selected objects.

Figure 3.5 *The Resource Editor*

The resource editor contains the following elements:

◆ **Palette Name** This combo box displays the name of the current palette. It can be edited.

Click the arrow button of the combo box to display a list of named palettes. You can then choose one of these named palettes. When a named palette is selected, all its resources are displayed in the Resources editor.

◆ **Foreground** The current foreground color is displayed in the box to the right of the palette name field. To change the foreground color, click the box and choose a new color from the color palette that appears.

◆ **Background** The current background color is displayed in the box to the right of the palette name field. To change the background color, click the box and choose a new color from the color palette that appears.

◆ **Pattern** The current fill pattern is shown in the box to the right of the background square. To change the fill pattern, click the box and choose a new pattern.

◆ **Pattern Color** The current pattern color is shown in the box to the right of the Pattern box. To change the pattern color, click the box and choose a new color.

◆ **Fill Style** The current fill style is displayed in the box to the right of the Pattern Color box. To change the fill style, click the fill style box and choose a new style.

◆ **Font** The current font name is displayed in the combo box to the right of the line width. You can select a predefined font name in the combo box or click Choose... to select another font.

◆ **Line Style** The current line style is shown by the horizontal line to the right of the fill pattern. To change the line style, click the line style box and choose a new style.

◆ **Line Width** The current line width value is shown in this combo box to the right of the line style. You can choose a new line-width value from the drop-down list.

◆ **Fill Rule Mode** The current fill rule mode is shown by the fill mode icon. This setting determines how a polygon is filled. Click the icon to change the fill mode.

◆ **Arc Mode** The current Arc mode is shown by the arc mode icon. Click the icon to change the arc mode.

◆ **Alpha** The current Alpha value is shown by a slider. The alpha value indicates the amount of transparency the drawing will be given. To change the alpha value, move the slider.

◆ **Antialiasing Mode** The current antialiasing mode is shown by the oblique line to the right of the Alpha. The anti-aliasing mode indicates whether smooth lines are drawn using anti-aliasing. To change the antialiasing mode, click the oblique line.

The fill rule indicates which points are to be considered as inside a polygon that is to be filled, depending on the count of crossing segments that define the shape of the area to be filled.

Click the even/odd rule icon ![icon] to select the even/odd rule for filling the area of a polygon. According to this rule, in the case of the complex polygon shown below, the central area of the star is not considered to lie inside the polygon, and therefore, is not filled. This is the default value.

Click the winding rule icon ![icon] to select the winding rule for filling the area of a polygon. According to this rule, the central area of the star is considered to lie inside the polygon, and therefore, is filled.
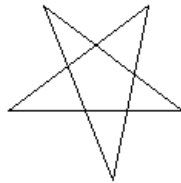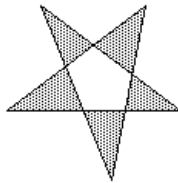
Original Polygon   Even/Odd Rule   Winding Rule
Before Fill
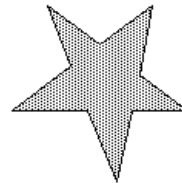
◆ **Arc Mode** The current arc mode is shown by the filled arc icon. This setting determines how an arc is closed. Click the icon to change the arc mode.

Click the arc pie mode icon ![icon] to select the arc pie mode. When this mode is selected, the arc is closed by adding two lines, from the center of the complete circle to the start and end points of the arc. This is the default mode.

Click the arc chord mode icon ![icon] to select the arc chord mode. When this mode is selected, the arc is closed by adding a line from the start point to the end point.

Arc Pie Mode   Arc Chord Mode

The Message Area

The Message Area contains current information about your IBM® ILOG® Views Studio session.
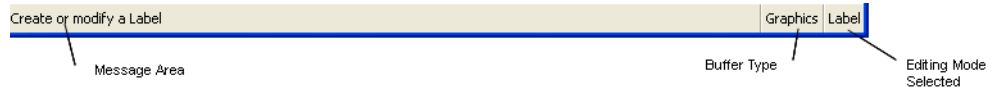
Figure 3.6   The Message Area

The message area displays:

◆ a short description of the command corresponding to the highlighted pop-up menu or toolbar items,

◆ the type of the object selected in the current buffer window, or

◆ information, warning and error messages.

Buffer Type

To the right of the message area appears the type of the current buffer window.

Current Editing Mode

The name of the current editing mode is displayed in the bottom-right corner of the Main window.

Palettes Panel

The Palettes panel provides you with predefined gadgets and graphics that you can drag and drop directly to your work space to create the graphic panels that will constitute your final application. See *Creating Objects* on page 28.

This panel is displayed by default when you launch IBM ILOG Views Studio, together with the Main window. If you decide to hide it, you can show it again by choosing Palettes from the Tools menu. To display it, you can also:

1. Choose Commands from the Tools menu.

2. Select the `ShowPalettePanel` command in the list and click Apply.
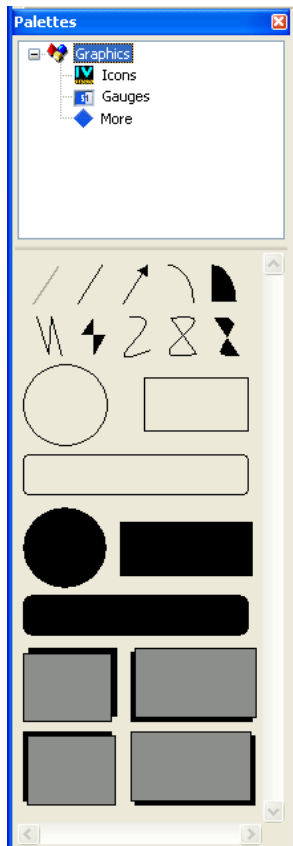
Figure 3.7   Palettes Panel

As you can see from the illustration, the Palette panel is divided into two panes. The upper pane is composed of a tree gadget in which you select the type of palette you want to display in the lower pane.

Graphics Palette

The following object creation modes are available in the Graphics palette.

Relief Line Mode

Use this mode to draw an `IlvReliefLine` object. After selecting this mode, press the left mouse button to indicate the first point, drag the cursor to the second point, and release the mouse button to complete the operation.

Line Mode

Use this mode to draw an `IlvLine` object. After selecting this mode, press the left mouse button to indicate the first point, drag the cursor to the second point, and release the mouse button to complete the operation.

Arrow Line Mode

Use this mode to draw an `IlvArrowLine` object. After selecting this mode, press the left mouse button to indicate the first node, drag the cursor to the second node, and release the mouse button to complete the operation.

Arc Mode

Use this mode to draw an `IlvArc` object.

After selecting this mode, do the following:

1. Drag a rectangle to indicate the bounding box of the arc:
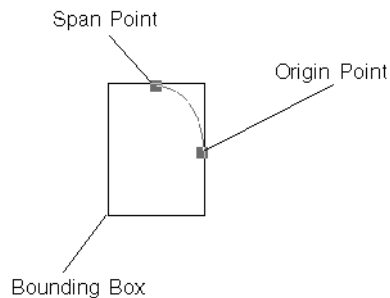
2. Drag the span point to indicate the span of the arc:

3. Drag the origin point to rotate the arc to the proper position.

4. Double-click to complete the operation.

Filled Arc Mode

Use this mode to draw an `IlvFilledArc` object in the same way as described for using arc mode.

Polyline Mode

Use this mode to draw an `IlvPolyline` object. After selecting this mode, click in the work space to indicate the points in the order you want them to be in the polyline.

● Press the mouse button and drag the pointer and release the mouse to add a point to your polyline.

● Press the spacebar or enter key to end the creation of your polyline. Alternatively, you can double click to end the operation, but this may result in extraneous points being added to the polyline, as a slow double click is interpreted as consecutive clicks, and therefore adds two points to the polyline.

Polygon Mode

Use this mode to draw an `IlvPolygon` object. After selecting this mode, click in the work space to indicate the points in the order you want them to be in the polygon.

● Press the mouse button and drag the pointer and release the mouse to add a point to your polygon.

● Press the spacebar or enter key to end the creation of your polygon. Alternatively, you can double click to end the operation, but this may result in extraneous points being added to the polygon, as a slow double click is interpreted as consecutive clicks, and therefore adds two points to the polygon.

Spline Mode

Use this mode to draw an `IlvSpline` object. After selecting this mode, click in the work space to indicate the points in the order you want them to be in the spline. See *Creating Bezier curves (open, closed and filled splines)*.

Closed Spline Mode

Use this mode to draw an `IlvClosedSpline` object. After selecting this mode, click in the work space to indicate the points in the order you want them to be in the closed spline. See *Creating Bezier curves (open, closed and filled splines)*.

Filled Spline Mode

Use this mode to draw an `IlvFilledSpline` object. After selecting this mode, click in the work space to indicate the points in the order you want them to be in the filled spline. Double-click to complete the operation. See *Creating Bezier curves (open, closed and filled splines)*.

Creating Bezier curves (open, closed and filled splines)

- Press the mouse button and drag the pointer and release the mouse to add a point and define its tangents and tension at that point.

- Press the Ctrl key while dragging a tangent handle to create a cusp at the Bezier curve's position and dissociate the left and right tangents and tensions at that point.

- Press the spacebar or enter key to end the creation of your curve. Alternatively, you can double click to end a curve, but this may result in extraneous points being added to the curve, as a slow double click is interpreted as consecutive clicks, and therefore adds two points to the curve.

Other Objects

The Graphics palette contains other objects that can be created using the normal creation mode or drag-and-drop operation.

Type	Icon	Description
Ellipse		Draws an `IlvEllipse` object.
Rectangle		Draws an `IlvRectangle` object.
Round Rectangle		Draws an `IlvRoundRectangle` object.

Type	Icon	Description
Filled Ellipse		Draws an `IlvFilledEllipse` object.
Filled Rectangle		Draws an `IlvFilledRectangle` object.
Filled Round Rectangle		Draws an `IlvFilledRoundRectangle` object.
Shadow Rectangle		Draws an `IlvShadowRectangle` object.

In the course of interaction, all editing operations can be cancelled by pressing the Escape key. Once an operation has been performed, it can be undone using the Edit > Undo command.

Icons Palette
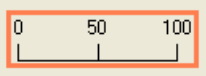
The Icons palette contains the following objects that can be created using the normal creation mode or drag-and-drop operation.
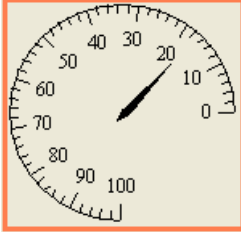
Type	Icon	Description
Icon		Draws an `IlvIcon` object.
Transparent Icon		Draws an `IlvTransparentIcon` object.

Type	Icon	Description
Zoomable Icon		Draws an `IlvZoomableIcon` object.
Zoomable Transparent Icon		Draws an `IlvZoomableTransparentIcon` object.

Gauges Palette

The Gauges palette contains various types of gauges that can be created using the normal creation mode or drag-and-drop operation.
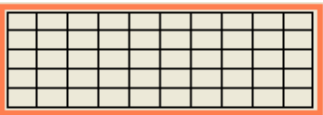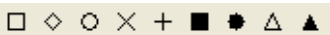
Type	Icon	Description
Rectangular Scale	0 50 100	Creates an `IlvRectangularScale` object. You can choose either a horizontal or a vertical orientation for the scale.
Rectangular Gauge		Creates an `IlvRectangularGauge` object. You can choose either a horizontal or a vertical orientation for the scale.
Relief Gauge		Creates an `IlvReliefGauge` object. You can choose either a horizontal or a vertical orientation for the scale.
Slider Gauge		Creates an `IlvSliderGauge` object. You can choose either a horizontal or a vertical orientation for the scale.

Type	Icon	Description
Filled Circular Gauge		Creates an `IlvFilledCircularGauge` object.
Circular Gauge		Creates an `IlvCircularGauge` object.

More Palette

The More palette contains other objects that can be created using the normal creation mode or drag-and-drop operation.

Type	Icon	Description
Label	A label	Creates an `IlvLabel` object.
List Label	This is an IlvListLabel	Creates an `IlvListLabel` object.
Zoomable Label	Zoomable Label	Creates an `IlvZoomableLabel` object.
Relief Diamond		Creates an `IlvReliefDiamond` object.

Type	Icon	Description
Relief Rectangle		Creates an `IlvReliefRectangle` object.
Shadow Label	Label with a Shadow	Creates an `IlvShadowLabel` object.
Relief Label	Relief Label	Creates an `IlvReliefLabel` object.
Polyline Arrow		Creates an `IlvArrowPolyline` object.
Outline Polygon		Creates an `IlvOutlinePolygon` object.
Grid Rectangle		Creates an `IlvGridRectangle` object.
Marker	□ ◇ ○ ✕ + ■ ✹ △ ▲	Creates an `IlvMarker` object. You can choose any of the marker shapes in the palette.

C   H   A   P   T   E   R

4

Using IBM ILOG Script

This chapter describes the Studio IBM ILOG Script extension (`jsstudio`) that integrates
the IBM ILOG Script language. This extension lets you use IBM ILOG Script with
IBM ILOG Views objects in Studio. You can attach callbacks written in IBM ILOG Script to
IBM ILOG Views objects, edit and test these callbacks, and generate C++ applications using
IBM ILOG Script for IBM ILOG Views.

The Studio IBM ILOG Script extension is included in IBM ILOG Views Foundation Studio.
You can find information on the following topics in this chapter:

◆ *Using IBM ILOG Script for IBM ILOG Views in your Application*

◆ *Commands Specific to the IBM ILOG Script Extension*

Using IBM ILOG Script for IBM ILOG Views in your Application

IBM ILOG Script for IBM ILOG Views lets you dynamically access IBM ILOG Views
objects using a scripting language. The chapter "*ILOG Script Programming*" in the
IBM ILOG Views *Foundation User's Manual* fully describes how you can manipulate
IBM ILOG Views objects from IBM ILOG Script. `jsstudio` lets you write and play
IBM ILOG Script code in your IBM ILOG Views applications.

Callbacks

A callback that you attach to an object can be written either in C++ or in IBM ILOG Script. In `jsstudio`, you can specify the language you want to use in the Generic Inspector at the bottom of the Main window or in the Callbacks Inspector.

If most of your callbacks are written in IBM ILOG Script for IBM ILOG Views, you might want the callback language to default to `JvScript`. (`JvScript` is the symbolic name of IBM ILOG Script for IBM ILOG Views.) In this case, you can set the `defaultCallbackLanguage` option to `JvScript` in an option file.

To attach a callback written in IBM ILOG Script to an object, you can either enter a name in the callback field and select the `IS` toggle button to its right in the Generic Inspector or enter a name and select the Script toggle button to the right of the Value field in the Callbacks page of the object inspector panel (see Figure 4.1).
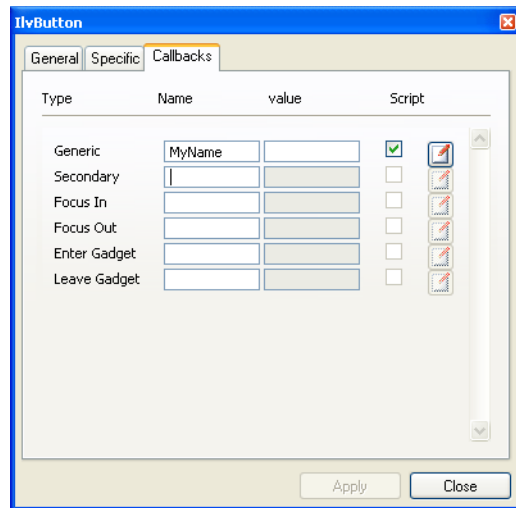
Figure 4.1 *Callbacks Page of the Button Inspector Panel*

You can access the object inspector panel by either double clicking the object or by selecting the object and clicking 👓 .

You can click the button to the right of the Script toggle to find and edit the IBM ILOG Script function corresponding to the specified callback. If this function is not defined in the inline script attached to the buffer, a sample function will be automatically generated.

The IBM ILOG Script function that you define must have the same name as its associated callback. It will be invoked whenever the callback is triggered. The function receives the graphic object to which the callback is associated as its first argument. If you specify a value

for the callback in the Value column of the Callback Inspector, this value is passed as the second parameter to the function.

Writing and Playing Scripts

You can edit an inline script attached to a buffer in the script editor. To display this editor, click on the Script Editor icon in the tool bar at the top of the Main window or execute the `ShowScriptEditor` command.

By default, the script editor appears as part of the Main window, underneath the drawing area. If you want the editor to be displayed in a separate top window, define a panel description for the `ScriptEditor` panel using the `topView` property, as shown below:

```
panel "ScriptEditor" {
    topView true;
}
```

For details, see *IBM ILOG Views Studio Panel Description File* on page 164.

When editing an inline script in the script editor, you do not have to explicitly validate or save it; the script is automatically saved in the `.ilv` file and read back when you edit your `.ilv` file again. The inline script you edit is attached to the current buffer. When a buffer is selected, the script editor is updated to edit the corresponding script.

When a container loads your `.ilv` file, the inline script is also loaded and all the expressions it contains are evaluated. In contrast, when you open an `.ilv` file in an IBM ILOG Views Studio buffer, script expressions are not evaluated.

When you test the currently edited application, application panels, which are IBM ILOG Views containers, load the `.ilv` files as well as the inline scripts. When you test the current buffer, IBM ILOG Views Studio writes its contents together with the inline script to a temporary stream and creates a container that loads that stream.

Scripts can also be written in a separate external file, with the `.js` extension, using any text editor. When loading an `.ilv` file, the IBM ILOG Views container checks whether there is a `.js` file with the same name in the current directory. If the container finds such a file, it loads it. You can explicitly load and evaluate an external script file by executing the IBM ILOG Views Studio `LoadScript` command or by calling the IBM ILOG Script function `loadFile`.

Application and Scripts

Note: This section is applicable only if you are using the GUI Application plug-in (to edit GUI applications) and GUI Generation plug-in (if you need to generate the code) with IBM ILOG Views Studio.

To use IBM ILOG Script for IBM ILOG Views in the generated C++ application, the appropriate libraries must be linked with your application and initialized by your application. If you want IBM ILOG Views Studio Script Edition to generate the required makefile and initialization code for you, check the Use IBM ILOG Script toggle button in the Script page of the Application inspector.
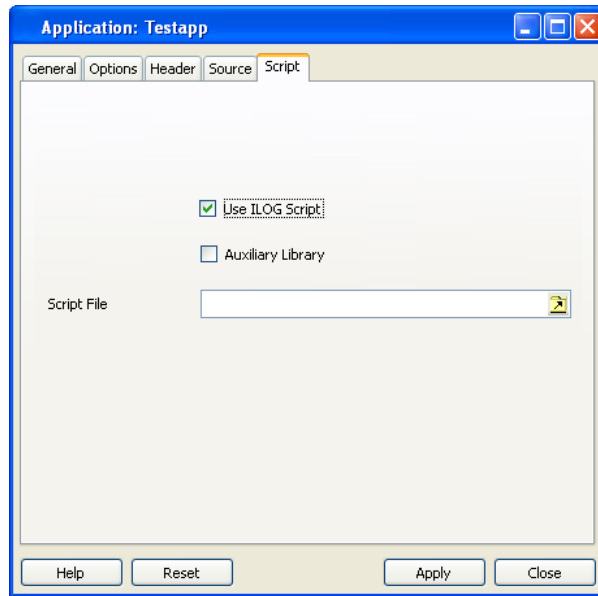
Figure 4.2   IBM ILOG Script Option Checked in the Application Inspector

Note: The Use IBM ILOG Script option is checked by default. If you want the option to default to the other choice, set the JVScriptApplication *option to* false *in an option file.*

A script file can be attached to an application. In this case, it is loaded when you test your application in ivfstudio and when you run the generated application. To specify a script file for your application, use the Script File field in the Script page of the Application Inspector.

Your application can be accessed by the global IBM ILOG Script variable Application. This variable can be used in the generated application, and in ivfstudio when the application is being tested.

Note: This global variable must not be used when you test the current buffer.

Syntax Errors

When you test the application or the current buffer, the script attached to the concerned buffers are evaluated. If your scripts contain syntax errors, these errors are displayed in the Script Error List. You can double-click on an error to locate it and edit the source code in the Script Editor. Note that the buffer corresponding to the script is selected. You can show or hide the script error list by using the Script Error List icon in the tool bar.

By default, the Script Error List is displayed in the Main panel on top of the Generic Inspector. If you want it to be created in a separate top window, define a panel description for the `ScriptErrorList` panel using the `topView` property, as shown below:

```
panel "ScriptErrorList" {
   topView true;
}
```

For more information, see the section *IBM ILOG Views Studio Panel Description File* on page 164.

The contents of the Script Error List is updated each time you execute a `Test` command.

Commands Specific to the IBM ILOG Script Extension

The following commands are specific to `jsstudio`:

◆ `LoadScript` loads and evaluates an IBM ILOG Script source file.

◆ `ShowScriptErrorList` shows the syntax error list.

◆ `ShowScriptDebugger` activates the IBM ILOG Script Debugger panel.

◆ `ShowScriptEditor` activates a simple script editor.

LoadScript

Label	Load Script...
Path	File menu > Open... > Files of type Script Files (`*.js`)
Category	script
Action	Loads and evaluates an IBM ILOG Script source file. This command opens a File Selector panel that lets you choose an IBM ILOG Script source file.

ShowScriptErrorList

Label	Script Error List
Path	Tool bar
Category	script
Action	Displays syntax errors. When you test your application or the current buffer, the script attached to the concerned buffers are evaluated. Syntax errors, if any, are displayed in the Script Error list. You can double-click on an error message to locate and edit the source code in the Script Editor. Note that the buffer corresponding to the script is selected.

ShowScriptDebugger

Label	Script Debugger
Path	Tool bar
Category	script
Action	Displays the IBM ILOG Script Debugger panel that lets you debug your scripts and access the IBM ILOG Script virtual machine.

ShowScriptEditor

Label	Script Editor
Path	Tool bar
Category	script
Action	Displays the Script Editor that lets you edit the inline script attached to the current buffer.

5

Commands

A command is an IBM® ILOG® Views Studio basic entry point. It is mainly defined by an action, a label, a prompt message, and often an icon. It can be attached to menu and toolbar items, buttons, and other active gadgets in any IBM ILOG Views Studio panel.

This chapter introduces general command definitions and describes the IBM ILOG Views Studio predefined commands.

Introduction

A command has the following characteristics:

◆ **Name** A command is identified by its name. When it is not attached to a graphic object, a command can be executed through the Commands panel by its name. See *Commands Panel* on page 141.

◆ **Label** If a command is attached to a menu item, its label is used to set the label of that menu item. If a command is attached to a toolbar or Editing Mode button, its label is displayed in a small window when you leave the mouse pointer on the button for a short while.

◆ **Prompt** This property is a short description of the command. It is displayed in the Message area of the Main window when you either highlight a menu item or leave the mouse pointer on a button associated with that command.

◆ **Bitmap** When a command is attached to a button, the bitmap of the command is used to display the button.

◆ **State** Some commands can have a Boolean state. This is useful when the command action changes the state of something. For example, commands that select editing modes have their state modified when the associated modes are selected or deselected. Buttons associated with commands whose state is "true" are inverted.

◆ **Category** A command can be part of a category or not. The Commands panel classifies the commands by category.

Predefined Commands

This section presents an alphabetical listing of the predefined commands that are available in the Foundation Studio. For each command, it indicates its label, the category to which it belongs, how to access it if it is accessible other than through the Commands panel, and what it is used for.

To display the Commands panel, choose Commands from the Tools menu in the Main window or click the Commands icon ▥ in the Action toolbar.

AddIcon

Deprecated - use creation mode instead.

Label	IlvIcon
Category	add
Action	Lets you insert an `IlvIcon` object in the work space. This command opens a bitmap File Selector panel that lets you choose a bitmap. After choosing a bitmap, click in the work space to indicate the position of the object.

AddTransparentIcon

Deprecated - use creation mode instead.

Label	IlvTransparentIcon
Category	add
Action	Lets you insert an `IlvTransparentIcon` object in the work space. This command opens a bitmap File Selector panel that lets you choose a bitmap. After choosing a bitmap, click in the work space to indicate the object position.

AddZoomableIcon

Deprecated - use creation mode instead.

Label	IlvZoomableIcon
Category	add
Action	Lets you insert an `IlvZoomableIcon` object in the work space. This command opens a bitmap File Selector panel that lets you choose a bitmap. After choosing a bitmap, click in the work space to indicate the position of the object.

AddZoomableTransparentIcon

Deprecated - use creation mode instead.

Label	IlvZoomableTransparentIcon
Category	add
Action	Lets you insert an `IlvZoomableTransparentIcon` object in the work space. This command opens a bitmap File Selector panel that lets you choose a bitmap. After choosing a bitmap, click in the work space to indicate the position of the object.

AlignBottom

Label	Align Bottom
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Moves the selected objects vertically so their bottom borders are horizontally aligned. Objects dimensions are not changed by this operation. If only one object is selected, it is vertically moved to the bottom border of the panel or to the bottom guide to which the object may be attached. If at least two objects are selected, the first selected object is used as the reference.

AlignHorizontal

Label	Center Horizontally
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Moves the selected objects vertically so their centers are horizontally aligned. Object dimensions are not changed. If only one object is selected, it is vertically moved so its center is horizontally aligned with the center of the panel or between the guides to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

AlignLeft

Label	Align Left
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Moves the selected objects horizontally so their left borders are vertically aligned. Object dimensions are not changed. If only one object is selected, it is horizontally moved to the left border of the panel or to the left guide to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

AlignRight

Label	Align Right
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Moves the selected objects horizontally so their right borders are vertically aligned. Object dimensions are not changed. If only one object is selected, it is horizontally moved to the right border of the panel or to the right guide to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

AlignTop

Label	Align Top
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Moves the selected objects vertically so their top borders are horizontally aligned. Object dimensions are not changed. If only one object is selected, it is vertically moved to the top border of the panel or to the top guide to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

AlignVertical

Label	Center Vertically
Path	Main window: Draw menu > Align/Distribute; Alignment Panel
Category	align
Action	Moves the selected objects horizontally so their centers are vertically aligned. Object dimensions are not changed. If only one object is selected, it is horizontally moved so the center is vertically aligned with the center of the panel or between the guides to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

CascadeWindows

Label	Cascade
Path	Main window: Window menu
Category	buffer
Action	Arranges open buffer windows in the work space so that they overlap.

ClipboardCopy

Label	Copy
Path	Main window: Edit menu and Copy icon in the toolbar
Category	edit
Action	Copies the selected objects to the clipboard so they can be copied in the current buffer or another buffer through the Paste command.

ClipboardCut

Label	Cut
Path	Main window: Edit menu and Cut icon in the toolbar
Category	edit
Action	Cuts the selected objects to the clipboard so they can be inserted in the current buffer or another buffer through the Paste command.

ClipboardPaste

Label	Paste
Path	Main window: Edit menu and Paste icon in the toolbar
Category	edit
Action	Copies the objects in the clipboard to the work space. This command waits for you to click in the work space to indicate the insertion position.

CloseAllBuffers

Label	Close All
Path	Main window: Window menu

Category	buffer
Action	Closes all the opened buffers and creates a new default buffer.

CloseBuffer

Label	Close
Path	Main window: File menu
Category	buffer
Action	Closes the current panel.

CloseDocument

Label	Close
Path	Main window: File menu
Category	document
Action	Closes the current buffer. If this buffer is modified, you have to confirm the action. If the current buffer is the application buffer, the current application is discarded and an application is created.

Delete

Label	Delete
Path	Main window: Edit menu and Delete icon in the toolbar
Category	edit
Action	Deletes the selected objects. Unlike the Cut command, the clipboard is not modified.

DoubleBuffering

Label	Double Buffering
Path	Main window: Views menu > Options
Category	studio
State	Is "true" if the work space is using the double-buffering mechanism.
Action	Turns on/off the double-buffering mode for the display in the work space.

Duplicate

Label	Duplicate
Path	Main window: Edit menu and Duplicate icon in the toolbar
Category	edit
Action	Duplicates the selected objects.

EditOptions

Label	Preferences...
Path	Main window: Tools
Category	studio
Action	Opens the Edit Options dialog box that allows you to set preferences for your IBM ILOG Views Studio session.

EditGroup

Label	Edit Group
Accelerator	Ctrl+E
Path	Main window: Draw Menu > Edit Group

Category	group
Action	When an `IlvGraphicSet` or other type of group object is selected, opens a new edition buffer containing the objects held in this group. Objects can be added, removed or edited in this buffer. When the buffer is saved, the content if the buffer replaces the group content.

FitToContents

Label	Fit to Contents...
Path	Main window: Edit menu
Category	buffer
Action	Opens a dialog box that lets you set a margin to resize the work space so that the right-most and bottom-most objects are completely visible along with the margin you set in the dialog box.

FitViewToContents

Label	Fit to View
Path	Main window: View menu
Category	view
Action	Applies a transformer to the buffer window so that all the contents of the buffer fit within the window.

FlipHorizontal

Label	Flip Horizontal
Path	Main window: Draw menu > Transform
Category	edit
Action	Creates a mirror image of the object selected in the buffer window by flipping it horizontally.

FlipVertical

Label	Flip Vertical
Path	Main window: Draw menu > Transform
Category	edit
Action	Creates a mirror image of the object selected in the buffer window by flipping it vertically.

GenerateStringData

Label	Generate String Data...
Category	buffer
Action	Generates in a file a C++ character string corresponding to the current buffer data. This command opens a File Selector panel that lets you enter the file where the data string is generated. The current buffer must not be empty.

Group

Label	Group
Path	Main window: Draw menu
Category	group
Action	Groups the selected objects into the preferred type of group set that is specified in the Group Options panel.

GroupIntoGraphicSet

Label	IlvGraphicSet
Path	Main window: Draw menu > Group Into

Category	group
Action	Groups the selected objects into an object of the IlvGraphicSet class.

GroupIntoSelector

Label	IlvSelector
Path	Main window: Draw menu > Group Into
Category	group
Action	Groups the selected objects into an object of the IlvSelector class.

GroupIntoSmartSet

Label	IlvSmartSet
Path	Main window: Draw menu > Group Into
Category	group
Action	Groups the selected objects into an object of the IlvSmartSet class.

HelpMainPanel

Label	Main Panel
Path	Main window: Help menu
Category	panel
Action	Displays the main panel online help.

HorizontalSpaces

Label	Distribute Horizontally
Path	Main window: Draw menu > Align/Distribute; Alignment Panel

Category	align
Action	Creates an equal amount of horizontal space between the selected objects. The space between two objects is at least equal to a predefined value. Object dimensions are not changed. You must select at least two objects.

InspectPanel

Label	Panel Inspector
Path	Double-click the title bar of the panel instance in the application buffer or use its menu.
Category	application
Action	Opens the Panel Inspector panel.

LoadMessageDatabase

Label	Load Messages
Path	Main window: File menu > Open (to select a .dbm file)
Category	studio
Action	Loads a message database file to the current display message database.

LockInspector

Label	Lock Inspector
Path	Main window: Draw menu
Category	inspector, studio
State	True if the inspector is locked.
Action	Locks or unlocks the inspector. If the inspector is not locked, when you select an object in the work space this object follows the selection. If it is locked, the object does not follow the selection and remains unchanged until you explicitly inspect another object.

MakeFilteredGraphic

Label	Use Filter
Path	Main window: Draw menu
Category	inspector, studio
Action	Sets a graphic filter on the selection.

MinimizeAllWindows

Label	Minimize all
Path	Main window: Window menu
Category	buffer
Action	Reduces all the open buffer windows to their title bar.

MotifLook

Label	Motif Look
Path	Main window: View menu > Options
Category	look, studio
Action	Sets the display current look to Motif.

NewGraphicBuffer

Label	2D Graphics
Path	Main window: File menu > New
Category	buffer, 2d
Action	Creates a new 2D graphics buffer. This buffer becomes the current buffer.

NudgeDown

Label	Nudge Down
Path	Main window: Draw > Nudge
Category	edit
Action	Moves the selected object down one pixel.

NudgeLeft

Label	Nudge Left
Path	Main window: Draw > Nudge
Category	edit
Action	Moves the selected object to the left one pixel.

NudgeRight

Label	Nudge Right
Path	Main window: Draw > Nudge
Category	edit
Action	Moves the selected object to the right one pixel.

NudgeUp

Label	Nudge Up
Path	Main window: Draw > Nudge
Category	edit
Action	Moves the selected object up one pixel.

OpaqueMoving

Label	Opaque Move
Path	Main window: View menu > Options
Category	studio
State	True if the property is on.
Action	Turns on/off the OpaqueMoving property. When this property is true, the objects you are moving or resizing in the work space are entirely displayed when pointing to them with the mouse. Otherwise, their ghost image is displayed in XOR mode.

OpenBuffer

Label	Open…
Path	Main window: File menu and Open icon in the toolbar
Category	buffer
Action	Loads a panel data file in a new buffer which becomes the current buffer. This command opens a File Selector panel that lets you choose a previously saved file. If the specified file is already loaded, the open buffer is selected.

OpenDocument

Label	Open…
Path	Main window: File menu and Open icon in the toolbar

Category	document
Action	Loads all the file types that are declared by the `fileLoader` option. This option is documented in Chapter 7, *Customizing IBM ILOG Views Studio.* The following are the predefined file extensions: ◆ `ilv` executes `OpenBuffer` ◆ `iva` executes `OpenApplication` ◆ `dbm` executes `LoadMessageDatabase` ◆ `js` executes `LoadScript` (`jsstudio` only)

PlayerOpen

Label	Open…
Category	player
Action	Reads an event file in order to play back the recorded events. This command opens a File Selector panel that lets you choose an event file.

PlayerPlay

Label	Play
Category	player
Action	Plays back the recorded or loaded events.

PlayerSaveAs

Label	Save As…
Category	player
Action	Saves the recorded events to a file. This command opens a File Selector panel that lets you enter a file in which to save the events.

PlayerSpeed

Label	Speed...
Category	player
Action	Lets you change the speed of events when they are played back.

PlayerStart

Label	Start
Category	player
Action	Starts recording events. All keyboard and mouse events are recorded until you execute the `PlayerStop` command.

PlayerStop

Label	Stop
Category	player
Action	Stops recording events. Recorded events are ready to be played back or saved to an event file.

Pop

Label	Bring to Front
Path	Main window: Draw menu
Category	edit
Action	Brings the selected objects to the front of the buffer.

Print

Label	Print...
Path	Main window: File menu
Category	buffer
Action	Prints the selected object.

PrinterPreview

Label	Preview...
Path	Main window: File menu
Category	buffer
Action	Previews the selected object.

PrintCurrentBuffer

Label	Print the Current Buffer
Path	Main window: File menu
Category	buffer
Action	Prints the contents of the current buffer.

PrintPreviewCurrentBuffer

Label	Preview the Current Buffer...
Path	Main window: File menu
Category	buffer
Action	Previews the contents of the current buffer.

Push

Label	Send to Back
Path	Main window: Draw menu
Category	edit
Action	Sends the selected objects to the bottom of the buffer (behind all other objects).

Quit

Label	Quit…
Path	Main window: File menu
Category	studio
Action	Quits IBM ILOG Views Studio.

Redo

Label	Redo
Path	Main window: Edit menu and Redo icon in the toolbar
Category	edit
Action	Restores the last undone command.

Repeat

Label	Repeat
Path	Main window: Edit menu
Category	edit
Action	Repeats the last executed command.

Refresh

Label	Refresh
Category	buffer
Action	Redraws the contents of the work space.

ResetView

Label	Original Size
Path	Main window: Edit menu
Category	view
Action	Returns the current buffer to its original size after it has been zoomed in or zoomed out. Resets the view to a 1:1 scaling factor.

ResizePanel

Label	Resize Panel...
Category	buffer
Action	Lets you resize the current Gadgets buffer window. This command pops up a panel that represents a copy of the buffer and waits for you to resize it through the window manager. Once the panel is resized, you can validate the operation, or cancel it, using the Validation dialog box. Note that you can resize the buffer window by dragging its borders, if it does not have scroll bars.

RestoreAllWindows

Label	Restore all
Path	Main window: Window menu

Category	buffer
Action	Restores all the open buffer windows to their original size.

RevertBuffer

Label	Revert
Path	Main window: File menu
Category	buffer
Action	Reverts the current buffer to the saved file.

RevertDocument

Label	Revert
Path	Main window: File menu
Category	buffer, document
Action	Reverts the current document to the saved file.

Rotate90

Label	Rotate Right
Path	Main window: Draw menu > Transform
Category	edit
Action	Rotates the selected object 90 degrees clockwise.

Rotatem90

Label	Rotate Left
Path	Main window: Draw > Transform

Category	edit
Action	Rotates the selected object 90 degrees counterclockwise.

SameHeight

Label	Equal Height
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Resizes the selected objects so they have the same height. Object positions are not changed. If only one object is selected, it is resized so it has the same height as the panel or the distance between the two guides to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

SameWidth

Label	Equal Width
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Resizes the selected objects so they have the same width. Object positions are not changed. If only one object is selected, it is resized so it has the same width as the panel or the distance between the two guides to which it may be attached. If at least two objects are selected, the first selected object is used as the reference.

SaveBuffer

Label	Save
Path	Main window: File menu and Save icon in the toolbar
Category	buffer
Action	Saves the data file.

SaveBufferAs

Label	Save As...
Path	Main window: File menu and Save icon in the toolbar
Category	buffer
Action	Saves the current panel to a new file.

SaveAll

Label	Save All
Path	Main window: File menu
Category	buffer
Action	Saves the modified buffers and the application.

SaveDocument

Label	Save
Path	Main window: File menu and Save icon in the toolbar
Category	document
Action	Saves the application or the panel data file edited in the current buffer. If the buffer is saved for the first time or if it still has the default file name, this command executes the SaveDocumentAs command in order to let you enter a file name.

SaveDocumentAs

Label	Save As...
Path	Main window: File menu and Save icon in the toolbar

Category	document
Action	Opens a file selector which lets you enter a file name for the application or panel data edited in the current buffer.

SelectActiveMode

Label	Active
Path	Main window: Editing Modes toolbar
Category	mode
State	True if this mode is selected.
Action	Selects the Active mode. See *Active Mode* on page 77.

SelectAll

Label	Select All
Path	Main window: Edit menu
Category	edit
Action	Selects all the objects in the work space.

SelectArcMode

Label	Arc
Path	Palettes Panel: Graphics palette
Category	mode
State	True if this mode is selected.
Action	Selects the Arc mode. See *Arc Mode* on page 84.

SelectArrowLineMode

Label	Arrow Line
Path	Palettes Panel: Graphics palette
Category	mode
State	True if this mode is selected.
Action	Selects the ArrowLine mode. See *Arrow Line Mode* on page 84.

SelectBuffer

Label	Select Buffer…
Category	buffer
Action	Selects a buffer. This command opens a dialog box that allows you to bring one of the currently opened buffers to the front.

SelectClosedSplineMode

Label	ClosedSpline
Path	Palettes Panel: Graphics palette
Category	mode
State	True if this mode is selected.
Action	Selects the ClosedSpline mode. See *Closed Spline Mode* on page 86.

SelectFilledArcMode

Label	FilledArc
Path	Palettes Panel: Graphics palette
Category	mode

State	True if this mode is selected.
Action	Selects the FilledArc mode. See *Filled Arc Mode* on page 85.

SelectFilledSplineMode

Label	FilledSpline
Path	Palettes Panel: Graphics palette
Category	mode
State	True if this mode is selected.
Action	Selects the FilledSpline mode. See *Filled Spline Mode* on page 86.

SelectGadgetContainerRectangleMode

Deprecated - use rectangular object creation mode instead.

Label	GadgetContainerRectangle
Category	mode
State	True if this mode is selected.
Action	Creates a GadgeContainerRectangle

SelectLabelListMode

Label	LabelList
Path	Palettes Panel: More palette
Category	mode
State	True if this mode is selected.
Action	Selects the LabelList mode. See *Label List Mode* on page 77.

SelectLabelMode

Label	Label
Path	Palettes Panel: More palette
Category	mode
State	True if this mode is selected.
Action	Selects the Label mode. See *Label Mode* on page 77.

SelectLineMode

Label	Line
Path	Palettes Panel: Graphics palette
Category	mode
State	True if this mode is selected.
Action	Selects the Line mode. See *Line Mode* on page 84.

SelectObject

Deprecated. Use `SelectSelectionMode` instead.

SelectPlugIns

Label	Select Plug-Ins ...
Path	Tools menu > Select Plug-Ins ...

Category	studio
Action	Activates a dialog box that lets you select the plug-ins you want to use for future sessions of Studio. Studio displays the plug-ins found in your dynamic module path. Check the ones you want to use and click OK. The selected plug-ins will be loaded next time you launch Studio. Note: The IBM ILOG Views Studio plug-ins are based on the IBM ILOG Views dynamic modules. Therefore, this command can only be used on platforms on which the IBM ILOG Views dynamic modules are implemented.

SelectPolySelectionMode

Label	PolySelection
Path	Main window: Editing Modes toolbar (when editing 2D Graphics or Grapher buffers)
Category	mode
State	True if this mode is selected.
Action	Selects the PolySelection mode. See *PolySelection Mode* on page 76.

SelectPolygonMode

Label	Polygon
Path	Palettes Panel: Graphics panel
Category	mode
State	True if this mode is selected.
Action	Selects the Polygon mode. See *Polygon Mode* on page 85.

SelectPolylineMode

Label	Polyline
Path	Palettes Panel: Graphics panel
Category	mode

State	True if this mode is selected.
Action	Selects the Polyline mode. See *Polyline Mode* on page 85.

SelectPreviousBuffer

Label	Select previous buffer
Category	buffer
Action	Lets you select the buffer previously edited in the work space.

SelectRectangularObjectCreationMode

Label	Rectangular Object Creation
Path	Palettes Panel: All objects in the palette that do not have a specific creation mode attached.
Category	mode
State	True if this mode is selected.
Action	Allows the creation of objects delimited by a rectangular bounding box.

SelectReliefLineMode

Label	ReliefLine
Path	Palettes Panel: Graphics
Category	mode
State	True if this mode is selected.
Action	Selects the ReliefLine mode. See *Relief Line Mode* on page 83.

SelectRotateMode

Label	Rotate
Path	Main window: Editing Modes toolbar (when editing 2D Graphics or Grapher buffers)
Category	mode, 2d
State	True if this mode is selected.
Action	Selects the Rotate mode.

SelectSCGadgetContainerRectangleMode

Deprecated - use rectangular creation mode instead.

Label	SCGadgetContainerRectangle
Category	mode
State	True if this mode is selected.
Action	Selects the scrolled gadget container rectangle mode.

SelectSCGrapherRectangleMode

Deprecated - use rectangular creation mode instead.

Label	SCGrapherRectangle
Category	mode
State	True if this mode is selected.
Action	Selects the scrolled grapher rectangle mode.

SelectSCManagerRectangleMode

Deprecated - use rectangular creation mode instead.

Label	SCManagerRectangle
Category	mode

State	True if this mode is selected.
Action	Selects the scrolled manager rectangle mode.

SelectSelectionMode

Label	Selection
Path	Main window: Editing Modes toolbar
Category	mode
State	True if this mode is selected.
Action	Selects the Selection mode. See *Selection Mode* on page 75.

SelectSplineMode

Label	Spline
Path	Palettes Panel: Graphics
Category	mode
State	True if this mode is selected.
Action	Selects the Spline mode. See *Spline Mode* on page 85.

SelectTranslateMode

Label	Translate
Path	Main window: Editing Modes toolbar (when editing 2D Graphics or Grapher buffers)
Category	mode, 2d
Action	Selects the Translate view mode.

SelectUnZoomMode

Label	Unzoom
Path	Main window: Editing Modes toolbar (when editing 2D Graphics or Grapher buffers)
Category	mode, 2d
Action	Selects the Unzoom mode which lets you zoom out of the current buffer.

SelectZoomMode

Label	Zoom
Path	Main window: Editing Modes toolbar (when editing 2D Graphics or Grapher buffers)
Category	mode, 2d
Action	Selects the Zoom mode which lets you zoom into the current buffer.

SetLanguage_en_US

Label	English (US)
Path	Main window: View menu > Options
Category	language, studio
Action	Sets the display current language to en_US.

SetLanguage_fr_FR

Label	French
Path	Main window: View menu > Options

Category	language, studio
Action	Sets the display current language to `fr_FR`.

SetLayer

Label	Layer…
Path	Main window: Draw menu
Category	edit
Action	Opens a dialog box that lets you change the layer number of the selected objects.

SetName

Deprecated. Use the object inspector instead.

Label	Name…
Category	property
Action	Opens a dialog box that lets you give a name to each selected object.

SetObjectInteractor

Deprecated. Use the object inspector instead.

Label	Interactor…
Category	property
Action	Opens a dialog box that lets you attach an object interactor name to each selected object.

SetThickness

Deprecated. Use the object inspector instead.

Label	Thickness…

Category	property
Action	Opens a dialog box that lets you change the thickness of the selected objects.

SetToolTip

Deprecated. Use the object inspector instead.

Label	ToolTip…
Category	property
Action	Opens a dialog box that lets you attach a tooltip to each selected object.

ShowAlignPanel

Label	Alignment
Path	Main window: Draw menu > Align/Distribute
Category	align, panel
Action	Opens the Alignment panel that provides you with alignment commands.

ShowAttachPanel

Deprecated.

Label	2.1 Attachments
Category	panel
Action	Activates the Attachments panel that was used with ILOG Views 2.1 IlvEdit. This panel is part of IBM ILOG Views Studio in order to ensure compatibility with previous versions. We strongly recommend that you use the new Attachments mode instead.

ShowCommandPanel

Label	Commands
Path	Main window: Tools menu and Commands icon in the toolbar

Category	panel
Action	Opens the Command panel that allows you to execute a command.

ShowErrorPanel

Label	Errors
Path	Main window: Tools menu
Category	panel
Action	Opens the Errors panel that lets you see the error, warning, and information message history.

ShowGenericInspector

Label	Generic Inspector
Path	Main window: Tools menu and Generic Inspector in the toolbar
Category	panel
Action	Shows (or hides) the Generic Inspector in the Main window. This command does not work with the Application buffer.

ShowGridPanel

Label	View Inspector
Path	Main window: Tools menu
Category	panel
Action	Opens the Grid panel that lets you set the grid options for the work space, resize the work space, and set the background color of the work space.

ShowGridOptionsPanel

Label	Grid Option ...
Path	Main window: Draw menu
Category	view
Action	Opens the Grid Options panel that lets you specify settings for the drawing grid that can be displayed in the buffer window.

ShowGroupOptionsPanel

Label	Group Into
Path	Main window: Draw menu
Category	group
Action	Lets you specify the type of group to create when you are grouping objects in the buffer window.

ShowInfoPanel

Label	Info...
Path	Main window: Help menu
Category	studio, panel
Action	Shows the IBM ILOG Views Studio information panel.

ShowInspector

Label	Object Inspector
Path	Main window: Draw menu or double-click an object in the work space.

Category	inspector, panel
Action	Opens the inspector panel corresponding to the selected (or double-clicked) object, if applicable. The look and behavior of the inspector panel depend on the class of the inspected object.

ShowLayerPanel

Label	Layer Editor
Path	Main window: Tools menu
Category	panel
Action	Opens the Layers panel, which lets you manage the layers of the current buffer.

ShowMessagePanel

Label	Message Editor
Path	Main window: Tools menu
Category	panel
Action	Opens the Message Editor, which lets you edit multilingual messages and manage the languages of the display.

ShowPalettePanel

Label	Palettes
Path	Main window: Tools menu
Category	panel
Action	Opens the Palettes panels from which you can create graphic objects in the current buffer window.

ShowPropertiesPanel

Label	Property sheet
Path	Tools Menu
Category	Studio, panel
Action	Shows (or hides) the properties panel.

ShowPrinterDialog

Label	Printer...
Path	File Menu
Category	all
Action	This command is available on Microsoft® Windows® platforms only. It lets you edit the printer settings.

TestDocument

Label	Test Application
Path	Main window: tool bar
Category	application
Action	Tests the application if the current buffer is the application or tests the panel data if the current buffer is a panel buffer.

TestPanel

Label	Test
Path	Main window: toolbar

Category	buffer
Action	Tests the current buffer.

TileHorizontally

Label	Tile Horizontally
Path	Main window: Window menu
Category	buffer
Action	Arranges the buffer windows in the work space as horizontal nonoverlapping tiles.

TileVertically

Label	Tile Vertically
Path	Main window: Window menu
Category	buffer
Action	Arranges the buffer windows in the work space as vertical nonoverlapping tiles.

ToggleActiveGrid

Label	Snap to Grid
Path	Main window: View menu
Category	view
State	True if this option is selected.
Action	When moving or resizing an object selected in the buffer window, the movements will snap to the grid.

ToggleForegroundGrid

Label	Foreground Grid
Path	Main window: View menu
Category	view
State	True if this option is selected.
Action	Shows the grid on top of the contents of the buffer window.

ToggleGrid

Label	Grid
Path	Main window: View
Category	view
State	True if this option is selected.
Action	Displays a grid in the current buffer window.

ToggleState

Deprecated. Use the object inspector instead.

Label	State
Category	property
Action	Switches the sensitivity state of the selected objects.

ToggleTransformed

Label	Use Transformer
Path	Main window: Draw menu
Category	edit

State	True if this option is selected.
Action	If an object is in "direct transform mode," places the object inside a transformed graphic. When the object is moved or resized, its coordinates will not risk being degraded. If the object is already inside a transformed graphic, this command removes it from its transformed graphic wrapper and returns it to direct transform mode. In direct transform mode, an object uses less memory.

UnDo

Label	Undo
Path	Main window: Edit menu and Undo icon in the toolbar
Category	edit
Action	Undoes the last executed command.

UnGroup

Label	Ungroup
Path	Main window: Draw menu
Category	group
Action	Ungroups the selected group of objects.

VerticalSpaces

Label	Distribute Vertically
Path	Main window: Draw menu > Align/Distribute; Alignment panel
Category	align
Action	Creates an equal amount of vertical space between the selected objects. The space between two objects is at least equal to a predefined value. Object dimensions are not changed. This command works only if at least two objects are selected.

Windows Look

Label	Windows Look
Path	Main window: View menu > Options
Category	look, studio
Action	Sets the display current look to Microsoft® Windows®.

Windows95Look

Label	Windows 95 Look
Path	Main window: View menu > Options
Category	look, studio
Action	Sets the display current look to Windows 95.

ZoomIn

Label	Zoom In
Path	Main window: View menu
Category	view
State	True if this option is selected.
Action	Zooms in the current view by a factor of 2.

ZoomOut

Label	Zoom out
Path	Main window: View menu
Category	view

State	True if this option is selected.
Action	Zooms out the current view by a factor of 2.

6

Panels

This chapter describes various panels and dialog boxes that you will encounter when using
IBM® ILOG® Views Studio:

◆ *Commands Panel*

◆ *View Inspector (View Options) Panel*

◆ *Alignment Panel*

◆ *Message Editor Panel*

◆ *Grid Options Panel*

◆ *Layers Editor*

◆ *Errors Panel*

◆ *Preferences Panel*

Commands Panel

The Commands panel lets you execute IBM ILOG Views Studio commands. Since the
standard menu and toolbars do not contain all the commands of IBM ILOG Views Studio,
you need this panel to execute commands that are not part of a menu or toolbar.
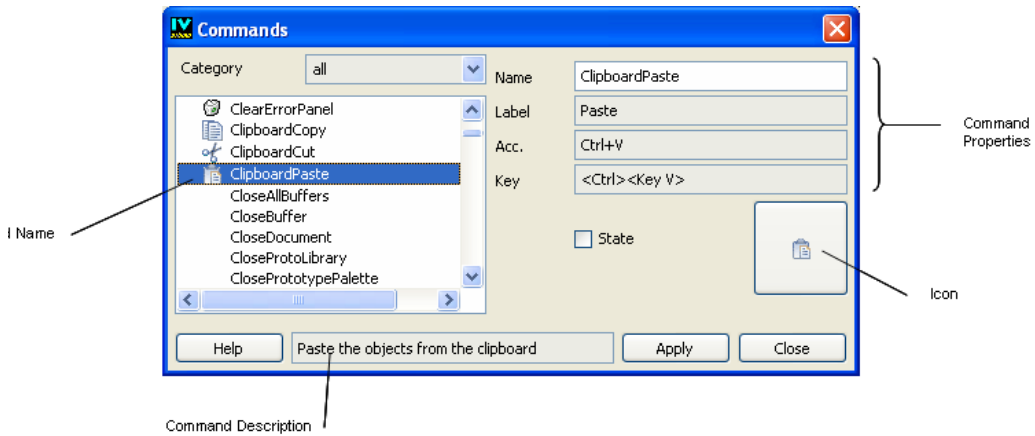
Figure 6.1   Commands Panel

Access to Panel

The panel is accessed by:

◆ Clicking the Commands icon in the Action bar.

Commands Icon

or

◆ Choosing Commands from the Tools menu.

Commands Panel Elements

The Commands panel contains the following elements:

◆ **Category** The IBM ILOG Views Studio commands are listed by category. Use the Category combo box to select the category of commands that are displayed in the Command list box. To see all the available commands, select all in the combo box.

◆ **Commands list** The commands list box shows the commands available in IBM ILOG Views Studio. The commands are listed in alphabetical order. If an icon is associated with a command it appears before the command.

To search for a command in the list, make sure the commands list owns the keyboard focus. Type the capitalized letters of the command you wish to find. For example, to find SelectArrowLineMode, type the following characters: s, a, l, and m. If more than one command can be found with these letters, press the space bar to find the next matching command. To cancel the input to search for another command, press the Escape key.

◆ **Command properties** When you select a command in the Commands list, the command properties are displayed. These properties cannot be modified in this panel.

- **Name** Shows the name of the currently selected command.

- **Label** Shows the information label of the currently selected command.

  If a command is attached to a menu item, the label is used to set the label of the menu item.

  If a command is attached to a toolbar or editing mode button, the label is displayed in a small window when you leave the mouse pointer on the button for a short while.

- **Acc.** Shows the keyboard accelerator that can be used to activate the currently selected command. This is the text that makes up the keyboard accelerator.

- **Key** Shows the actual keys that make up the key board accelerator for the currently selected command.

- **State** Shows the Boolean state of a command. A Boolean state is useful when the command action changes the state of something. For example, commands that select editing modes have their state modified when the associated modes are selected or deselected. When the state of the command is true, the State button appears inverted.

- **Icon** Shows the icon associated with the command. If no icon is associated with the command, this area is blank.

◆ **Command description** Shows a short description of the command. This description is displayed in the message area of the Main window when you highlight a menu item or when you leave the mouse pointer on a button associated with the command.

Executing a Command

To execute a command, select the command from the list of commands and click Apply. You can also double-click the command name or click its icon.

Searching for a Command in the Commands List

The Commands panel provides you with an easy way to find a command.

To search for a command in the list, make sure the commands list owns the keyboard focus. Type the capitalized letters of the command you wish to find. For example, to find `SelectArrowLineMode`, type the following characters: s, a, l, and m. If more than one command can be found with these letters, press the space bar to find the next matching command. To cancel the input to search for another command, press the Escape key.

View Inspector (View Options) Panel

The View Options panel lets you display a grid to help you align objects in the work space, and define its properties (spacing between the grid points for example). Using this panel, you can also change the size and background of the current buffer window.
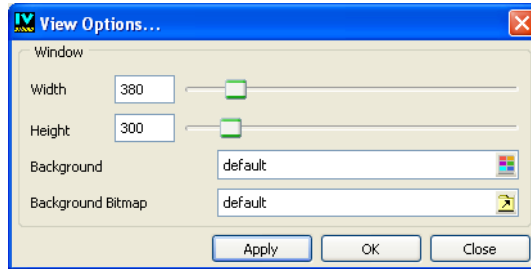
Figure 6.2   View Inspector (View Options) Panel

Access to Panel

The panel is accessed by:

◆ Choosing View Inspector from the Tools menu.

   or

◆ Selecting the `ShowViewOptions` panel in the Commands panel and clicking Apply.

View Inspector Panel Elements

The View Inspector panel contains the following elements:

◆ **Width** Use this field to specify the width of the current buffer window.

◆ **Height** Use this field to specify the height of the current buffer window.

◆ **Background** The box to the right of this label displays the current background color of the current buffer window. To select another color, click the combo selector.

◆ **Background Bitmap** Use this field to specify a bitmap file that can be used as the background for the buffer window.

Alignment Panel

The Alignment panel lets you align objects selected in a buffer window. It allows you access to the same alignment commands that are found in the Align/Distribute submenu of the Draw menu.
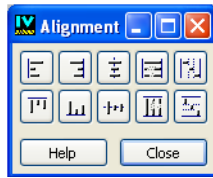
Figure 6.3 *Alignment Panel*

Access to Panel

The panel is accessed by:

◆ Clicking the Alignment icon in the Action toolbar.

Alignment Icon

or

◆ Choosing Align/Distribute from the Draw menu, then choosing Alignment Panel from the submenu that appears.

or

◆ Selecting the `ShowAlignPanel` command in the Commands panel and clicking Apply.

Alignment Panel Elements

The Alignment panel contains the following buttons that allow you to perform various alignment operations on the objects selected in a buffer window. When at least two objects are selected, the first selected object is used as the reference for the other objects. If only one object is selected, it aligns with either the whole panel or with any guides to which the object may be attached. See *Aligning Objects* on page 39 for a complete description of these buttons.

Align Left

Align Right

Center Vertically

Equal Width
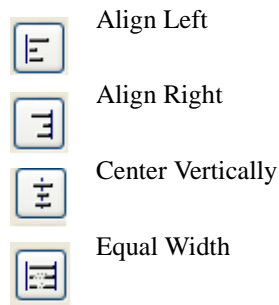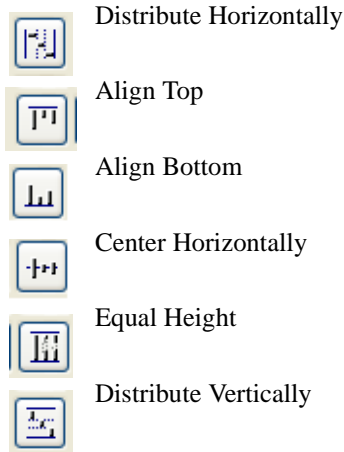
	Distribute Horizontally
	Align Top
	Align Bottom
	Center Horizontally
	Equal Height
	Distribute Vertically

Message Editor Panel

The Message Editor panel lets you edit multilingual messages and save them to database files.
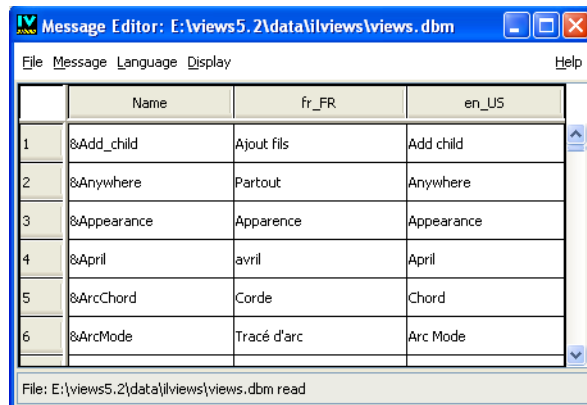
Figure 6.4   Message Editor Panel

Access to Panel

The panel is accessed by:

◆ Choosing Messages Editor from the Tools menu.

or

◆ Selecting the command `ShowMessagePanel` in the Commands panel and clicking Apply.

Message Editor Panel Elements

You use the table that appears in this panel to edit the messages of the database file. The Name column contains the messages identifiers. The definitions for each language are edited in another column. Except for the first line that displays the column headers, each row of the table corresponds to a message.

Editing a Table Cell

To edit a cell of the table in the message editor:

1. Double-click the cell.

2. Type the string of characters required for the message.

3. Press Enter to validate the new string.

Using Panel Menus

At the top of the Messages Editor panel are the various menus that you use to manage message database files.

File Menu

Use the File menu for file maintenance tasks for your message database files.

Menu Item	Description
New	Empties the panel and creates a new database for editing.
Open	Opens a dialog box that lets you select and load a message database file previously saved by the message editor.
Insert	Opens a dialog box that lets you select and insert a file previously saved by the message editor. The messages of this file are appended to the current database.
Save	Saves the current database to a file. If the database has no associated file name, a dialog box opens for providing a filename.
Save As	Opens a dialog box that lets you save the current database in a new file.
Quit	Closes the Messages panel.

Message Menu

Use the Message menu to add, remove, search and sort messages for the database being edited.

Menu Item	Description
New Message	Inserts a new line in the table for defining a new message in the current database. If a cell is selected, the new line is inserted into it. Otherwise, it is appended to the end of the table.
New Messages	Opens a dialog box for you to enter the number of new messages to be inserted.
Remove	If a cell is selected, removes the message corresponding to the selected cell.
Search	Opens a dialog box where a string is typed for which the Message editor searches the next occurrence. You can specify the column where the Message editor searches for the string.
Sort by name	Sorts the database by name.
Sort by definition	Displays a submenu for choosing a language column to sort the database.

Language Menu

Use this menu to add or remove languages.

Menu Item	Description
New Language	Opens a dialog box to enter a new language for the database. When you validate the language, the Message editor appends an empty column to the table for entering the message definitions for the new language.
Remove Language	If a cell is selected, removes the language corresponding to the column containing that cell. This operation removes all the message definitions for that language from the database.

Display Menu

Use this menu to modify the display database.

Menu Item	Description
Reset	Clears the display database and reloads the default databases.
Append	Appends the edited messages to the display database so your new messages are known by the display.
Choose Language	Displays a submenu for choosing the current language for the display.

Grid Options Panel

The Grid Options panel lets you specify the settings for the grid that can be displayed to help you draw and align objects in the current buffer window.
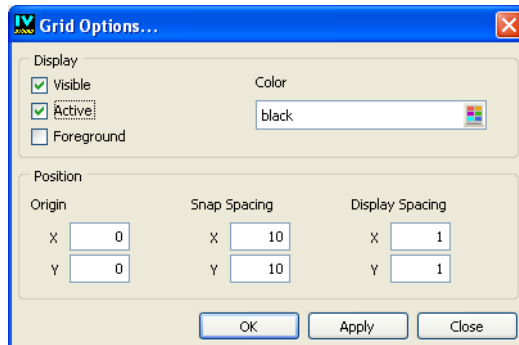
Figure 6.5   Grid Options Panel

Access to Panel

The panel is accessed by:

◆ Choosing Grid Options from the View menu.

   or

◆ Selecting the ShowGridOptionsPanel command in the Commands panel and clicking Apply.

Grid Options Panel Elements

The Grid Options panel contains the following elements:

◆ **Visible** Select this toggle button to show the grid in the current buffer window.

◆ **Active** Select this toggle button to activate the "snap-to-grid" feature. When this option is selected the grid attracts objects to it. As you draw, rotate, resize, or drag objects, they snap to the grid.

◆ **Foreground**   Select this toggle button to have the grid appear in the foreground of the buffer window. You will be able to see the grid on top of the objects in the window.

◆ **Color**   Select this toggle button to display the grid color.

◆ **Origin** Use the x and y fields to change the origin of the grid.

◆ **Snap Spacing** Use the x and y fields to set the spacing between grid points.

◆ **Display Spacing** Use the x and y fields to specify the interval at which the grid points are visible.

To see how the snap spacing settings and the display spacing settings work together, suppose the snap spacing is set to 10,10 and the display spacing is set to 1,1. You will see a point for every snap grid point. You will see your objects snap to each point on the screen as you draw, resize, move them and so on. If you change the display setting to 2,2. You will a point for every second grid point. The objects will still snap to each grid point even though you do not see all the points.

Layers Editor

The Layers editor lets you define different layers of an object. Each object can be grouped and then displayed on specified layers. The layers can be made visible according to a specified order.
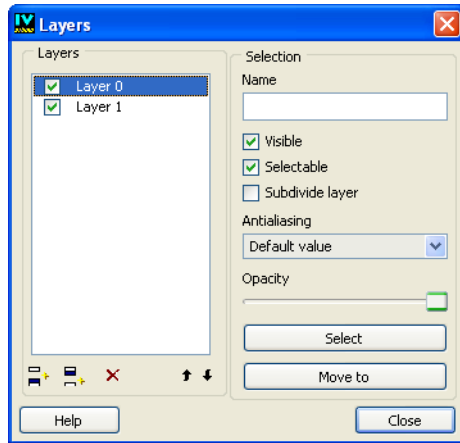
Figure 6.6   Layer Editor Panel

Access to Panel

The panel is accessed by:

◆ Choosing Layer Editor from the Tools menu.

    or

◆ Selecting `ShowLayerPanel` from the Commands panel and clicking Apply.

Layers Editor Elements

The Layers editor panel consists of the Layers area and the Selection area.

◆ **Layers** Add or remove a layer from the object and specify the order of the layers.

◆ **Selection** Change individual properties of a layer.

Errors Panel

The Errors panel allows you to see a list of errors, warnings, or information messages that
have been displayed during your current IBM ILOG Views Studio session.
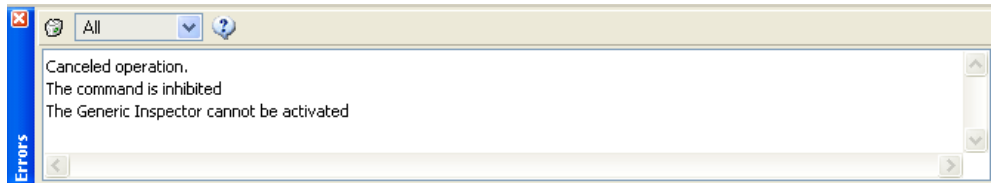
Figure 6.7   Errors Panel

Access to Panel

The panel is accessed by:

◆ Choosing Errors from the Tools menu.

    or

◆ Selecting `ShowErrorPanel` from the Commands panel and clicking Apply.

Errors Panel Elements

The Errors panel contains the following elements:

◆ **Message list box** A list of the messages that have been displayed during your current session shown here. They are displayed in the order they occurred.

◆ **Clean** Use the button to erase all messages from the message list box.

◆ **Type** Use the combo box to choose the type of messages you want to be shown in the list box. Select All, Information, Warning, or Fatal.

◆ **Help** Use the button to call the help.

Preferences Panel

The Edit Options panel allows you to set preferences for your IBM ILOG Views Studio sessions.
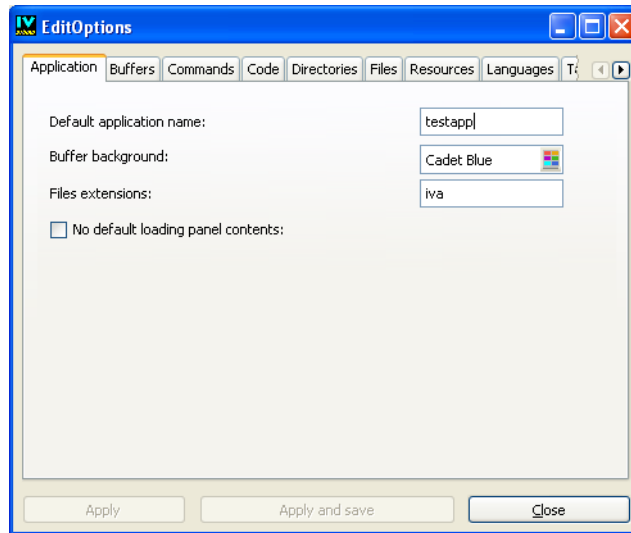
Figure 6.8 *Preferences (EditOptions) Panel*

Access to Panel

The panel is accessed by:

◆ Choosing Preferences from the Tools menu.

   or

◆ Selecting `EditOptions` from the Commands panel and clicking Apply.

Edit Options Panel Elements

The Edit Options panel consists of notebooks pages on which you will set various properties for your IBM ILOG Views Studio sessions.

◆ **Application Page** Set preferences for generating applications, including the default application name, the application buffer window background color, and the default file extension.

◆ **Buffers Page** Set preferences for buffer windows, including the default buffer name, the default buffer window background color, and the buffer window size.

◆ **Commands Page** Set default command options.

◆ **Code Page** Set preferences for generating code, including the default callback language, system name, user subclass prefix, user subclass suffix, and panel base class.

◆ **Directories Page** Set default directories for header files and source files.

◆ **Files Page** Set default file extensions.

◆ **Resources Page** Set default resources, including font names and font foundries.

◆ **Languages Page** Set language preferences, including the multilingual language database to be used and the default language.

◆ **Miscellaneous Page** Set miscellaneous options including displaying the buffer menu, displaying the generic inspector, sorting requirements, the amount of time the startup information panel is displayed, the player speed, and the object interactor button.

7

Customizing IBM ILOG Views Studio

This chapter describes how to customize IBM® ILOG® Views Studio. It contains information on the following topics:

◆ *Configuration Files*

◆ *Resource and Environment Variables*

Configuration Files

IBM® ILOG® Views Studio uses three kinds of configuration files:

◆ Option files, with the `.opt` extension.

◆ Command files, with the `.opt` extension.

◆ Panel files, with the `.pnl` extension.

The Option File

When IBM® ILOG® Views Studio is launched, it reads an option file to set up its properties. You can alter the predefined configuration by using an additional option file and setting its path to the `ILVSTOPTIONFILE` environment variable or the `stOptionFile`

resource. IBM ILOG Views Studio loads your option file after loading the `$ILVHOME/studio/data/studio.opt` file.

When building an IBM ILOG Views Studio extension, you can use the `IlvStudio` member function `addOptionFile(const char*)` before calling the `initialize` function to programmatically add an option file that will be loaded for all the users. Your option file is loaded after the `studio.opt` file and before the option file specified by the final user, if any (with the environment variable or resource).

The option file has the following syntax:

```
Studio {
    <option-1> <value-1>;
    // ...
    <option-n> <value-n>;
}
```

Configuration Options

IBM ILOG Views Studio provides you with the following configuration options:

◆ `alignmentSpacing <size>` lets you specify the value used when aligning objects with a given space distribution. The default value is 5.

◆ `bitmapAlias "<oldName>" "<newName>"` makes IBM ILOG Views Studio use the bitmap `<newName>` instead of `<oldName>`. This option can be used several times.

◆ `bufferBackground "<colorName>"` lets you specify the default background color of the buffer windows when they are created. If this option is not specified, the new buffer has the same background as the current buffer.

◆ `bufferSize <width> x <height>` lets you specify the dimensions of the panels when they are created.

For example:

```
studio {
    bufferSize 400 x 200;
}
```

If this option is not specified, the new buffer has the same size as the current buffer.

◆ `command <commandDeclaration>` lets you directly declare a command in the `.opt` file. See the section *IBM ILOG Views Studio Command File* on page 162 for the syntax of command declarations. This option can be used several times.

For example:

```
studio {
   command ShowScriptEditor {
      label   "&scriptEditor";
      prompt  "&scriptEditorP";
      bitmap  "icsced.gif";
      category script;
```

```
}
```

◆ `commandFile "<cmdFile>"` lets you specify a command declaration file (`.cmd`) file that IBM ILOG Views Studio loads after loading the `studio.cmd` file provided. The `ILVPATH` environment variable is used to locate the specified file. This option can be repeated.

For example:
```
studio {
    commandFile "mystudio.cmd";
}
```

◆ `dataFileExtension "<extension>"` lets you specify the IBM ILOG Views data file extension. The default value is `.ilv`.

◆ `defaultBufferName <name>` lets you specify the name of a new buffer. The default value is `noname`.

For example:
```
studio {
    defaultBufferName newBuffer;
}
```

◆ `defaultDragDropPalette "<palette name>"` lets you specify the palette that is selected by default when the Palettes panel is initialized.

◆ The following option lets you add a palette:
```
dragDropPalette "<palette name>"
{ <option-1 <value-1>;
  ...
  <option-n <value-n>;
}
```

`<palette name>` is the name of the new palette. It must be unique.

The following is a list of options that can be used in the palette description:

● `dataFileName "<filename>"` specifies the file name of the palette data.

● `label "<label>"` specifies the label of the palette in the tree gadget.

● `bitmap "<bitmap>"` specifies the name of the bitmap representing the palette in the tree gadget.

● `path "<palette1>" [... "<paletteN>"] [-before "<paletteX>"]` specifies the location of the palette in the tree gadget. `palette1 ... paletteN` are the names of the parent nodes, and `palette1` is the root node. If you want your palette to be inserted before another palette, use the `-before` keyword followed by the name of the palette. If the path is not specified, the new node is added as a root node in the tree gadget.

- ● `containerClass "<class>"` specifies the class name of the container to be used. The default value is `"IlvGadgetContainer"`.

- ● `containerWidth <width>` specifies the width of the container used when creating the palette container.

- ● `containerHeight <height>` specifies the height of the container used when creating the palette container.

◆ `editionName "<name>"` lets you specify the name of your IBM ILOG Views Studio edition. Use this option to identify the specialization of your IBM ILOG Views Studio extension. The specified name appears in the Studio Information Panel.

For example:

```
studio {
    editionName "ILOG Script Edition";
}
```

◆ `fileBrowserType "<filter>" "<message>"` lets you add a file type to the IBM ILOG Views Studio file browser. This option can be repeated. Use this option if you extend IBM ILOG Views Studio to load your own file type. See also the `fileLoader` option.

For example:

```
studio {
    fileBrowserType "*.js" "&scriptFilesFilter";
}
```

◆ `fileLoader "<extension>" "<commandName>"` lets you specify the command to be executed to load a file that has a given extension. This option can be repeated. See also the `fileBrowserType` option. For example:

```
studio {
    fileLoader "js" "LoadScript";
}
```

◆ `fontFoundries <foundryList>` lets you customize the font foundries listed by the Font Selector. Note that the font foundries only apply to X11 platforms.

For example:

```
studio {
    fontFoundries {
        "sun",
        "dec",
        "sony",
        "myFoundry"
    }
}
```

◆ `fontNames <fontList>` lets you customize the preferred font names used by the Resource panel.

For example:

```
studio {
    fontNames {
        "fixed",
        "normal",
        "myfont"
    }
}
```

◆ `hideGenericInspector <true/false>` lets you hide the Generic inspector when the editor is launched. This option defaults to `false`.

◆ `ignoringSize <true/false>` lets you save the data file without saving the size of the gadget buffers. This option defaults to false. Use this option if you do not want the size of the buffer to be saved in the data file.

◆ `ignoringBackground <true/false>` lets you save the data file without saving the background color of the gadget buffers. This option defaults to `false`. Use this option if you want your panels to have the default background color.

◆ `include "<fileName>"` lets you load another option file in your `.opt` file. The specified file is searched for in `ILVPATH`.

For example:

```
studio {
    include "myfile.opt";
}
```

◆ `infoPanelDuration <seconds>` lets you specify the period of time (in seconds) during which the Information panel is displayed when the editor is launched. The default value is `5` seconds.

For example:

```
studio {
    infoPanelDuration 10;
}
```

◆ `infoPanelTitle "<title>"` lets you specify the title of the Information panel.

◆ `language "<languageName>" "<commandLabel>" "<commandPrompt>"` lets you define a command selecting `<languageName>` as the current language of the `IlvDisplay`. This option can be repeated. The name of the defined command is `SetLanguage_<languageName>`, and its label and prompt are specified by `<commandLabel>` and `<commandPrompt>` respectively.

For example:

```
studio {
    language "en_US" "English (U.S.)" "Speak English (U.S.)";
}
```

defines the following command:

```
command SetLanguage_en_US {
    label  "English (U.S.)";
    prompt "Speak English (U.S.)";
}
```

Use this option if you provide a message database for `<languageName>`.

◆ `messageDB "<dbname>"` lets you load a multilingual message database specified by `<dbname>`.

For example:

```
studio {
    messageDB "mymessages.dbm";
}
```

◆ `movingPointer <true/false>` lets you specify whether you want the editor to move the mouse pointer to the center of a newly opened window (`true`) or not (`false`).

For example:

```
studio {
    movingPointer true;
}
```

◆ `objectInteractor <interactorName>` lets you add an interactor name to the predefined interactor list used by the Interactor dialog box (activated by the `SetObjectInteractor` command). Use this option if you want to add an object interactor to IBM ILOG Views Studio.

For example:

```
studio {
    objectInteractor MyInteractor;
}
```

◆ `panel <panelDescription>` lets you directly specify panel properties in the `.opt` file. This option can be repeated. See *IBM ILOG Views Studio Panel Description File* on page 164 for the syntax of panel description. Unlike the panel descriptions you specify in a separate `.pnl` file, the panel description you provide in the `.opt` file is loaded before the provided `studio.pnl` file.

For example:

```
studio {
    panel "Classes" {
        topView true;
    }
}
```

◆ `panelFile "<pnlFile>"` lets you specify the panel description file (`.pnl`) file that IBM ILOG Views Studio will load after the `studio.pnl` file provided has been loaded. The `ILVPATH` environment variable is used to locate the specified file. This option can be repeated.

◆ `playerSpeed <milliseconds>` lets you specify the event player speed in milliseconds. The default value is `500`.

◆ `recentFileListLength <length>` lets you specify the length of the recent file list that appears in the File menu. This option defaults to `6`.

◆ `removeDragDropPalette "<palette name>"` lets you remove a predefined palette from the tree gadget in the Palettes panel.

◆ `removeToolBarItem <commandName> <toolBarName>` lets you remove a command from a named tool bar. Use this option if you want to hide an icon in a tool bar defined in the delivered `studio.opt` file.

   For example:

```
studio {
    removeToolBarItem SelectSpinBoxMode IlvStGadgetBuffer;
}
```

◆ `sortingRequirements <true/false>` sorts, if `true`, the state requirements in the State Inspector.

◆ `startUpCommand <commandName>` lets you specify the command that will be executed when the editor is launched. This option can be repeated.

   For example:

```
studio {
    startUpCommand EditStates;
}
```

◆ `studioName "<name>"` lets you customize the name of your IBM ILOG Views Studio extension, saved in the `.iva` file. The default string is `"ILOG Views Studio"`.

   For example:

```
studio {
    studioName "My Studio Extension";
}
```

◆ `studioShortName "<name>"` lets you change the "short name" of the editor. This name is displayed in the title bar of the Main window, before the application name. The default string is `"ivstudio"`.

   For example:

```
studio {
    studioShortName "mystudio";
}
```

◆ `toolBarItem <commandName> <toolBarName> [-before <refCommandName>]` lets you add a command `<commandName>` in the tool bar `<toolBarName>`. This option can be repeated. Optionally, you can specify a command `<refCommandName>` before which you want to insert the new command by using the keyword `-before`.

For example:

```
studio {
    toolBarItem SelectLabelMode IlvStGadgetBuffer -before
SelectFocusMode;
}
```

◆ `toolBarCommands <name> <commands>` lets you define a new tool bar. The defined
  tool bar can then be used by any panel.

◆ `toolBarItemHeight <height>` lets you specify the height of the tool bar items. Use
  this option if you want to use bigger icons. See also `toolBarItemWidth`.

◆ `toolBarItemWidth <width>` lets you specify the width of the tool bar items. Use this
  option if you want to use bigger icons. See also `toolBarItemHeight`.

Example

The following option file sets the default directories for the generated files and adds three
editing modes to the Gadgets buffer window tool bar.

```
// /usr/smith/mystudio.opt: My configuration file.
studio {
    defaultHeaderDir    "include";
    defaultSrcDir       "src";
    defaultObjDir       "obj";
    defaultHeaderFileScope  "myinclude/";
    toolBarItem SelectLabelMode
        IlvStGadgetBuffer -before SelectActiveMode;
    toolBarItem SelectLabelListMode
        IlvStGadgetBuffer -before SelectActiveMode;
    toolBarItem SelectReliefLineMode
        IlvStGadgetBuffer -before SelectActiveMode;
}
```

If the file `mystudio.opt` file is saved in the directory `/usr/smith`, the
`ILVSTOPTIONFILE` environment variable must be set to the access path to this file, as
shown below:

On Microsoft® Windows® platforms:

```
set ILVSTOPTIONFILE=/usr/smith/mystudio.opt
```

On UNIX® platforms:

```
setenv ILVSTOPTIONFILE /usr/smith/mystudio.opt
```

IBM ILOG Views Studio Command File

The predefined IBM® ILOG® Views Studio commands are declared in a command
description file: `studio.cmd`. This file is located in the directory `$ILVHOME/studio/
data`. You can declare your own commands directly in an `.opt` file or in a `.cmd` file and use
the `commandFile` option in the `.opt` file to declare that file. Your command declaration file
will be loaded after the supplied `studio.cmd` file. In your command declaration file, you

can override declarations of predefined commands in order, for example, to modify labels or icons.

Command Properties

An IBM ILOG Views Studio command is defined by the following set of properties:

◆ `label "<label>"` lets you specify the label of the menu item when the command is attached to a menu. If the label is not specified, the command name is used. The specified label can be either a real string or a multilingual message identifier.

◆ `toggleLabel "<label>"` is used for a command having a state, such as the `TestDocument` command. When the command is activated, this string becomes the label of the command tooltip. The specified label can be either a real string or a multilingual message identifier.

◆ `tooltip "<string>"` is used to specify a string used to display the tooltip when the command is attached to a tool bar item or a gadget. By default, the tooltip simply displays the command label. The specified string can be either a real string or a multilingual message identifier.

◆ `toggleTooltip "<string>"` is used for a command having a state, such as the `TestDocument` command. When the command is activated, the command tooltip displays this string. The specified string can be either a real string or a multilingual message identifier.

◆ `acceleratorText "<string>"` is used when the command is attached to a menu item. Used to set up the menu item, it represents the accelerator key that executes the command.

◆ `acceleratorDefinition "<string>"` is used when the command is attached to a menu item. The specified string is decoded and set to the menu item.

◆ `bitmap "<name>"` is used when the command is attached to a tool bar. It lets you specify a bitmap name for the corresponding tool bar item.

◆ `prompt "<string>"` lets you specify a single line string describing your command. The specified string can be either a real string or a multilingual message identifier. This string is displayed in the message area when the command menu item is highlighted or when its tooltip is activated.

◆ `togglePrompt "<string>"` is used for a command having a state. It lets you specify a string to be used instead of the `prompt` string when the command is turned on.

◆ `messages <messageList>` lets you specify one or more IBM ILOG Views Studio messages to be broadcast when the command is executed.

◆ `interactive <true/false>` tells whether the command is to appear in the Commands panel or not.

◆ `category <categoryName>` lets you specify a category for the command. To specify more than one category for a command, use this option repeatedly. If the category you specify is not yet used, it is automatically created and displayed in the Commands panel.

◆ `selector "<name>"` lets you specify a mutually exclusive selector for your command. This property is meaningful when the command can have a state. Only one command in the same selector can be selected at a time. If a command is selected, the previously selected command is automatically deselected.

Example

The following command file changes the label of the Quit command to Exit and defines an accelerator key for executing the CloseDocument command.

The corresponding command declarations are copied from the `$ILVHOME/studio/data/studio.cmd` file to `/users/smith/mystudio.cmd`.

```
command Quit {
    label    "^Exit"; // modified property
    tooltip   "&ttmenu_quit";
    prompt   "&quitEditor";
    bitmap   "icquit";
    category studio;
}

command CloseDocument{
    label "&mclose";
    prompt "&closeDocumentP";
    acceleratorText "F4";                // added property
    acceleratorDefinition "<Key F4>"; // added property
    category document;
}
```

The `/users/smith/mystudio.opt` file declares the above command file:

```
studio {
    commandFile "/users/smith/mystudio.cmd";
}
```

If the `/users/smith` directory is in the `ILVPATH` environment variable, the full path of the `mystudio.cmd` file does not need to be specified:

```
studio {
    commandFile "mystudio.cmd";
}
```

The `ILVSTOPTIONFILE` is set to `/users/smith/mystudio.opt`.

IBM ILOG Views Studio Panel Description File

The `$ILVHOME/studio/data/studio.pnl` file provides you with the default descriptions of the panels supplied. You can alter these descriptions and add your own panel descriptions by using an additional panel description file (`.pnl`). To make IBM ILOG Views

Studio read your `.pnl` file, declare it in a studio option file. For more information, see *The Option File* on page 155.

Panel Properties

An IBM ILOG Views Studio panel is configured by the following set of properties:

◆ `title "<title>"` specifies the panel title. By default, the panel name is used.

◆ `visible <true/false>` indicates whether the panel is displayed when the editor is launched.

◆ `menu "<menuBarItem>" [,"<menuItem1> ...] { <menuItemList> }` lets you set a menu to the panel. This option can be repeated. It only works if the panel contains a menu bar (its `getMenuBar` function returns a menu bar or its container contains a menu bar named `MenuBar`). The menu location is specified by a menu bar item, and, if it is a submenu, it is followed by the list of the menu items leading to the new menu position. A menu item is identified by its label or its command name. Each item of `<menuItemList>` can be a command name, a label for a submenu, or the '-' character for adding a separator. For example, the File > New menu of the Main window is defined by the following specification:

```
panel "MainPanel" {
    menu "&mfile", "&new" {
        NewGadgetBuffer,
        NewGraphicBuffer
        NewGrapherBuffer,
        -,                  // a separator.
        NewApplication,
        MakeDefaultApplication
    }
}
```

◆ `menuItem <itemSpec> : "<menuBarItem>" [,"<menuItem1> ...] [-before <refItem>]` is used to insert an item in the specified menu. This option can be repeated. The new item specified by `<itemSpec>` can be a command name, a label or the '-' character (for a separator). The menu location is specified by a menu bar item, and, if it is a submenu, it is followed by the list of the menu items leading to the target menu position. You can insert the new item before a `<refItem>` by using the `-before` keyword; otherwise, the item is added to the end of the menu.

◆ `removeMenuItem <commandName> : "<menuBarItem>" [,"<menuItem1> ...]` lets you remove a menu item previously specified by a configuration file. The `<commandName>` parameter is the name of the command attached to the menu item to be removed. The menu location is specified by a menu bar item, and, if it is a submenu, it is followed by the list of the menu items leading to the target menu position.

◆ `toolbar <location> { <toolBarItemList> }` lets you add a tool bar to the panel. This option can be repeated. The `<location>` parameter can be either `left`, `top`, `right`, or `bottom`. There can be several tool bars at each location. `<toolBarItemList>` is a list of command names or tool bar names (defined in a `.opt` file) separated by a comma (,).

For example:
```
panel "MainPanel" {
    toolbar right {
        SelectLabelMode,
        SelectLabelListMode,
        SelectReliefLineMode
    }
}
```

◆ `x`, `y`, `width`, and `height` let you specify the position and the size of the panel.

◆ `command "<objectName>" <commandName>` lets you attach a command to a named gadget contained in your panel. For example, if your panel contains a button named `"Open"`:
```
panel "MyPanel" {
    command "Open" OpenDocument;
}
```

◆ `filename "<filename>"` lets you specify a data file (`.ilv`) for your panel. The specified file must be found in `ILVPATH`. It will be loaded by your panel. For example, the Alignment panel is created entirely without coding. It uses the `alignmt.ilv` data file and attaches commands to its buttons:
```
panel "Alignment" {
    commandName ShowAlignPanel;// the command that shows the panel
    filename "alignmt.ilv";          // load that data file
    x 500;
    y 100;
    command "left"      AlignLeft;
    command "right"     AlignRight;
    command "vertical"  AlignVertical;
    command "samew"     SameWidth;
    command "hspace"    HorizontalSpaces;
    command "top"       AlignTop;
    command "bottom"    AlignBottom;
    command "horizontal" AlignHorizontal;
    command "sameh"     SameHeight;
    command "vspace"    VerticalSpaces;
}
```

◆ `commandName <commandName>` lets you specify the name of the command that shows or hides your panel. This is necessary to maintain the command state according to the panel visibility, unless the command name corresponding to your panel is named `Show<PanelName>`.

◆ `topView <true/false>` is used by certain particular panels that are created by default as panes in the Main window, like the Classes palette:

```
panel "Classes" {
   topView true;
}
```

Example

The following panel file, `/users/smith/mystudio.pnl`, displays the Commands palette instead of the Palettes panel when the IBM ILOG Views Studio is launched.

```
panel "Commands" {
    visible true;
}

panel "PalettePanel" {
    visible false;
}
```

The `/users/smith/mystudio.opt` file declares the above panel file:

```
studio {
    panelFile "/users/smith/mystudio.pnl";
}
```

The following panel creates a panel containing editing modes without coding:

```
panel "MyEditingModes" {
   title "Modes";            // Set the panel title
   visible true;             // visible when the editor is launched
   x 300;                    // the panel horizontal position
   y 300;                    // the panel vertical position
   width  150;               // the panel width
   height 150;               // the panel height
   toolbar top {             // first tool bar
       SelectSelectionMode,
       SelectPolylineMode,
       SelectLabelMode,
       SelectLabelListMode
   }
   toolbar top {             // second tool bar
       SelectFocusMode,
       SelectAttachmentsMode,
       SelectMenuMode,
       SelectMatrixMode
   }
   toolbar top {             // third tool bar
       SelectLineMode,
       SelectReliefLineMode,
       SelectArrowLineMode,
       SelectPolylineMode
   }
}
```

Resource and Environment Variables

IBM® ILOG® Views Studio sets up its own resources and environment variables in addition to those used by IBM ILOG Views.

Resources

IBM ILOG Views Studio uses the following resources:

◆ `menuBackground` specifies the background color of IBM ILOG Views Studio menus.

◆ `menuForeground` specifies the foreground color of IBM ILOG Views Studio menus.

◆ `menuFont` specifies the font of IBM ILOG Views Studio menus.

◆ `panelBackground` specifies the background color of IBM ILOG Views Studio panels.

◆ `toolBarBackground` specifies the background color of IBM ILOG Views Studio tool bars.

◆ `focusPathColor` specifies the color of the lines drawing the focus path when using the Focus editing mode.

◆ `attachmentColor` specifies the color of the lines delimiting the attachment guide when using the Attachments mode.

◆ `attachmentHighlightColor` specifies the color of the selected item when using the Attachments mode.

Environment Variables

IBM ILOG Views Studio uses the following environment variables:

◆ `ILVSTOPTIONFILE` specifies the user-defined option file to be loaded. The editor uses the `ILVPATH` environment variable to locate this file.

◆ `ILVSTIGNORINGBACKGROUND=true` specifies whether the Gadget buffers should be saved without saving the background color. If this variable is set to `true`, the panels in your application will use the default background color.

◆ `ILVSTIGNORINGSIZE=true` indicates whether the Gadget buffers should be saved without saving their dimensions.

◆ `ILVPRINTERCOMMANDLINE` is used on UNIX® platforms only to specify the command line to be executed when the editor prints the buffer. The default value is `"lpr -v %s"`. The specified string must be a valid `C printf` format, where `%s` is the name of the file to be printed.

◆ `ILVSTPLUGINS` indicates a list of plug-ins you want to load. You can specify the full paths or the paths relative to the working directory, separated by a semicolon (;) character.

Index