IBM ILOG Views

Data Access V5.3

Getting Started

June 2009

Copyright notice

Trademarks

Notices

For further information see <*installdir*>/license/notices.txt in the installed product.

Table of Contents

About This Manual

Welcome to IBM® ILOG Views, referred to as Data Access, a library dedicated to the development of client-server database applications. Data Access is fully integrated with IBM ILOG Views, therefore allowing you to build graphical user interfaces and to link them to data sources to provide intuitive data.

You will find in this manual a step-by-step tutorial which demonstrates how to build a typical SQL database application.

Quick Start

If you wish to immediately begin the tutorial, go to Chapter 2, *Introduction*.

What You Need to Know

The guide assumes that you are familiar with the UNIX or PC environment in which you are going to use Data Access, including its specific windowing system. Since Data Access is written for C++ developers, this guide 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. A basic knowledge of database and SQL concepts is also required.

Finally, as this product is an add-on to IBM® ILOG Views Controls, you must be familiar with how to use IBM ILOG Views Controls and its graphical editor IBM ILOG Views Studio.

Manual Organization

This manual consists of a table of contents, preface, glossary, six chapters and three appendixes:

- Chapter 1, *Overview* gives an overview of how Data Access works to help you create database applications that run in an SQL environment.
- Chapter 2, *Introduction* introduces you to the tutorial that is designed for you to learn the basic functionality of Data Access.
- Chapter 3, *Database Features* guides you step-by-step as you create the database part of the first window in the example application.
- Chapter 4, *Designing Windows* continues with the first window by leading you through the procedures for making your window exactly as you want it by adding color, aligning objects, resizing the window, and so forth.
- Chapter 5, *Additional Features* shows you other features of Data Access as you create three other small windows to complete the application.
- Chapter 6, *Nested Tables* describes how to use the nested tables feature of Data Access.
- Appendix A, *Setting Up Data Access* provides installation information.
- Appendix B, *Format Syntax* provides the symbols and formulas you can use to create formats for displaying data.
- Appendix C, *SQL Schema Editor* shows how to use the Data Access Schema Editor to create, delete, or edit tables in a database.

Notation

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 put between carets in `courier: <file name>`. (The user should not type the carets.)

Related Documentation and Bibliography

Certain IBM ILOG manuals can help you get started with Data Access, while various books found in the marketplace can be a good source of information to create SQL database applications.

IBM ILOG Manuals

These IBM ILOG manuals can help you use Data Access:

- The *IBM ILOG Views Data Access Reference Manual* describes the C++ classes used in Data Access.
- For more help in using the graphical user interface and a description of IBM® ILOG Views Studio, see the *IBM ILOG Views Studio User's Manual* provided with IBM ILOG Views.
- The *IBM ILOG Views User's Manual* provides helpful information and numerous examples to help you quickly get proficient in the use of IBM ILOG Views.
- The *IBM ILOG Views Reference Manual* provides information on the IBM ILOG Views classes that are referred to in this manual.
- For information on IBM ILOG Views DB Link, see the *IBM ILOG Views DB Link Reference Manual*.

C++ Programming Language Publications

The following books provide information on the C++ programming language:

- Lippman, Stanley B. *C++ Primer*. Reading, MA: Addison-Wesley, 1989.
- Stroustrup, Bjarne. *The C++ Programming Language*. Reading, MA: Addison-Wesley, 1986.
- Stroustrup, Bjarne. *The Design and Evolution of C++*. Reading, MA: Addison-Wesley, 1994.

Database Publications

These books provide helpful information on databases in general:

- Date, C.J. *A Guide to the SQL Standard.* Reading, MA: Addison Wesley.
- Date, C.J. *An Introduction to Database Systems*. Reading, MA: Addison Wesley.

CHAPTER

1

Overview

This chapter introduces you to the basic components of Data Access, the SQL database GUI editor. It also provides the general procedures you need to follow to create panels and applications using Data Access.

You can find information on the following topics:

- *What is Data Access*
- *Architecture*
- *Connections*
- *Data Access Window*
- *General Procedures*

> ***Note:** This manual uses procedures and screen captures made with Data Access using Windows 95 look and feel running the Windows version on a PC. If you are using another platform or system, there might be slight differences in the appearance of the screens or in some naming conventions. Otherwise, the procedures are essentially the same.*

What is Data Access

Data Access is an IBM® ILOG Views graphic-based environment that is dedicated to building SQL database client-server applications. Through the use of data-source-aware gadgets and other features, Data Access greatly reduces coding requirements. Basically, Data Access internally creates the SQL statements based upon the screen operations you perform and then sends these statements to your database management system (DBMS). The DBMS then implements the code in the database, with the resulting data being displayed in the GUI you have built.

The heart of Data Access is the *data source*, which allows you to create tables in a local data cache based on values taken from the database tables or other sources. You can then perform operations with the data with a user-friendly graphical user interface to provide selection and display criteria.

Through the IBM ILOG Views Studio GUI editor, Data Access supplies various data-source-aware gadgets for displaying data and provides powerful application-building and panel-creation features. By making use of its C++ class library—and the extensive library of IBM ILOG Views—Data Access provides you with scores of objects you can use with your database applications as well as the means for their handling and display.

Architecture

The architecture of Data Access can be viewed from two viewpoints:

- Externally—the interaction of Data Access with the environment in which it is situated.
- Internally—the interaction between the components making up Data Access to retrieve, edit, and display data from a database or from other sources.

External Environment

Data Access is integrated into IBM® ILOG Views Component Suite, thus profiting from all of the IBM ILOG Views features and from its extensive libraries. In particular, it makes use of the IBM ILOG Views Studio GUI editor by adding palettes of database-specific gadgets with predefined functionality.

A simple database schema editor is also furnished to help you get started with the examples in this manual. For your own applications, you should refer to your DBMS documentation to find the right tool for building database tables.

Figure 1.1 shows how Data Access is situated in relation to IBM ILOG Views, to the database application, and to the database.

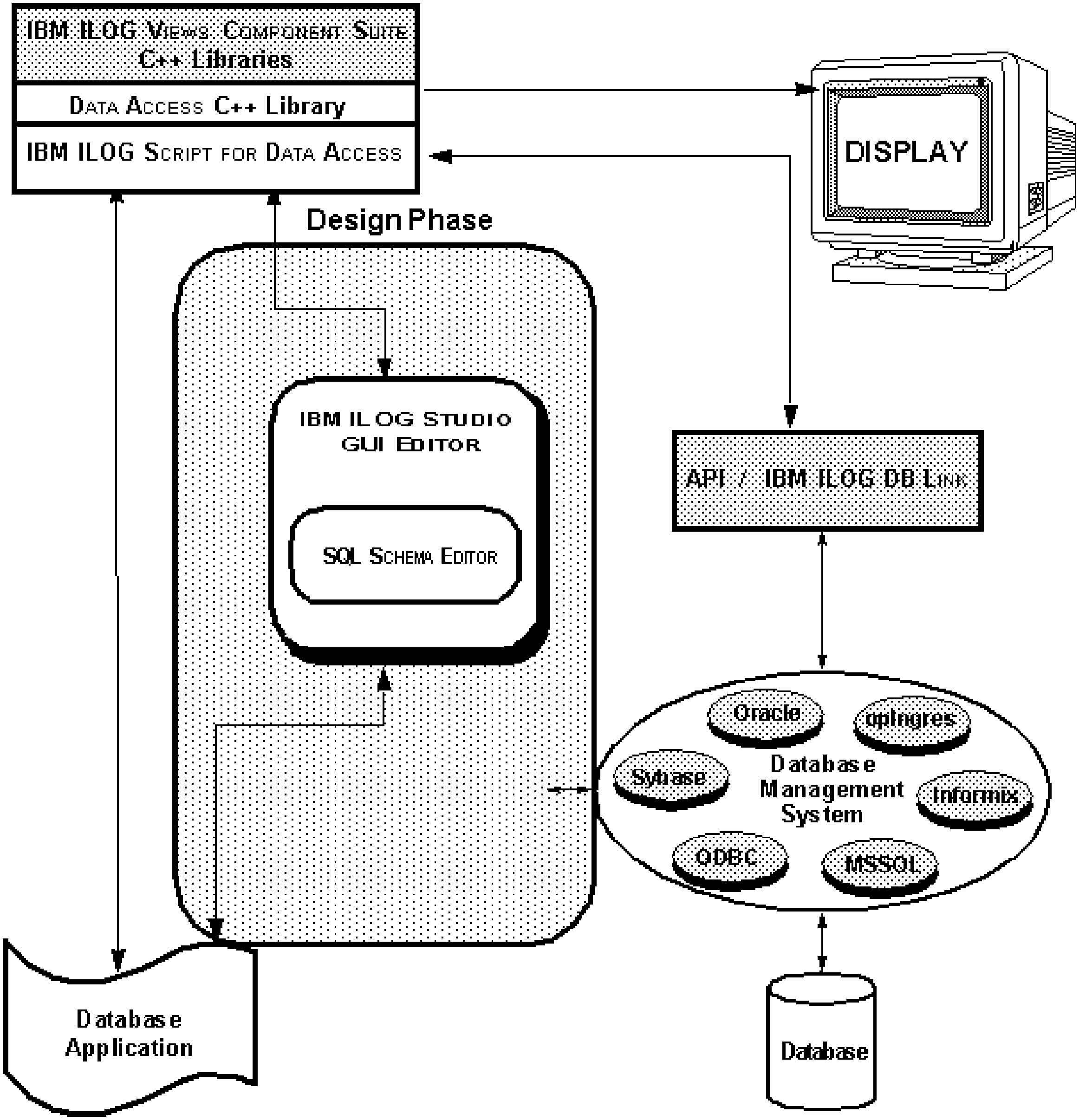


Figure 1.1 *Data Access Environment*

Internal Components

This section introduces the three internal components of Data Access:

- *Palettes Panel*
- *Data Source*
- *SQL Schema Editor*

Palettes Panel

The Palettes panel contains the Data Access gadgets especially designed for Data Access applications and is entirely integrated into IBM® ILOG Views. When you launch Data Access with `dbstudio` or dynamic launching via `ivfstudio`, the Palettes panel appears with the Main window. The command Palettes in the Tools menu opens the Palettes panel if it is closed.

You can also access the IBM ILOG Views gadgets from the Palettes panel. These gadgets—such as buttons, text fields, notebook pages, and many more—while not specific to database applications, can also be used in your interface panels.

You can also use the graphics capabilities of IBM ILOG Views Studio to enhance your application. All the functionality of Data Access can be accessed from the gadgets in the Palettes panel (except the Data Access SQL Schema Editor). For a description of these palettes, see *Palettes Panel* on page 30.

Data Source

Data Access makes use of a *data source,* which defines how values are taken from the database, how the database is updated, and how values are to be displayed using Data Access gadgets. The data source uses a *data cache* to store data in local memory as a *data source table.* A distinction should be made among:

- the *tables* that make up the database. These can be created by the Data Access SQL Schema Editor.
- the *data-source table* stored in the data source data cache. This is a result of computations made by the data source from the user's specifications made in the SQL Data Source inspector panel (see Figure 1.4 on page 19).
- the Data Access *table gadgets.* These are connected to the data source and used for displaying data in table form.

The following diagram shows how the data source is related to both the database tables and the table gadgets:

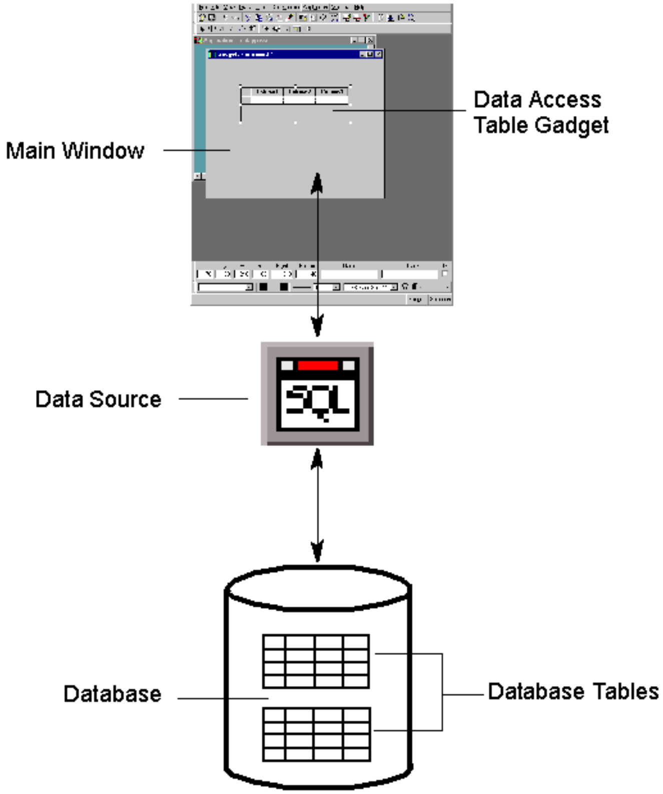
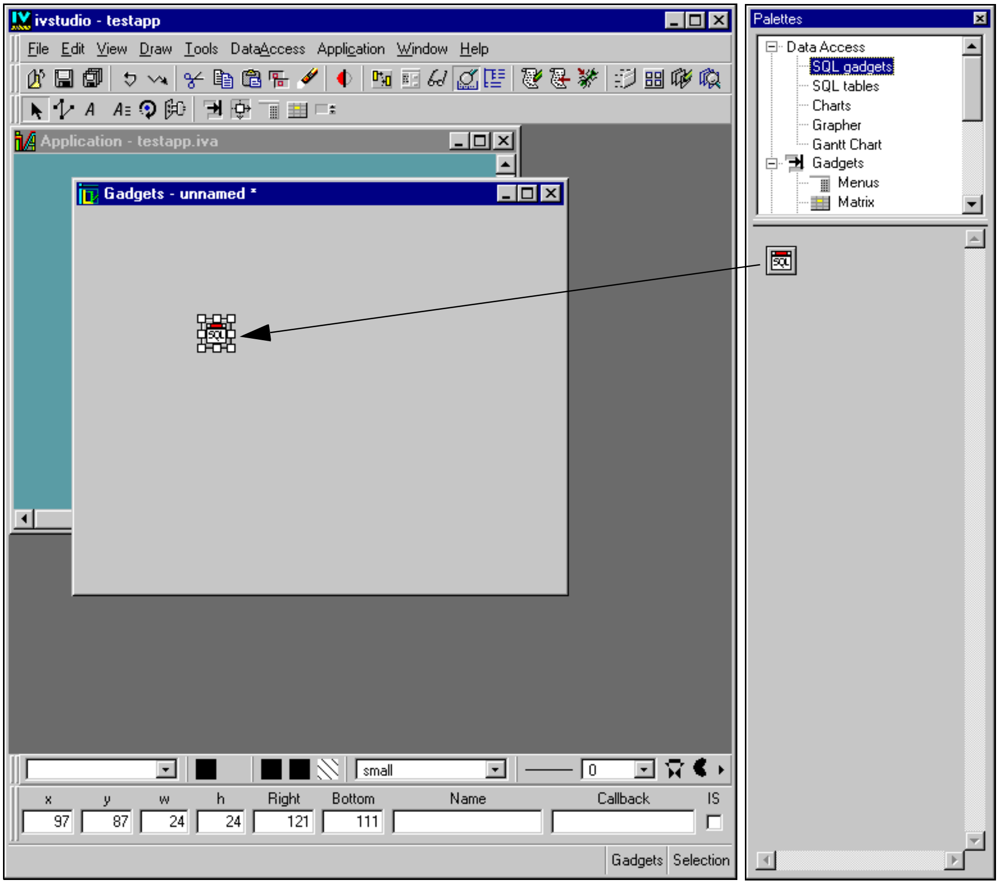


Figure 1.2 Data Source Functionality

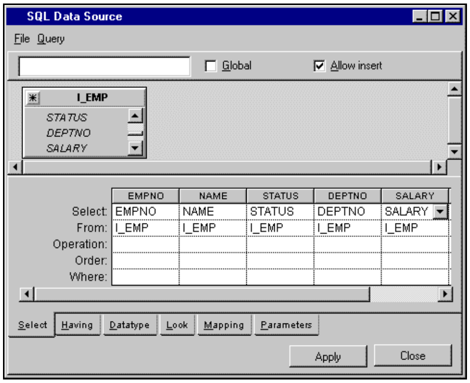
When you drag a SQL Data Source gadget from the Palettes panel to the Main window the SQL Data Source icon appears on the work space of the Main Window:



x	y	w	h	Right	Bottom	Name	Callback	IS
97	87	24	24	121	111			□

__Figure 1.3__ Accessing the Data Source

By double-clicking the icon on the work space, the SQL Data Source inspector panel appears:



□ Global

☑ Allow insert

	EMPNO	NAME	STATUS	DEPTNO	SALARY
Select:	EMPNO	NAME	STATUS	DEPTNO	SALARY
From:	I_EMP	I_EMP	I_EMP	I_EMP	I_EMP
Operation:					
Order:					
Where:					

Figure 1.4 *SQL Data Source Inspector Panel*

This panel has two main sections:

- **FROM section**—the area where you place representations of the database tables from which the data source takes its data.
- **SELECT section**—the area that allows you to define a data source table. To do this, the data source applies your selection and your display criteria to the values taken from the database tables represented in the FROM section. Using these criteria, the data source computes a table that it places in the data cache for later display or editing of the database.

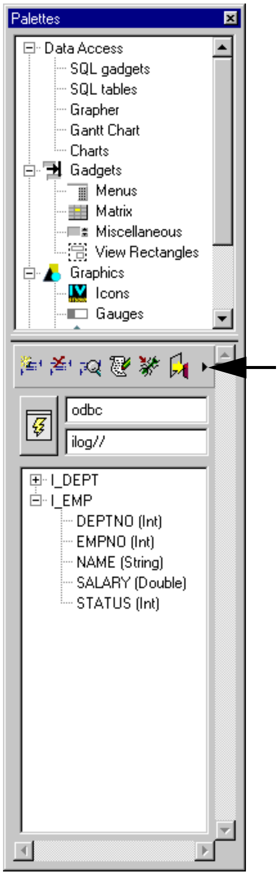
You use four basic steps to create a table in the SQL data source inspector panel:

1. Add the desired database tables to the FROM section.
2. Select the columns to be used from the database tables by dragging a line from a desired table column in the FROM section to a desired position in the SELECT section.
3. Apply selection criteria in the SELECT section to specify the data that will be placed in the table rows.
4. Define in the SELECT section how the data source tables will be displayed.

You will be doing all these procedures in the tutorial chapter that follows.

SQL Schema Editor

The Data Access SQL Schema Editor, which is provided to help you get started with the examples, allows you to create and edit the schema (the structure of tables) in which the data is stored in the database itself. It also enables you to edit the data within a table or delete a table from the database. This editor is described in Appendix C: Appendix C, *SQL Schema Editor*.



SQL Schema Editor toolbar

Figure 1.5 *SQL Schema Editor*

Connections

Three different types of connections allow Data Access to function:

- The communication channel between the application and the DBMS, which is called an *SQL session*.
- The mapping of the data source to the database tables.
- The connection between the data source and the display gadgets.

The connection procedures in this section are presented to help you understand the underlying concepts and as a reference section for future use. You should not attempt these procedures at this point, as you will be doing them in the following tutorial chapter.

SQL Session: Communication Channel Between the Application and the DBMS

When you create an SQL Session, you are creating an Data Access object that represents a communication channel between the application and the DBMS. An application can have several SQL sessions opened at once, thus allowing you to have more than one transaction active at the same time in your application. Also, the same SQL session can be shared by different data sources.

There are two basic ways in which you can establish an SQL session:

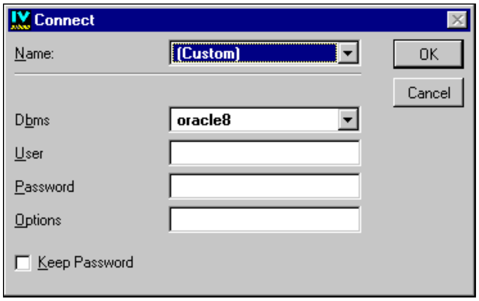
- By creating an SQL session specific to one data source.
- By creating an application-wide SQL session.

Creating an SQL Session Specific to One Data Source

Use one of the following procedures for creating an SQL session specific to one data source.

- **Specifying the SQL session the first time you add a database table to a data source.**

  The connection to the database can be made through a Connect panel that is specific to the type of DBMS being used. The Connect panel automatically appears when you first request access to a table in the database by choosing Add Tables from the Query menu in the SQL Data Source inspector panel.



Connect

Name: (Custom)

Dbms: oracle8

User:

Password:

Options:

☐ Keep Password

  By filling out this panel and clicking OK, you establish a communication channel between the application and the DBMS; that is, you establish an SQL session.

- **Using the SQL Data Source Properties panel.**

  You can also establish an SQL session at any time by using the following procedure:

1. In the Gadgets buffer, double-click the SQL Data Source gadget that you have previously dragged there from the Palettes panel.

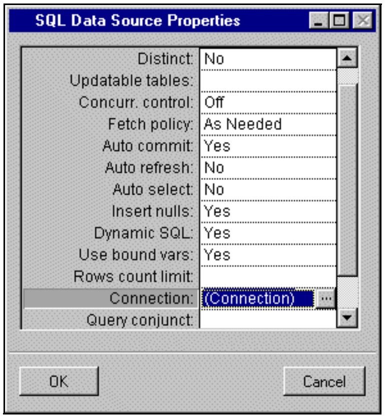
   The SQL Data Source inspector panel appears.

2. Choose Properties... from the File menu of the SQL Data Source inspector.

   The SQL Data Source Properties appears.

3. Click the Connection row.

   A Connection button ... appears in the Connection field:

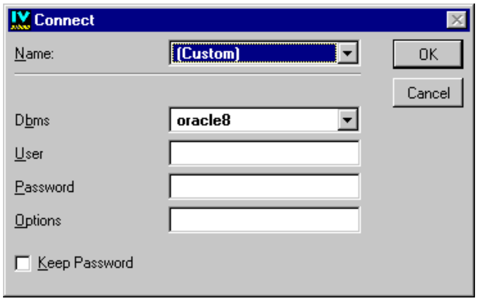


SQL Data Source Properties

Distinct:	No
Updatable tables:	
Concurr. control:	Off
Fetch policy:	As Needed
Auto commit:	Yes
Auto refresh:	No
Auto select:	No
Insert nulls:	Yes
Dynamic SQL:	Yes
Use bound vars:	Yes
Rows count limit:	
Connection:	(Connection) ...
Query conjunct:	

4. Click the Connection button.

   The Connect panel appears:



Connect

Name: (Custom)

Dbms oracle8

User

Password

Options

□ Keep Password

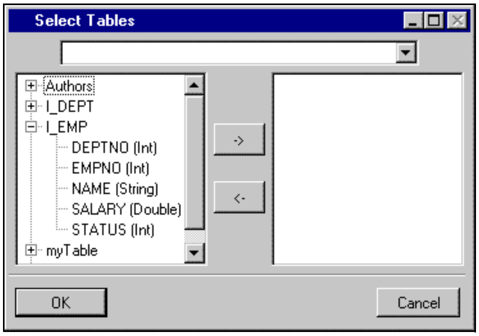
5. Fill out the Connect panel and click OK.

   An SQL session is established.

6. Choose Add Tables from the Query menu in the SQL Data Source inspector panel.

In both cases, if you ignore the Name field of the Connect panel (see “Creating an application-wide SQL session” below) and fill in the other fields, an SQL session is established that is specific to the data source. This means that the SQL commands COMMIT and ROLLBACK, used to update the database, apply only to the data source.

If the SQL session is successfully established using the Connect panel, the Select Tables panel appears. This panel is used to select the table in the database that you want to add to the data source.

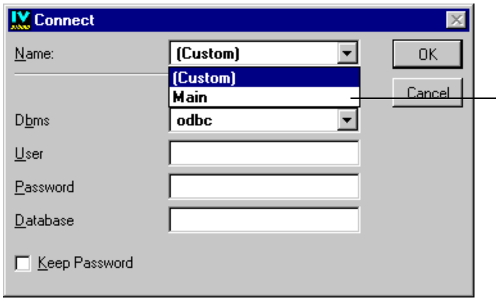


Select one or more tables from the list on the left side of the panel and click the -> button to move the table(s) to the list on the right side of the panel. By clicking OK, you add the table(s) to the data source. The table(s) appear in the FROM section of the SQL Data Source panel.

Creating an Application-Wide SQL Session

You can create an SQL session which, besides the parameters needed for connecting to a database, also has a name. You can then use this name when you want to make use of the SQL session it represents. The connection name belongs to the application, can be used with all loaded buffers, and is saved with the application.

For example, you can use the Name field of the Connect panel to select a connection to a database. You would use this field instead of filling out the other fields of the Connection panel. The current data source can thus immediately make use of this predefined SQL session:



Name: (Custom)
(Custom)
Main
Dbms odbc
User
Password
Database
□ Keep Password

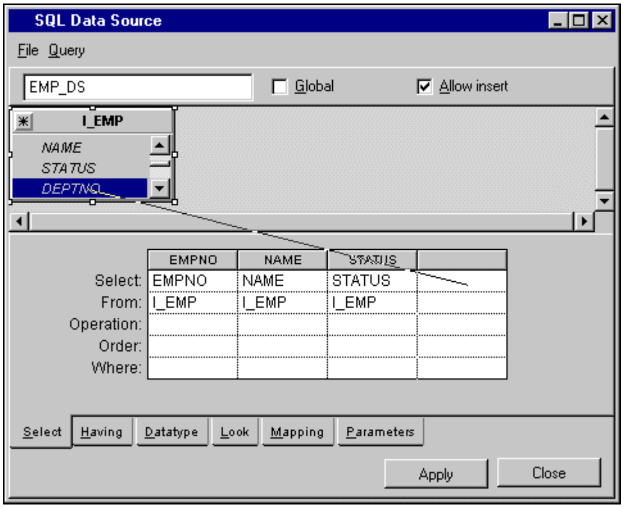
Predefined SQL session names

To create an application-wide SQL session, you choose SQL Application Properties in the Data Access menu. You then use the Sessions notebook page in the SQL Application Properties panel that appears to establish the session. For a detailed description of this procedure, see *Defining a Named SQL Session* on page 51.

With an application-wide SQL session, the COMMIT and ROLLBACK commands apply to all the data sources using the session.

Mapping the Data Source Table to Database Tables

To define a data source from values taken from one or more database tables, you use the SQL Data Source inspector to connect the database table representations in the FROM section to the data source table in the SELECT section. You do this by dragging a line from the name of a table column in the FROM section to the desired column position in the SELECT section.



EMP_DS □ Global ☑ Allow insert

I_EMP
NAME
STATUS
DEPTNO

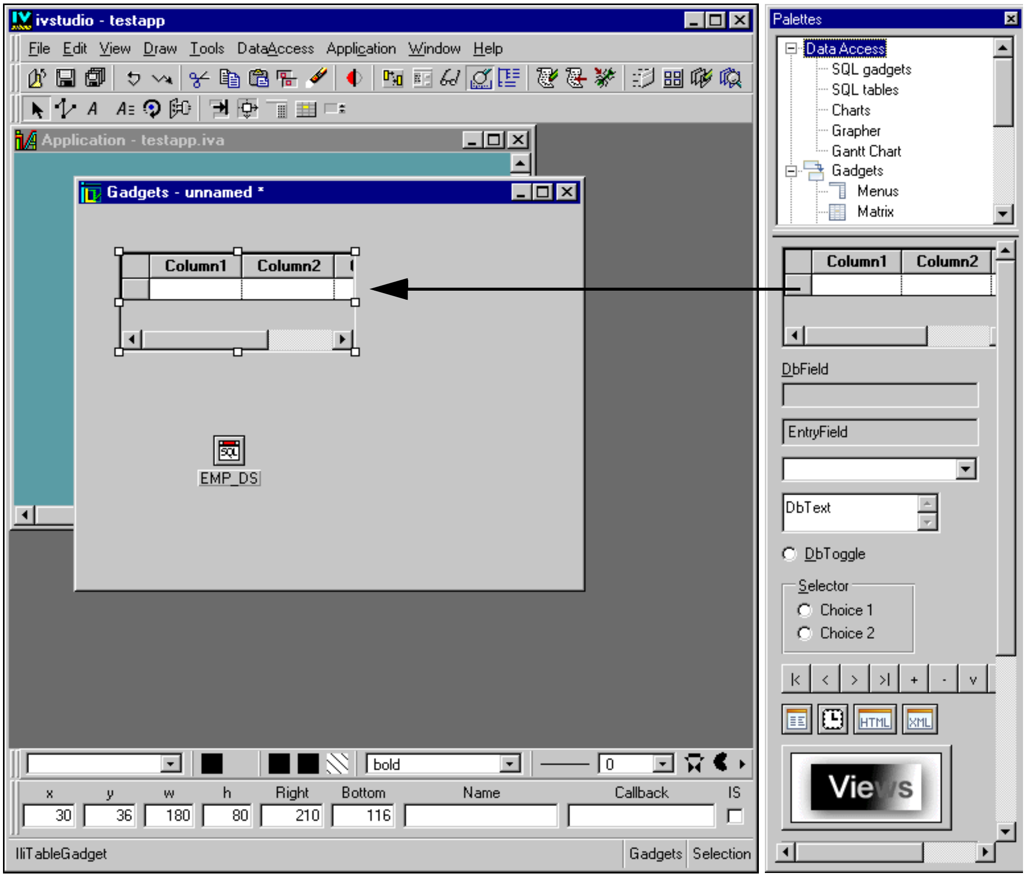
	EMPNO	NAME	STATUS	
Select:	EMPNO	NAME	STATUS	
From:	I_EMP	I_EMP	I_EMP	
Operation:				
Order:				
Where:				

> ***Note:*** *It is possible to set the data source table columns to be the same as the database table columns. You do this by dragging a line from the asterisk in the top left corner of the database table to the data source table in the SELECT section.*

Connecting Display Gadgets to a Data Source

Once you have mapped the data source to the desired tables in the database, you can then display the data by means of Data Access gadgets. To connect a gadget to the data source, see the following procedure:

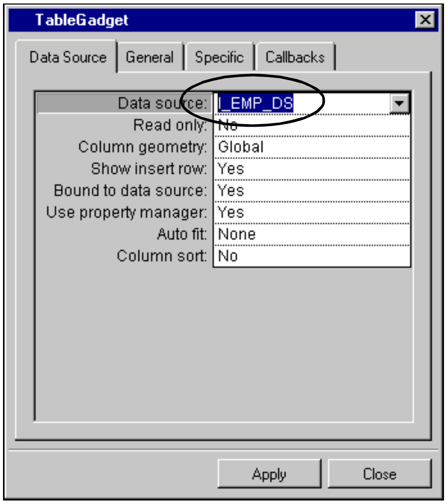
1. Drag the table gadget to the work space of the Main Window:



x	y	w	h	Right	Bottom	Name	Callback	IS
30	36	180	80	210	116			□

2. Double-click the table gadget in the work space to open the Table Gadget inspector panel.

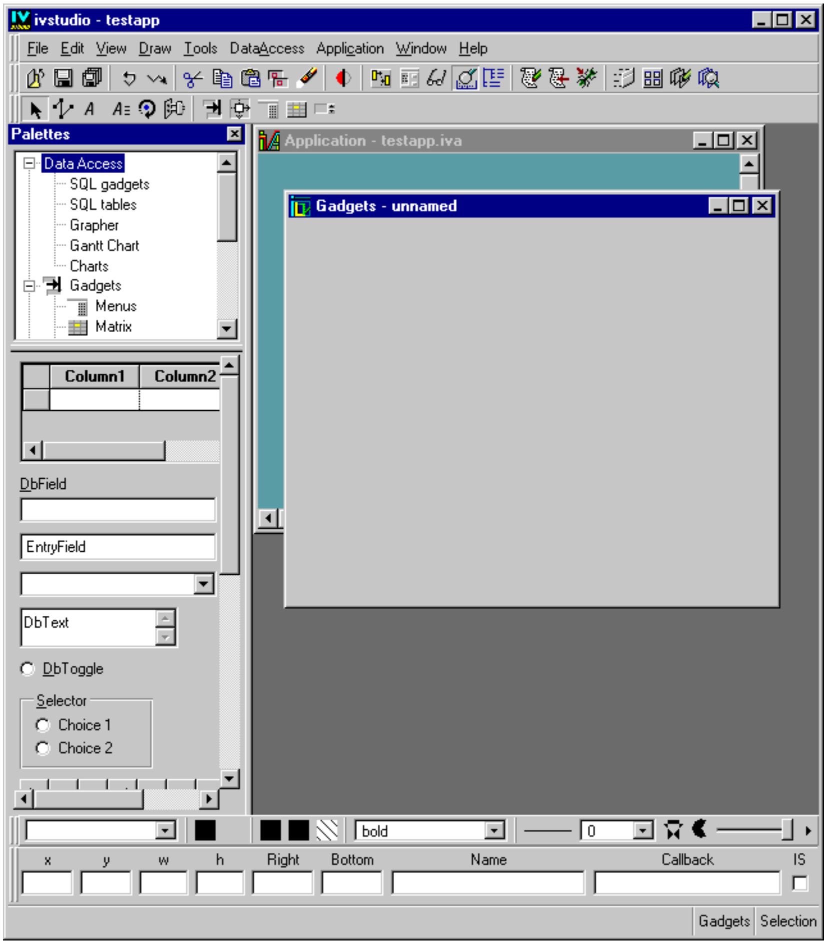
3. In the Table Gadget inspector panel choose the data source to which the gadget is to be connected.



Data source: I_EMP_DS
Read only: No
Column geometry: Global
Show insert row: Yes
Bound to data source: Yes
Use property manager: Yes
Auto fit: None
Column sort: No

Data Access Window

When you launch Data Access, the following window appears:



The Data Access window is divided in two parts:

- Main Window
- Palettes Panel

Main Window

The Main window contains the following areas:

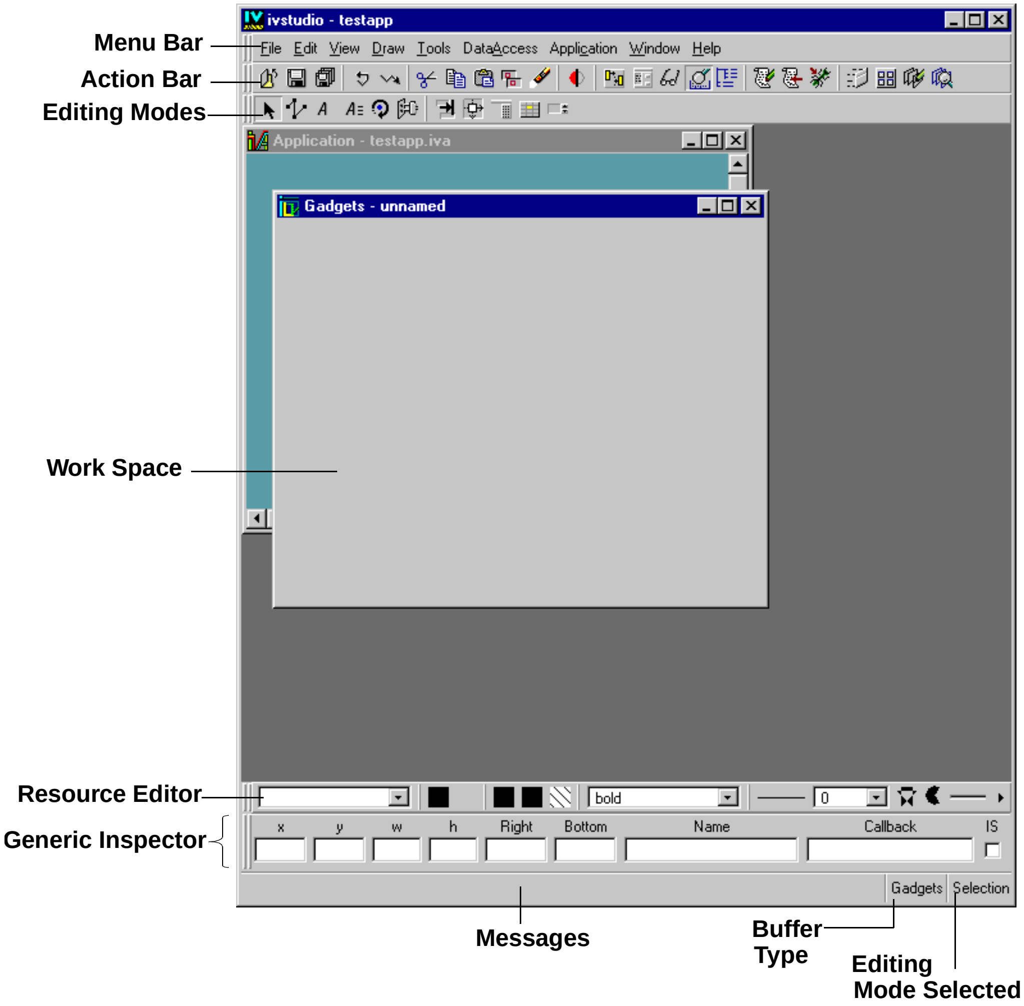


Figure 1.6 Data Access Main Window

- **Menu Bar**—Use these menus to choose various commands (such as Save in the File menu) and to display option panels (through such commands as Resources in the Tools menu). Certain menu commands have their own button just below the menu bar.

- **Action Bar**—These buttons are provided for often-used commands. Some commands are the same as in the menu bar and others are unique.

- **Editing Modes**—By clicking one of the buttons in this area, you choose a particular editing mode, such as selecting, text editing, menu editing, or and so on.

Here are the editing mode buttons you will need in the following tutorial, although there are many others:

Selection Mode—To select, move, and resize objects.

Active Mode—To make the gadgets in the work space active.

Focus Mode—To check the focus chain, that is, the order in which objects in a window are selected (own the keyboard focus) as the Tab key is repeatedly pressed.

Attachments Mode—To attach objects in the panel to guides so that the behavior of the objects can be controlled when the panel is resized.

Menu Mode—To edit pop-up menus.

- **Work Space**—Use this large rectangular area to edit your panels. Two empty buffer windows (Gadgets and Application) are displayed by default when Data Access is launched. You can open new buffer windows at any time by choosing New from the File menu and then the type of buffer window you want (Gadgets, 2D Graphics, and Grapher). Only one Application buffer window can be open at a time. Buffer windows have buttons in the top-right corner that you can use to maximize, minimize, or close them.
- **Resource Editor**—Use this palette to select the following attributes for the objects selected in the work space:
  - Name of predefined palette
  - Foreground
  - Background
  - Pattern color
  - Fill Style
  - Font
  - Line Style
  - Line Width
  - Font and font style
  - Fill Rule Mode
  - Arc Mode
  - Alpha

- Antialiasing Mode

- **Generic Inspector**—Use this area below the work space to display or edit a selected object general properties. The properties of the item you select in the work space are immediately displayed in the fields, but only if you select one object:
  - **x, y***:* upper left corner of the object.
  - **w, h:** width and height of the object.
  - **Right, Bottom**: bottom right corner of the object.
  - **Name:** name of the object.
  - **Callback**: callback name for the object.
  - **IS**: indicates whether the callback is an IBM ILOG Script callback or not.

> ***Note:*** *A callback is the relation between a gadget and a function. That is, when a particular gadget is activated, an associated function is called. When you click a button or a menu item, for example, the system recognizes the click and calls the previously defined function associated with that button or menu item.*

- **Messages**—This area displays two types of information:
  - The C++ class of the object you select in the work space.
  - Procedural information, such as error messages.
- **Buffer Type**—This area shows the type of buffer that is currently being edited. It may be any of the following:
  - Gadgets
  - 2D Graphics
  - Grapher
  - Application
- **Editing Modes Selected**—This area shows which editing mode is active.

> ***Note:*** *The buffer currently being edited can be changed using the Window menu.*

Palettes Panel

The Palettes panel provides predefined gadgets that you can drag-and-drop directly to your work space to create the panels of your final application. This panel is displayed by default when you launch Data Access, together with the Main window. If closed, it can be reopened with the Palettes command from the Tools menu. The Palettes panel is a dockable panel, that you can use outside from the Main Window in order to have more space in the buffer area.

As you can see from the illustration on the next page, the Palettes 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. There are six different types of palettes:

- **Data Access**—This palette contains predefined gadgets specifically designed for SQL database applications.
- **SQL gadgets**—This palette contains SQL gadgets.
- **SQL tables**—This palette allows you to display a tree gadget of the SQL tables in your database. You can drag-and-drop a table from this table into the current buffer of the Main window, just as you drag-and-drop gadgets from other palettes. This palette also contains a toolbar that is the SQL Schema Editor.
- **Charts**—This palette contains a chart graphic that can be connected to data sources.
- **Grapher**—This palette contains Grapher gadgets that can be connected to data sources.
- **Gantt chart**—This palette contains Gantt gadgets that can be connected to data sources.

***Note:** The Grapher and Gantt chart functionality is accessible only if you have purchased and installed these two optional packages.*

For a description of the Data Access and SQL gadgets palette, see *Data Access and SQL Gadgets* on page 32. For a description of the SQL tables palette, see *SQL Tables* on page 33. For a description of the Charts, Grapher and Gantt chart palette, see *Charts, Grapher and Gantt Chart Gadgets* on page 35.

To use a gadget, just drag the one you want from the palette and drop it into the work space of the Main window.

Data Access and SQL Gadgets

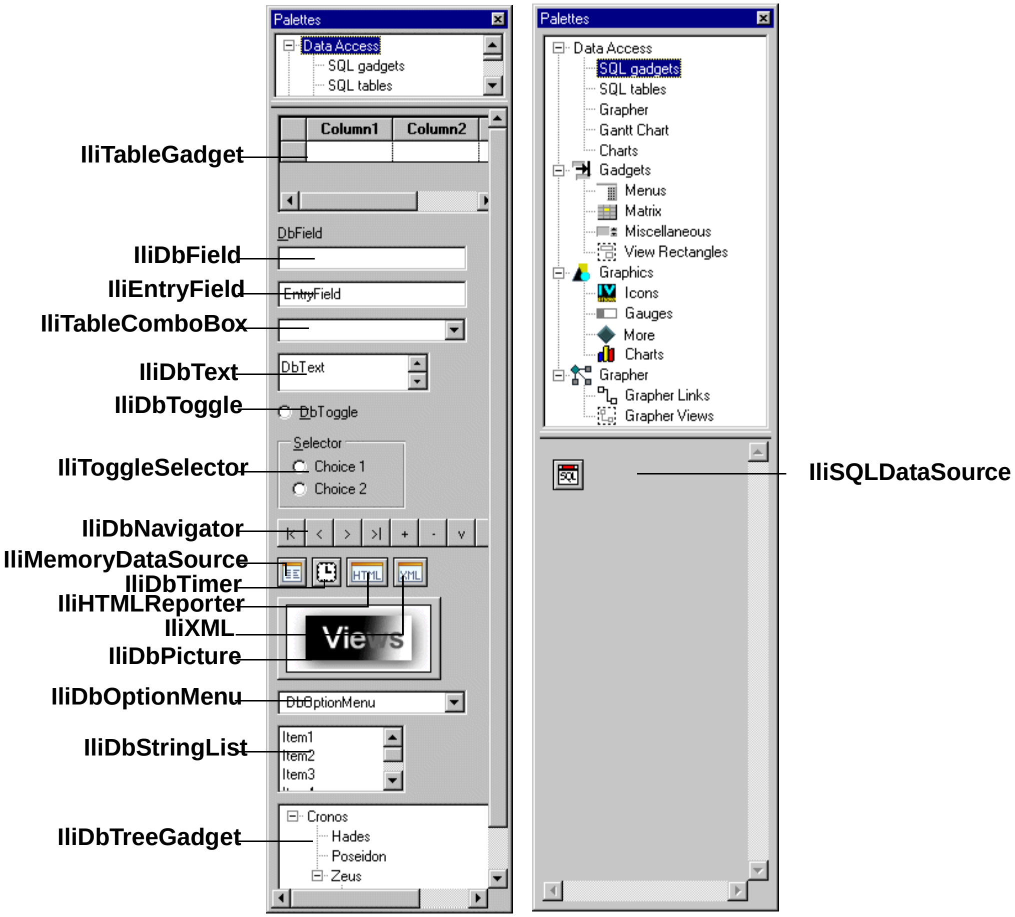


Figure 1.6 Palettes Panel with Data Access and SQL Gadgets

Use the gadgets shown above for the following purposes:

IliTableGadget — For editing tables.

IliDbField — For displaying data with a data-source-aware gadget whose appearance can be dynamically changed (to an entry field, toggle, and so on).

IliEntryField — For displaying text in a data-source-aware text field.

IliTableComboBox — For listing items in a data-source-aware pop-up menu and displaying the item chosen.

IliDbText — For displaying text in a data-source-aware, multi-line scrollable text field.

IliDbToggle — For choosing between three states (True, False, and null) with a data-source-aware toggle button.

IliToggleSelector — For selecting among any number of items using data-source-aware selector buttons.

IliDbNavigator — For creating a toolbar with buttons to navigate through rows and edit data in a data-source table.

IliMemoryDataSource — For defining a local memory data source.

IliDbTimer — For calling a callback periodically.

IliHTMLReporter — For generating an HTML document from a data source.

IliXML — For managing the communication between a data source and an XML stream.

IliDbPicture — For displaying a picture in a data-source-aware gadget.

IliDbOptionMenu — For listing items in a data-source-aware pop-up menu and displaying the item chosen.

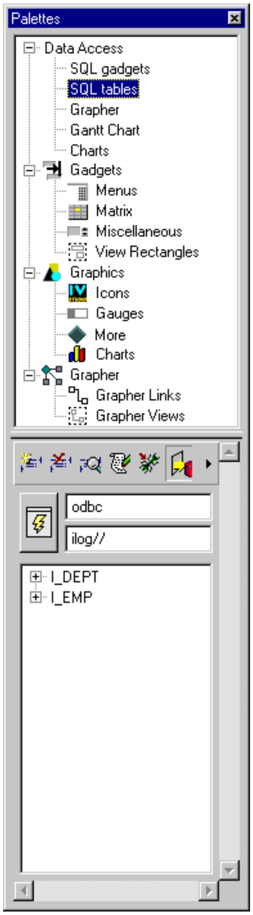
IliDbStringList — For displaying a list of labels in a data-source-aware, multi-line string list.

IliDbTreeGadget — For displaying the contents of data sources in a tree gadget based on a parent/child relationship.

IliSQLDataSource — For providing a link to an SQL database from which a table is defined and displayed.

SQL Tables

When you select SQL Tables under Data Access on the Palettes panel, the lower pane is empty. Click the button at the left that contains the arrow to open the Connect dialog box. After you type your name, password, and the database name, the user-created SQL database tables appear in the lower pane. You can drag the tables from the pane, drop them in the Gadgets buffer window, then double-click a table to open the SQL inspector with the table already in place.



odbc

ilog//

Figure 1.7 Palettes Panel with SQL Tables Gadgets

In the SQL Tables palette you can find the SQL Schema Editor toolbar with the following buttons:

Create table.

Drop selected table.

Edit schema of the selected table.

Edit data of the selected table.

Enable / disable SQL trace.

Export selected table.

Import selected table.

Charts, Grapher and Gantt Chart Gadgets

The gadgets illustrated below appear when you select Charts, Grapher and Gantt chart under Data Access on the Palettes panel.

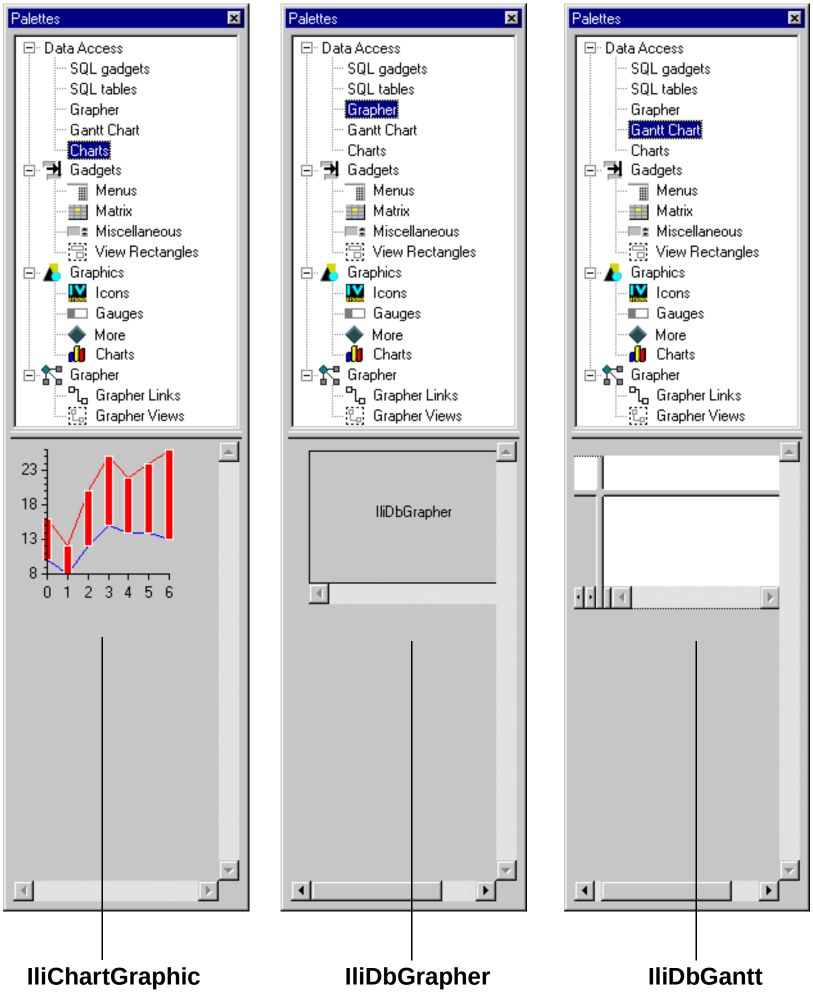


Figure 1.8 Palettes Panel with Charts, Grapher and Gantt Chart Gadgets

Use the above gadgets for the following purpose:

IliChartGraphic — For defining a chart graphic connected to various data sources.

IliDbGrapher — For displaying contents of a nodes and links data source in a grapher.

IliDbGantt — For defining a Gantt chart connected to various data sources.

General Procedures

The procedures in this section provide the basic principles and steps for creating an application using Data Access. They are given for reference only and you should not attempt them now, since you will be doing them in the tutorial chapters that follow. The following is one possible order of procedure for creating an application using Data Access:

1. Create an application.
2. Create panels.
3. Integrate panels into the application.
4. Create a new instance of a panel class.
5. Generate an application.
6. Edit an application.

You are free to choose the order in which these steps are completed. For example, you could first create all your panels, then the application, and then integrate the panels. However, the advantage of creating the application first is that all paths of the panel data files are automatically relative to the application file.

> ***Notes:***
>
> 1. *Applications and panels are created within the Main window. The current buffer determines the type of file that you are currently working on. The current buffer type is shown at the bottom of the Main window. When creating a new buffer, you must specify the type of buffer required. A submenu accessed via the New option in the File menu allows you to do this.*
> 2. *You can move between all the buffers that are currently loaded using the Window menu. The buffer currently being edited is displayed at the top of the list of buffers in the drop-down menu and also in the Main window title bar.*

In the tutorial in this manual, we will be using the numbered procedure shown above. If you like, you can now go directly to the tutorial, which starts in the next chapter *Introduction* on page 45, or you can look over the general procedures that follow.

How to Create an Application

Creating an application consists of creating an application file that has an `.iva` extension. Here is the procedure for creating this file:

1. Choose New > Application from the File menu. The Buffer Type area displays `Application`.

2. Choose Save from the File menu. A file selection dialog box appears in which you can specify the name and directory of the application file. Application files are saved with the `.iva` file extension.

3. To change the application properties, choose Application Inspector from the Application menu. or click the Application Inspector button . The Application Inspector panel appears for you to edit.

    - On the General page, enter the following information:

      Class—Type a valid C++ name for the application, then press Apply.

      Base Class—If you enter a new base class, it must be a subclass of `IlvApplication.`

      File Name—Contains the name of the file where you saved your application.

      Directories—The directories chosen here are also used for the panel class files, unless you override them in the Panel Class inspector.

      By default, the application file directories are set to the current directory; that is, the directory from which IBM ILOG Views Studio was launched. The Data directory field designates the directory where the application `.iva` file is saved. This field is read-only and is updated when the Save As... command is used.

      System—Use the combo box in the System field and the Motif toggle button to choose the system corresponding to your system. Note that the Motif toggle button is only displayed when an X Window system is selected.

    - On the Options page, enter the following information:

      Exit—Select to generate an Exit button in a separate panel that allows you to terminate the application.

      main()—Select to generate a `main` function in the source file.

      Make—Select to generate a `make` file for compiling and testing the generated source code.

      Panel Accessors —Select to generate a member function for each panel instance.

      Include in Header— Select to include the header of the panel classes in the application header file.

      Bitmap Readers—Select the formats of bitmaps that you may wish to load into the application.

4. If necessary, click the Header and Source pages of the Application Inspector to specify code that should be added to the application header and source files.

5. If you are using `jsstudio` or an extension of `jsstudio`, click the Script page to specify whether you want to use IBM ILOG Script and the file you want to load.

6. Click the Apply button to save any changes and the Close button to quit the inspector.

How to Create Panels

The following are the basic steps for building GUI panels with IBM® ILOG Views Studio. The panels are created in the work space of the Main window.

1. If necessary, choose New > Gadgets from the File menu in the Main window to create a new Gadgets buffer. The Buffer Type area should show `Gadgets`.
2. Make sure that the current editing mode is Selection mode. Drag the desired gadget from the Palettes panel to the work space.
3. If applicable, give names to the object and its callback function by entering them in the Name and Callback fields, respectively, in the Generic Inspector area at the bottom of the Main window.
4. If applicable, edit the object properties by double-clicking the object to open its inspector. After making any changes in the inspector, click Apply to apply the changes to the object.
5. If necessary, change colors and font resources of the object using the Resources editor located at the bottom of the Main window.
6. If necessary, align the object with the borders or other objects. Choose Align/Distribute from the Draw menu of the Main window, then select Alignment Panel to display the Alignment panel. The objects align on the first object selected.
7. Directly edit possible object values by first clicking the Active button in the Editing Modes toolbar, then testing the object. To leave Active mode, click another editing mode button.
8. Repeat steps 2-7 for each object needed to complete your panel.
9. If necessary, click the Focus button in the Editing Modes toolbar to define the order of a focus chain. The focus chain is the order in which objects in the window get the keyboard focus as you repeatedly press the Tab key.
10. Specify the size of the panel by choosing Fit to Contents from the Edit menu of the Main window.
11. Define how the objects will behave when the panel is resized. Click the Attachments button in the Editing Modes toolbar. Attach the objects in the panel to the border guides or to other guides you create. You do this by clicking a middle handle of the object with the left mouse button and dragging a line to a guide.
12. Test the object behavior and its attachments. Click the Test button in the toolbar and resize the panel by using the standard resizing mechanism of your windowing system. Click the Test button again to close the Test window.

13. Choose Save As... from the File menu in the IBM ILOG Views Studio window. A file selection dialog box appears for you to save the panel in a data directory of your choice. It is saved as an ASCII file with an `.ilv` file extension.

How to Integrate Panels into an Application

1. Choose Open from the File menu in the Main window and select the panel buffer that you want to integrate into the application. The panel buffer is opened in the Main window work space.

   If the required panel buffer is already loaded, select it from the Window menu to set it as the current buffer.

2. Click the Panel Class Editor button in the toolbar of the Main window. The Panel Class palette is opened at the bottom of the Main window.

3. In the Panel Class palette toolbar, click the New Panel Class button to add the current buffer to the palette. This operation creates a panel class to handle the `.ilv` panel and adds an icon to the application panel classes list.

4. If necessary, while the panel class icon is selected in the Panel Class palette, click the Panel Class Inspector button to change the directory in which the C++ files for the panel class will be generated and to select code to be generated in the source code (Data, Names, Callbacks).

   The Panel Class inspector appears for you to edit.

   - On the General page, you can specify the directories in which any files generated for the panel class are stored. If you do not enter anything in the directory fields, files are automatically generated in those directories specified in the Application Inspector.

   - On the Options page, the Data, Names, and Callbacks toggle buttons enable you to specify whether data, names, and callbacks are generated in the source code for the panel classes. These toggle buttons should be used to set the following:

     Data—Select to generate the data in the source code so that your application does not need to read the data file at run time. The code generated using this data will only be applied on the UNIX platform; with MS-Windows, the generated data string is not used.

     Names—Select to generate in the source code the member functions returning the named objects.

     Callback Declarations—IBM® ILOG Views Studio provides you with a simple way to deal with callbacks. When the Callback Declarations toggle button is checked, it generates an `IlvGraphicCallback` function and declares a default virtual member function. The generated `IlvGraphicCallback` invokes the corresponding virtual member function that has the same name as the callback that you specified in

IBM ILOG Views Studio. Therefore, the names you use for callbacks in IBM ILOG Views Studio must be valid C++ function names.

The Callback Declarations toggle button must be checked for the Callback Definitions toggle button to have any effect.

Callback Definitions—The default definition code (the body function) for these callback virtual member functions can be generated, letting you test your application before defining the real callbacks. By checking the Callback Definitions toggle button, you redefine your own versions of the callbacks in your derived classes.

If you do not want these function definitions to be generated, deselect the Callback Definition toggle button. This is useful if you do not want to derive a class from the generated class. In this case, you can write your own definition of these member functions in a separate file that will not be erased by future code generation.

The callback registering task is generated in the C++ code so that you only have to define the callback methods.

5. If necessary, click the Header and Source pages of the Panel Class inspector to specify code that should be added to the class header and source file.
6. Click the Apply button to save any changes. Click the Close button to quit the inspector.

How to Create a New Instance of a Panel Class

1. Choose Edit Application from the Window menu. You will notice that the Buffer type changes to `Application`.
2. Click and drag the required panel class icon from the Panel Classes palette to the application buffer. The instance appears as a window in the Application buffer. This operation adds a new instance of the panel class to the application.
3. To edit the panel properties (name, title, size, and so on), double-click the title bar of the panel instance. The Panel Instance inspector appears for you to edit.

   The Destroy Callback menu in the General page of the Panel Inspector allows you to define what will happen when you close your panel with the system window manager:

   Exit Quit the application.

   Hide Hide the panel.

   None Nothing happens.

   Click Apply to validate your changes, then click Close.
4. Repeat the steps in the section *How to Integrate Panels into an Application* on page 40 and *How to Create a New Instance of a Panel Class* on page 41 for each of the panels you created for your application.

How to Generate an Application

1. Ensure that the current buffer is the Application buffer by selecting <Application> in the Window menu, if required. Choose Save from the File menu to save your application in an `.iva` file. If you have not already saved the application file, a file selection dialog box appears in which you can enter the directory and name of the application file.

2. To generate C++ code for the application and for the panel classes, choose Generate All from the Application menu. If you later make changes or add panels, choose Generate to generate only what has changed.

3. To compile the application with all its panel(s), run the make utility on the generated `make` file outside of IBM® ILOG Views Studio.

4. To test the application, run the generated program.

How to Edit an Application

This section provides a general description of how to edit an application. For more detailed information on how to carry out the processes involved, see the *IBM ILOG Views Controls User's Manual*.

An Application buffer can be edited within the Main window work space. Use the Panel Class palette, which is displayed at the bottom of the Main window, to edit an application.

The Panel Class Palette

Using the Panel Class palette, you can do the following:

- Create a new panel class for the current buffer.
- Remove an existing panel class (the selected panel class).
- Inspect a panel class (the selected panel class).
- Create a new panel instance in an application by dragging the selected panel class into the application buffer.

Once an instance of a panel class instance has been created in the Application buffer, you can inspect it, test it, minimize it, and manipulate it in the same way as a normal window.

Testing an Application

When the current buffer is your application buffer, all the panels of the application can be tested with the Test button. A window is opened for each of the visible panel instances in the application.

Code Generation

The following options in the Application menu allow you to generate your code:

- **Generate**—Only the application class files and panel classes that have changed and have not yet been saved.
- **Generate All**—All files for the application: application file (`.iva`), panel files (`.ilv`), header files (`.h`), source files (`.cc`), and make file.
- **Generate Application**—Only the application file (`.iva`).
- **Generate Panel Class**—Only the current panel class file (`.ilv`).
- **Generate Makefile**—Only the make file.
- **Generate Panel Subclass**—Only the panel subclass file (`.ilv`).

CHAPTER

2

Introduction

This chapter introduces you to the Data Access tutorial in which you are going to create an employee database application.

You will be performing the following tasks:

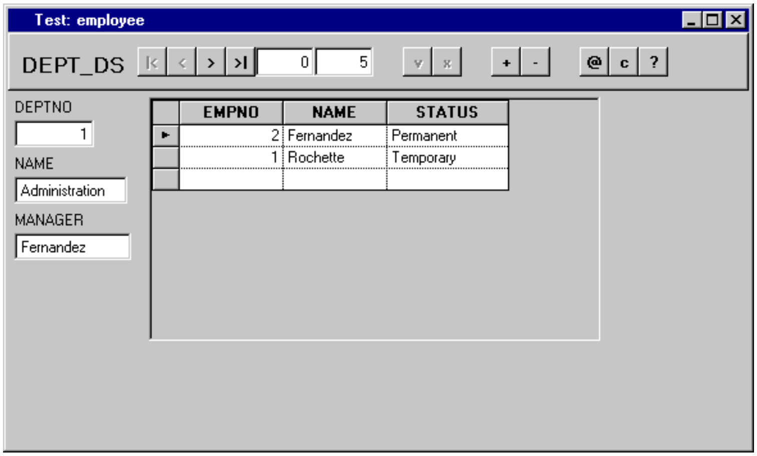
- *Setting Up the Database*
- *Launching Data Access*
- *Creating the employdb Application*
- *Saving and Opening Windows*

What You Are Going To Do

This section shows the finished windows used in an application named `employdb` and describes the procedure for creating the application. The `employdb` database application consists of four windows:

- **Employee**

  This window allows you to automatically display a list of employees in a table gadget for the department whose number is displayed in the text field of a form.



DEPTNO: 1

NAME: Administration

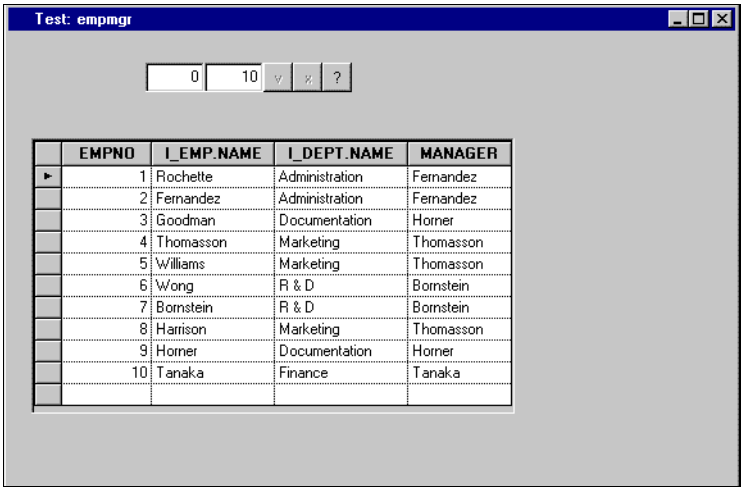
MANAGER: Fernandez

EMPNO	NAME	STATUS
2	Fernandez	Permanent
1	Rochette	Temporary

Figure 2.1 Employee Window

- **Employee Manager**

This window uses a table gadget to display employee information taken from two tables, the I_EMP and I_DEPT database tables. It provides a navigator gadget so you can use query mode to enter your own selection criteria for the display in the table gadget.

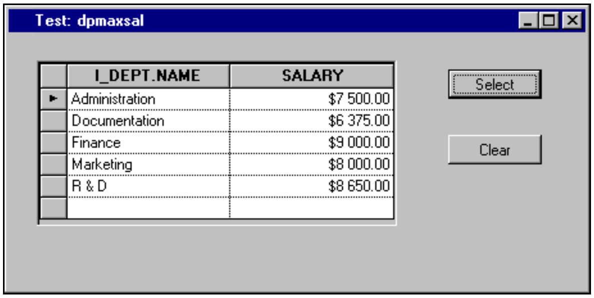


EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
1	Rochette	Administration	Fernandez
2	Fernandez	Administration	Fernandez
3	Goodman	Documentation	Horner
4	Thomasson	Marketing	Thomasson
5	Williams	Marketing	Thomasson
6	Wong	R & D	Bornstein
7	Bornstein	R & D	Bornstein
8	Harrison	Marketing	Thomasson
9	Horner	Documentation	Horner
10	Tanaka	Finance	Tanaka

Figure 2.2 Employee Manager Window

- **Department Maximum Salary**

This window uses a table gadget to display the maximum salary for each department.



I_DEPT.NAME	SALARY
Administration	$7 500.00
Documentation	$6 375.00
Finance	$9 000.00
Marketing	$8 000.00
R & D	$8 650.00

Figure 2.3 *Department Maximum Salary Window*

- **Control**

  This window consists of the menus used to display the other windows and to quit the application.

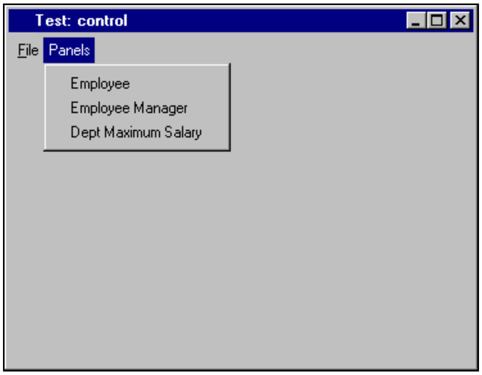


Figure 2.4 *Control Window*

Procedures for Creating the employdb Application

Here are the major steps you will be following to create the application:

1. Create or import the database schema. The database schema is the structure in the form of tables in which the data is to be stored in the database.

2. Create the application file.

3. Create an application Employee window, that is, the window in which the user displays and edits the employee data stored in the tables.

4. Perform various operations to select data from the database and to display it in different ways.

5. Save the window and add it to the application.
6. Repeat steps 3-5 for the Employee Manager and Department Maximum Salary.
7. Create, save, and add a Control window that displays the other windows and quits the application.
8. Test the completed application.

Setting Up the Database

This tutorial uses a database schema with two tables, I_EMP (employee) and I_DEPT (department), that you can set up by either of the following methods:

- Use your own database system tools to create the two tables, using the following SQL statements as a model. The statements are based on the SQL2 standard.

```
I_EMP Table
CREATE TABLE I_EMP (
EMPNO     INTEGER      NOT NULL,
NAME      VARCHAR(25)  NOT NULL,
STATUS    INTEGER      NOT NULL,
DEPTNO    INTEGER      NOT NULL,
SALARY    REAL         NULL,
PRIMARY KEY (EMPNO))

I_DEPT Table
CREATE TABLE I_DEPT (
DEPTNO    INTEGER      NOT NULL,
NAME      VARCHAR(25)  NOT NULL,
MANAGER   VARCHAR(25)  NOT NULL,
PRIMARY KEY (DEPTNO))
```

- Use the Data Access SQL Schema Editor delivered with the product to define the tables. See Appendix C, *SQL Schema Editor* for the procedure to create the tutorial database schema.

Launching Data Access

To launch Data Access on a Microsoft windows platform, select *Programs>ILOG>VIEWS*. With Data Access, it is no longer necessary to use `dbstudio` to design your application. You can use `studio` (the `ivfstudio` program) and load the Data Access plug-ins. Data Access provides the following plug-ins:

- `dbmdstud`: Data Access gadgets and tools - IBM® ILOG Views Data Access license needed
- `dbmdsql`: Data Access SQL gadgets - IBM ILOG Views DB Link license needed

- `dbmdgrap`: Data Access grapher gadgets - specific license
- `dbmdgant`: Data Access Gantt chart gadgets -specific license
- `dbmdchart`: Data Access chart gadgets - specific license
- `dbmdchart30`: old Data Access chart gadgets
- `dbmdinform30`: old Data Access gadgets

However, if you want to use `dbstudio`, you must have a license for the Manager package and install it. After this installation, you can go to the `dbstudio` directory (`bin/dbstudio`) to compile it.

The Main window and the Palettes panel appear.

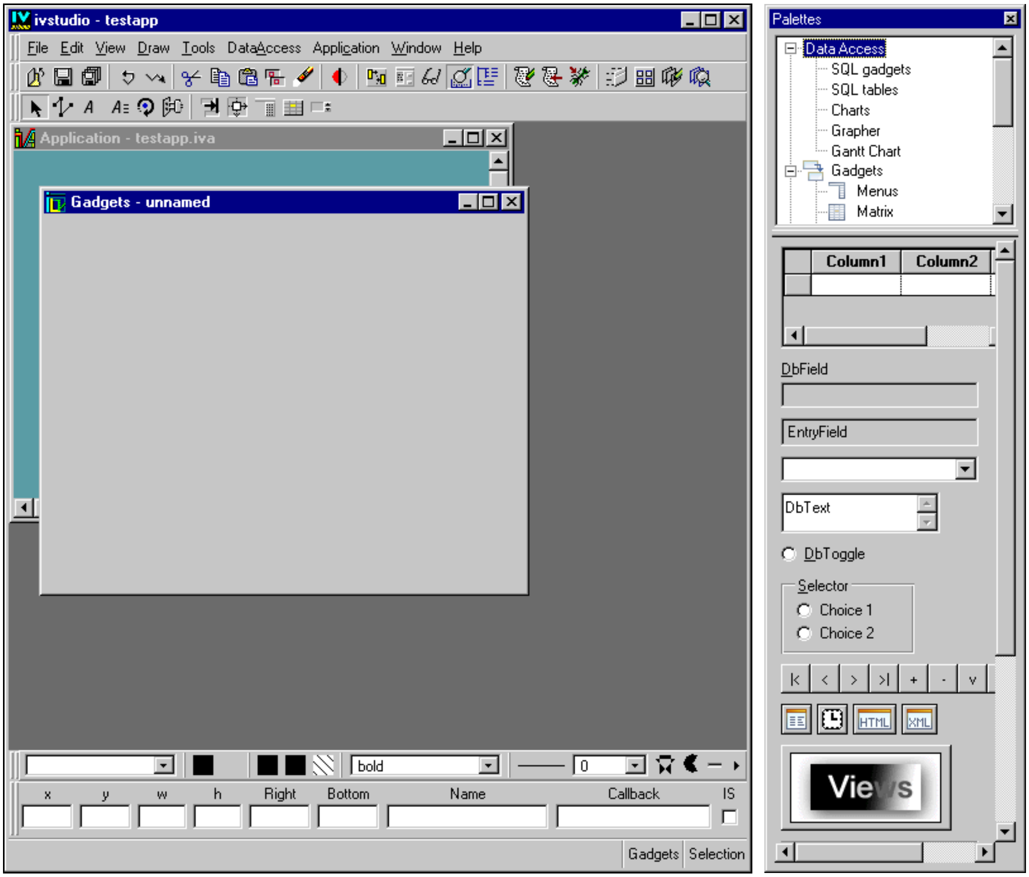


Figure 2.5 Data Access Start-up Windows

See the section *Main Window* on page 27 and *Palettes Panel* on page 30 for descriptions of these windows.

Creating the employdb Application

In this section, you are going to set up the application by creating these elements:

- The application file.
- An application-wide, SQL session that can be used by all the *data sources* you are going to define. See *Data Source* on page 16 for an explanation of this term.

Creating the Application File

To create an application with the name `employdb`, do the following:

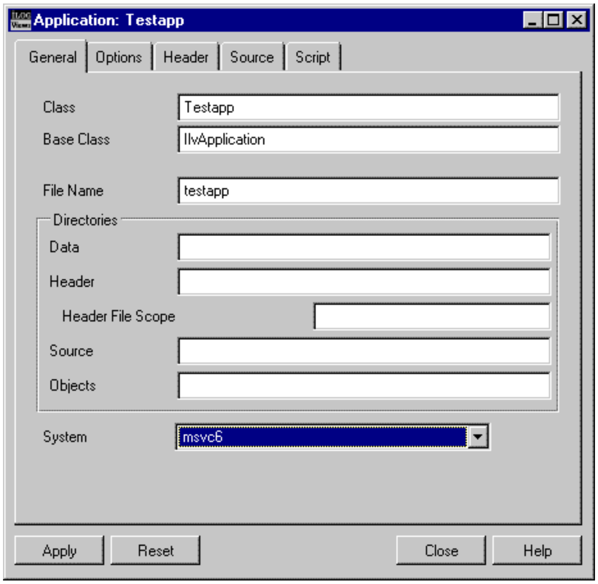
1. Outside of Data Access, create a subdirectory named `data`. You will use this subdirectory to save the application file and the files for the windows that you will be building in this tutorial.

2. Choose <Application> from the Window menu.

   This changes the current buffer to the Application buffer. By default, this Application buffer is called `testapp`, provided you have not already opened or created an application file.

3. Choose Application inspector from the Application menu.

   The Application inspector appears.



Class: Testapp
Base Class: IlvApplication
File Name: testapp
Data:
Header:
Header File Scope:
Source:
Objects:
System: msvc6

4. On the General notebook page of the Application inspector, type `employdb` in the Class field. Leave `IlvApplication` in the Base Class field. Click Apply. Then click Close to close the Application inspector.

5. In the Application buffer window, choose Save As... from the File menu.

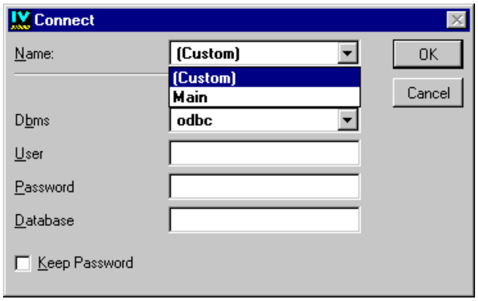
   A file selection dialog box appears.

6. In the dialog box, give the application the name `employdb.iva` and select the `data` subdirectory you created in step 1. Then click Save.

   The application is saved as the `employdb.iva` file in the `data` subdirectory.

Defining a Named SQL Session

The application in this tutorial will use several data sources, all using the same SQL session. You are now going to define this application-wide SQL session, giving it a name. The following figure shows how the name of the selected SQL session (“Main”) appears in the Name combo box menu of the Connect panel when you have finished defining it.


Connect

Name: (Custom)

(Custom)
Main

Dbms odbc

User

Password

Database

☐ Keep Password

Using the Connect panel, you will be able to select a name in the combo box to establish an SQL session between the application and the database. For more information on SQL sessions, see *SQL Session: Communication Channel Between the Application and the DBMS* on page 21.

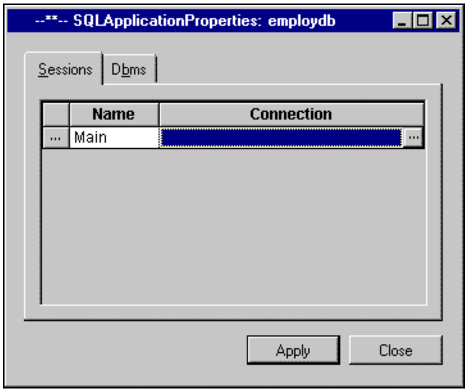
To define a named SQL session, do the following:

1. Choose SQL Application Properties from the DataAccess menu.

   The SQL Application Properties panel appears.

2. If the Sessions notebook page is not active, click it and type `Main` in the Name field.

3. Click the Connection field, then click the Connection button ... that appears on the right.

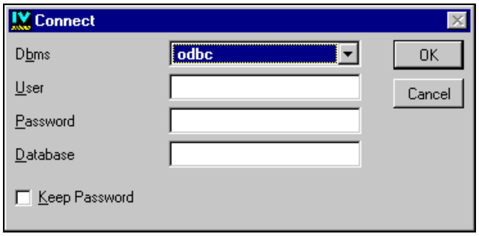

--**-- SQLApplicationProperties: employdb

Sessions | Dbms

	Name	Connection
...	Main	

Note: *The "--**--" characters appear in the title bar until you click Apply.*

The Connect panel appears.



Connect

Dbms odbc

User

Password

Database

□ Keep Password

4. Enter the connection parameters. These parameters depend on the DBMS being used. You can activate the Keep Password button to save the password with the application. Otherwise, you are prompted for the password in the next application session. Click OK.

   The Connect panel disappears and the parameters, prefaced by the database system name, appear in the Connection field of the SQL Application Properties panel.

5. Click Apply in the SQL Application Properties panel.

   The named session is validated and a new row appears for defining another named SQL session, if desired.

6. Click Close to close the panel.

Saving and Opening Windows

You are now going to start creating the windows for the `employdb` application. It is a good idea to save your windows periodically as you build them or you may simply want to save the current window while taking a break during this tutorial. To save and reopen your windows, follow the procedures below as you work through the tutorial.

Saving a Window for the First Time

1. Make sure you have already created a `data` subdirectory *outside* of Data Access in which all the windows will be saved (See *Creating the Application File* on page 50).

2. In the Main window, choose Save As... from the File menu.

   A file selection dialog box appears.

3. Select the `data` subdirectory in which you want to save the window.

4. Type the name of the window as `filename.ilv`. You should use the following names for the windows you will be building when you save them for the first time later in this tutorial:

   - `employee.ilv`

- `empmgr.ilv`
- `dpmaxsal.ilv`
- `control.ilv`

5. Click Save in the dialog box.

   The window is saved with the file name you just typed.

Saving an Edited Window

To save an edited window (that is, one that has already been saved and that you have opened and edited), click the Save button in the tool bar in the Main window or choose Save from the File menu.

The updated window is immediately saved in the same place with the same name.

Opening a Saved Window

1. To open a window that was previously saved, click the Open button in the tool bar in the Main window or choose Open from the File menu.

   A file selection dialog box appears. You can use the File type option menu to specify the type of file that you want to load.

2. Select the name of the window you saved, then click Open.

   The saved window appears in the work space of the Main window, replacing any existing window as the current buffer.

> ***Note:*** *The window that is replaced in the work space can be redisplayed by selecting it from the Window menu.*

CHAPTER

3

Database Features

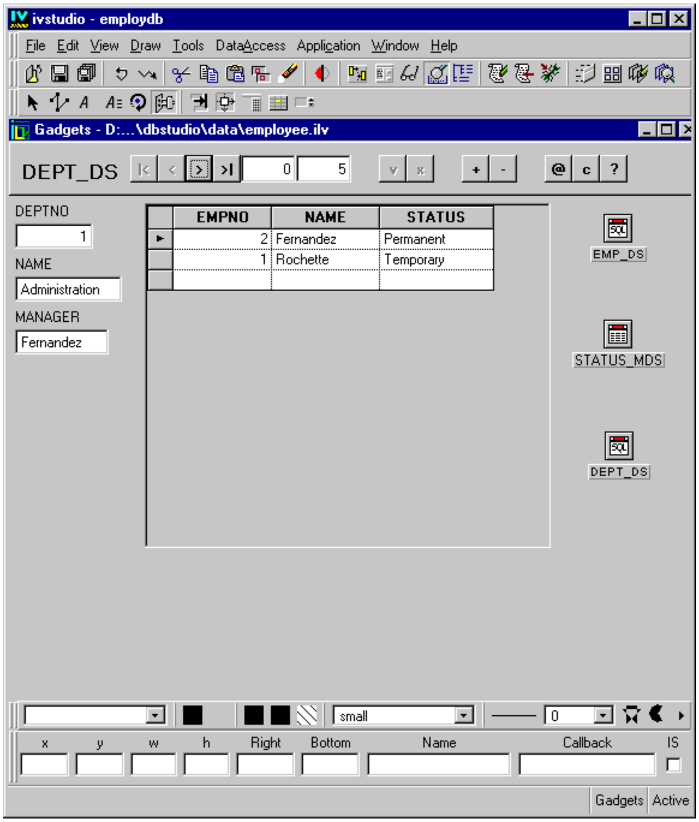
In this chapter, you begin to construct the Employee window of the `employdb` application by defining data sources and database-specific gadgets. You will also use a memory data source, a form, and various operations to establish selection and display criteria.

In this chapter, you will perform these operations:

- *Defining the EMP_DS SQL Data Source*
- *Connecting a Table Gadget to the EMP_DS Data Source*
- *Using Selection and Display Criteria*
- *Defining the STATUS_MDS Memory Data Source*
- *Connecting a Table Gadget to the STATUS_MDS Data Source*
- *Mapping Cells Between Data Sources*
- *Defining the DEPT_DS SQL Data Source*
- *Connecting a Table Gadget to the DEPT_DS Data Source*
- *Mapping and Formatting the Cells*
- *Defining a Form*, which consists of data-source-aware gadgets, and connect it to the DEPT_DS Data Source
- *Using Parameters to Link Data Sources*.

What You Are Going To Do

You are now going to create the Employee window for the `employdb` application. When finished, it will look like this:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv
DEPT_DS
0
5
+ -
@ c ?
EMP_DS
STATUS_MDS
DEPT_DS
small
0
x y w h Right Bottom Name Callback IS
Gadgets Active

DEPTNO
1

NAME
Administration

MANAGER
Fernandez

	EMPNO	NAME	STATUS
▸	2	Fernandez	Permanent
	1	Rochette	Temporary

Figure 3.1 *Employee Window*

On the left side of the window the DEPTNO, NAME and MANAGER fields represent the three Form Elements. The table gadget is located in the middle of the work space, while on the right side the three icons represent the EMP_DS SQL Data Source Gadget, the STATUS_MDS Memory Data Source Gadget and the DEPT_DS SQL Data Source Gadget, respectively.

Defining the EMP_DS SQL Data Source

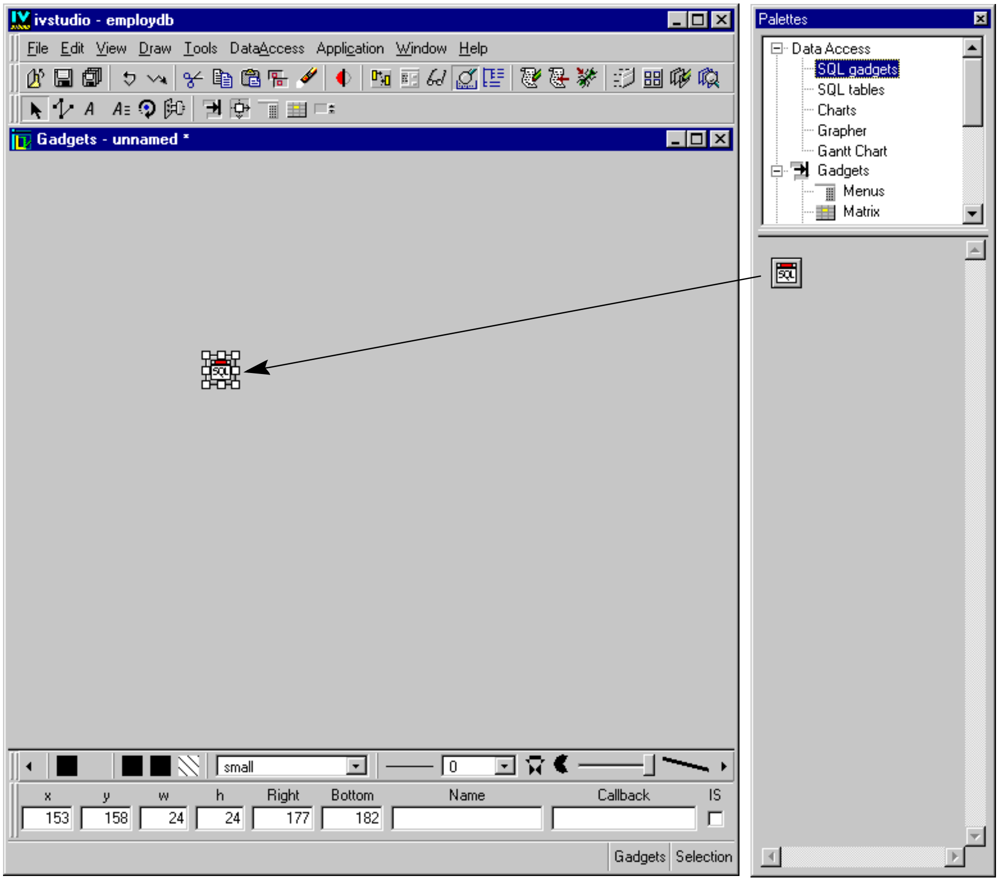
To define an SQL data source using the I_EMP table in the Employee database, do the following:

1. Make sure that you have an empty Gadgets buffer open. Select the unnamed buffer from the Window menu.

   If for some reason the unnamed buffer does not exist, you will need to create a new Gadgets buffer. Choose New > Gadgets from the File menu. When you choose New > Gadgets, the new buffer becomes the current buffer. Any buffers that were already opened can be accessed from the Window menu.

   To give yourself more room in the work space, you can minimize the Application buffer window and resize the Gadgets buffer window by dragging a corner or side.

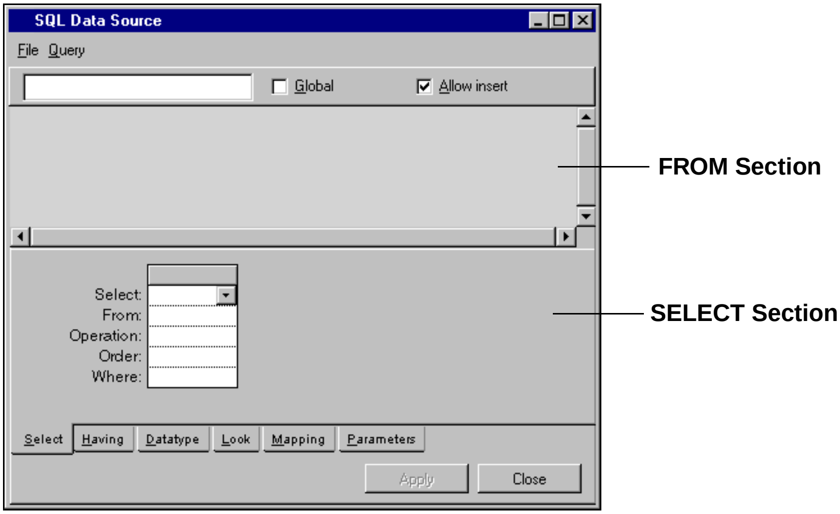
2. Drag an SQL Data Source gadget from the SQL gadgets palette to the Gadgets buffer of the Main window.



x	y	w	h	Right	Bottom	Name	Callback	IS
153	158	24	24	177	182			□

3. Double-click the SQL data source gadget.

The SQL Data Source inspector appears.



□ Global

☑ Allow insert

FROM Section

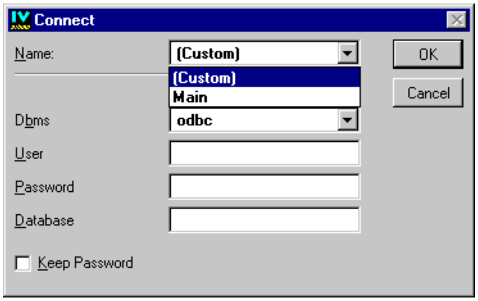
SELECT Section

This panel has two main sections:

- **FROM section**—the area where you place representations of the database tables from which values are to be taken.
- **SELECT section**—the area where you select the criteria for defining the data source tables and their display.

4. Type the data source name `EMP_DS` in the Name field at the top of the panel.
5. Choose Add Tables... from the Query menu to add a table to the SQL data source.

   Since this is the first time an operation needs a connection to the database, the Connect panel appears.
6. In the Name field combo box menu, select the SQL session Main. You created this session earlier in section *Defining a Named SQL Session* on page 51.



Name: (Custom)

Dbms: odbc

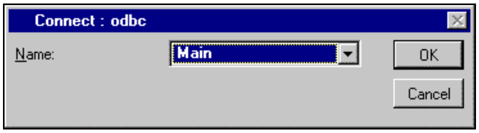
User

Password

Database

□ Keep Password

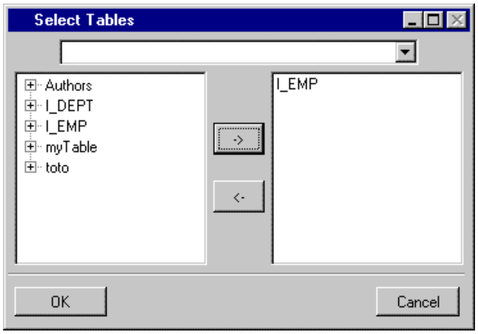
The Connect panel changes to show only the name of your selected session.



Note: *If you filled in the connection parameters instead of selecting a name, you would have created an SQL session specific to the current data source. The application-wide SQL session you created can be used by all the tutorial data sources.*

7. Click OK.

The Select Tables panel appears with the names of the I_EMP and I_DEPT tables. These are the tables that you added to the Employee database at the beginning of the tutorial.



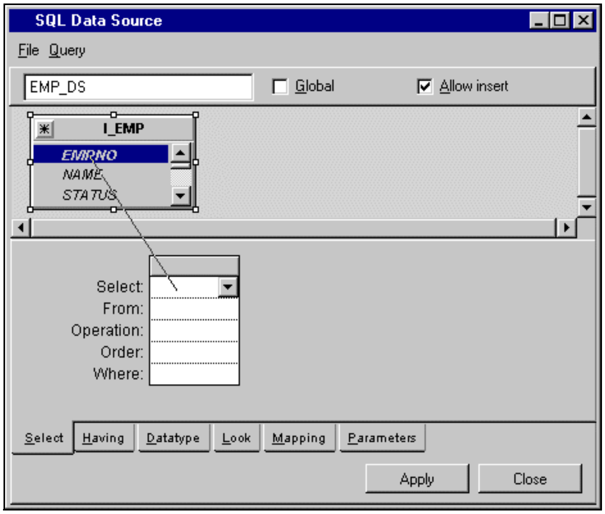
8. Select the name I_EMP in the list on the left of the panel and click the `->` button to add it to the right list. Then click OK to add a representation of the I_EMP database table to the EMP_DS SQL data source.

Notes:

1. *To remove a name from the right list, select it and click the* `<-` *button.*

2. *To display the fields of a table in the list, double-click its name or on the* `'+'` *to the left of its name.*

The table appears in the FROM section of the SQL Data Source inspector.



9. You can position the table in the FROM section by doing one or more of the following:
   - Use the scroll bars that appear with the table if the table does not fit in the FROM section.
   - Increase the size of the FROM section by using the system Window manager to enlarge the entire SQL Data Source inspector panel.
   - Change the size of the table by dragging its handles.
   - Move the table within the FROM section by clicking the tables title bar and dragging.

10. Leave the Allow Insert check box activated to allow users to insert new employees later.

    Leave the Global check box deactivated. When the check box is activated, the current data source can be used by more than one panel.

11. Click EMPNO in the table (which becomes highlighted) and drag a line to the empty column in the SELECT section as shown in the previous figure.

    EMPNO appears in the header and in the Select cell. I_EMP appears in the From cell, showing from which table the data is taken.

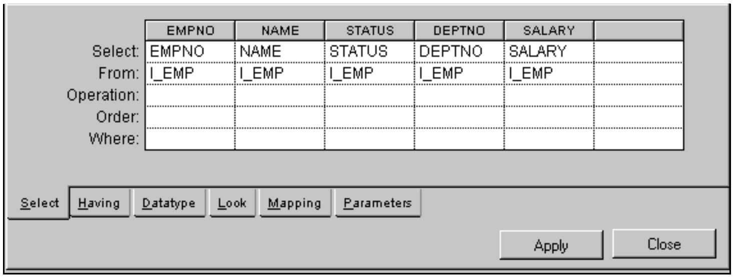
12. Repeat step 11 for the NAME, STATUS, DEPTNO and SALARY columns of the I_EMP table. If you make a mistake, you can edit the column display by choosing column commands from the Query menu:
    - **Append Column** adds a column to the right of all existing columns.
    - **Insert Column** adds a column to the left of the selected column.

- **Delete Column** removes the selected column. Alternatively you can remove the selected column using the Delete key.

You can also drag a column by its header and drop it where you want.

> ***Note:*** *In this tutorial, you want the data source table to have exactly the same columns as the database table. Therefore, instead of dragging each column manually from the FROM section to the SELECT section, you could have dragged the whole table by using the asterisk in the top left corner of the database table.*

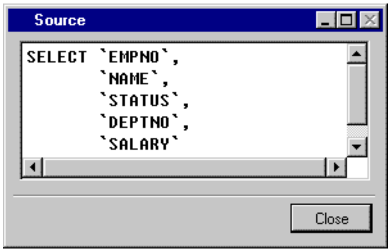
The SELECT section now looks like this after adjusting the size of the window and columns. See *Additional Information about Editing Tables* on page 67 for information on adjusting columns.



	EMPNO	NAME	STATUS	DEPTNO	SALARY	
Select:	EMPNO	NAME	STATUS	DEPTNO	SALARY	
From:	I_EMP	I_EMP	I_EMP	I_EMP	I_EMP	
Operation:						
Order:						
Where:						

13. Choose View Source... from the File menu in the SQL Data Source inspector.

The Source panel appears. This is a read-only panel allowing you to see the SQL statements created by your entries.

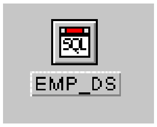


```
SELECT `EMPNO`,
       `NAME`,
       `STATUS`,
       `DEPTNO`,
       `SALARY`
```

Click Close in the Source panel when you are finished with it.

14. Click Apply in the SQL Data Source inspector.

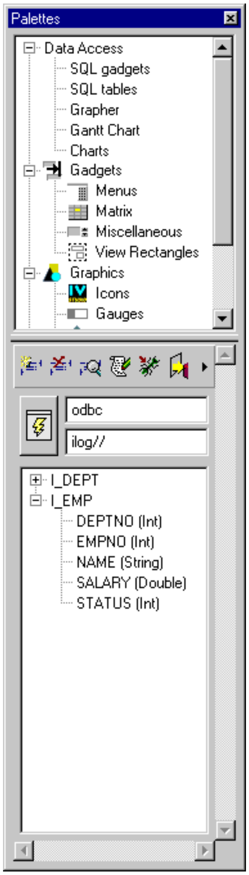
EMP_DS appears under the SQL Data Source gadget in the Main window.



You have now defined the EMP_DS data source.

Additional Information About Using the SQL Tables Palette

The SQL Table palette is opened from the upper pane of the Palettes panel when you click SQL tables under the Data Access tree item.



odbc

ilog//

Click the Connect button to establish a database connection or to change the database connection parameters (user, server or database name). When you are connected to a database, the SQL Table palette displays the list of tables in the database that belong to the user. Each table can be expanded to show the name and types of its columns. To expand a table, click the + sign to the left of the table name.

In addition, any table can be dragged with the mouse and can be dropped into the current buffer in the Main window. This creates a new SQL data source automatically and places it in the buffer. The data source is based on the database table that has just been dragged and contains all the columns of the table.

In this way, you can create SQL data sources instantly and then customize them through the SQL Data Source inspector panel.

The SQL Tables palette has also the SQL Schema Editor toolbar with the following buttons:

Create table.

Drop selected table.

Edit schema of the selected table.

Edit data of the selected table.

Enable / disable SQL trace.

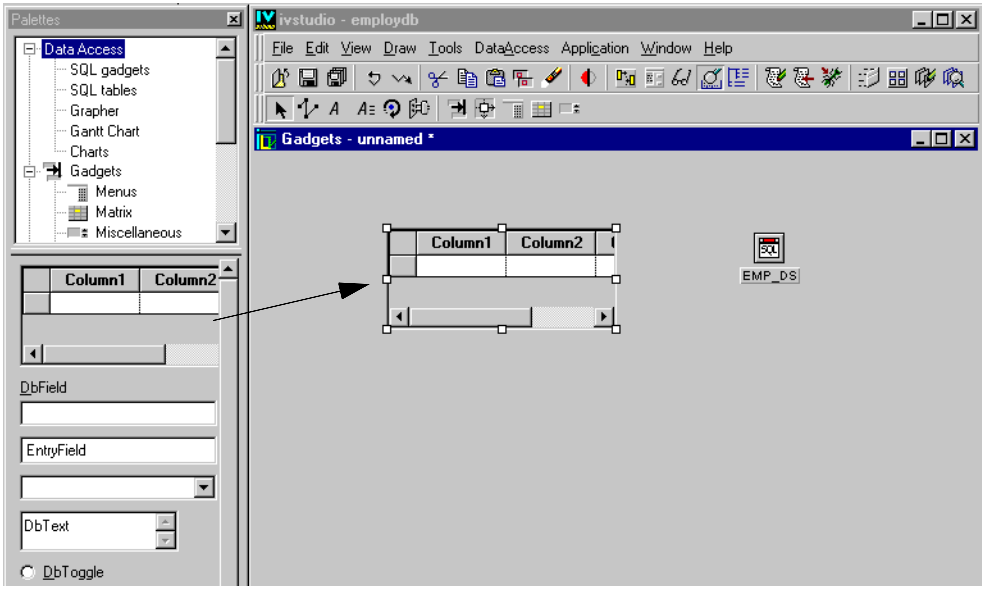
Export selected table.

Import selected table.

Connecting a Table Gadget to the EMP_DS Data Source

To connect a table gadget to the EMP_DS data source, do the following:

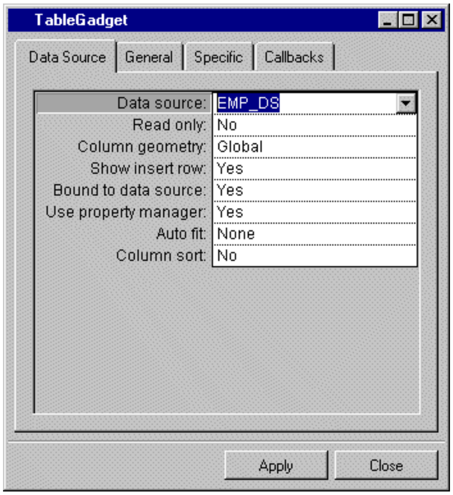
1. Drag a table gadget from the Data Access palette to the Main window.



2. Double-click the table gadget.

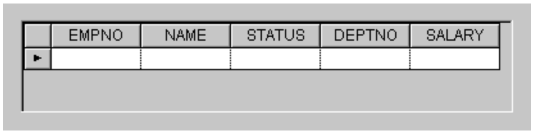
   The Table Gadget inspector appears.

3. Use the combo box to enter `EMP_DS` in the Data source field of the Table Gadget inspector. Click Apply.



Data source:	EMP_DS
Read only:	No
Column geometry:	Global
Show insert row:	Yes
Bound to data source:	Yes
Use property manager:	Yes
Auto fit:	None
Column sort:	No

The column headers in the table gadget in the Main window change to match those of the EMP_DS data source table. Adjust the size of the table gadget so you can see all of the columns in the table. Your table gadget should now look like this:

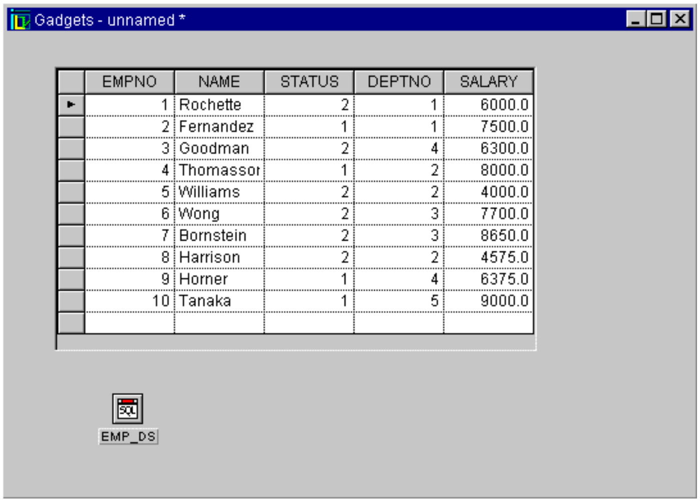


EMPNO	NAME	STATUS	DEPTNO	SALARY

4. In the Main window, click the Active button in the Editing Mode toolbar at the top.

   You can now edit the table gadget.

5. Fill in the table gadget cells as shown in the following figure. The data is automatically entered in the database. See *Additional Information about Editing Tables* on page 67.



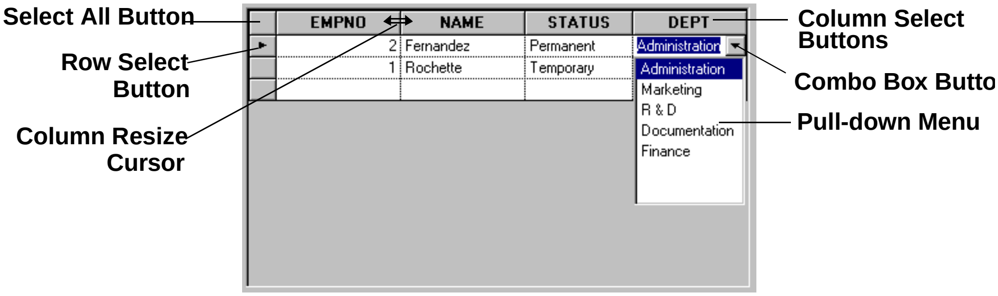
EMPNO	NAME	STATUS	DEPTNO	SALARY
1	Rochette	2	1	6000.0
2	Fernandez	1	1	7500.0
3	Goodman	2	4	6300.0
4	Thomassor	1	2	8000.0
5	Williams	2	2	4000.0
6	Wong	2	3	7700.0
7	Bornstein	2	3	8650.0
8	Harrison	2	2	4575.0
9	Horner	1	4	6375.0
10	Tanaka	1	5	9000.0

Leave the table gadget as it is, as you will be using it in the next section.

> ***Note:*** *You can always refresh your table by clicking the table in Active editing mode to give it the focus and then pressing the* ***F9*** *key. The F9 key clears the data source cache and then updates the table by querying the database. The display will thus reflect the current state of the database. If your keyboard does not have an F9 key, do the following:*
>
> 1. *Drag the database navigator gadget (the one with ten buttons) from the Data Access palette to the Main window.*
> 2. *In Selection mode, click the gadget to open its DbNavigator inspector.*
> 3. *Connect the gadget to the data source by selecting the data source in the combo box menu of the Data Source field.*
> 4. *Click the Active button in the Main window.*
> 5. *Click the* ***@*** *button of the database navigator gadget. The @ button works like the F9 key described above.*

Additional Information about Editing Tables

Data Access provides a table gadget for creating and editing tables. The figure below shows the different parts of this table gadget.



EMPNO	NAME	STATUS	DEPT
2	Fernandez	Permanent	Administration
1	Rochette	Temporary	

To select a cell for editing

1. Click the cell.

or

2. Press the Tab key to move forward from cell to cell or Shift-Tab to move backwards.

To edit cells

1. Type the data desired in the cell.

or

2. If the cell has a combo box, select an item from a pull-down menu

or

3. If the completion function is provided for the cell, type the first unique letters of a menu item without opening the menu. Then press the Return key or leave the cell.

To open a menu

When you click a cell, a combo box button appears if the cell has a combo box. To open the pull-down menu:

1. Click the button.

or

2. Press the F4 key.

To select a menu item

1. Click the item.

or

2. Use the Up/Down arrow keys to select and press Return.

To close a menu

Click anywhere or press the Escape key.

To validate edits

1. Press Return to keep the selection made in a cell and move to the next cell. If the cell is the last one in a row, you validate the entire row and move to the next row.

or

2. Click the Row Select button.

or

3. Move to another row.

A right-arrow symbol appears on the Row Select button of a validated row when any cell in that row is selected. When a row is being edited, but is not yet validated, the Row Select button has three dots (...).

To cancel edits

If the edits are not yet validated, press the Escape key to revert to the original state of the row.

To select an entire row or column

Click a Row or Column Select button.

To select a set of rows or a set of columns

To select more than one row or column in a table gadget, the Multi Selection option must be set to Yes in the Table Gadget inspector. Once this is set, you can select more than one column/row by clicking and dragging across the Row/Column Select buttons. Alternatively, you can select a set of adjacent rows or columns by clicking on the first Row/Column Select button, holding down the Shift key, and clicking the last Row/Column Select button in the desired set. The rows/columns between and including the first and last will be selected.

To resize a column

In the Column Select Buttons row, place the cursor on a line defining a column. The cursor changes to a double arrow. Drag the line right or left to resize the column.

To move a column or row (when enabled)

Click the Column Select or Row Select button of the column or row to be moved. Then click the button again and drag the column or row to the desired location in the table. This option is only available if the Column Geometry option is set to Local in the Table Gadget inspector.

To move a set of columns or rows (when enabled)

Select a set of rows or columns. To move them, click the button again and drag the complete set to a new location in the table. This option is only available if the Column Geometry option is set to Local in the Table Gadget inspector.

To select the entire table

Click the Select All button.

To refresh the data source cache and its display

Click the table while in Active mode. Then press the F9 key or, if you are using the database navigator gadget, click the @ button.

To sort the rows in a table (when enabled)

The rows in a table can be sorted according to the values in a particular column. Click the Column Select button of the required column. A small arrow appears indicating that the table has been sorted in ascending order. The option is only available if the Column Sort option is set to `Yes` in the Table Gadget inspector.

Using Selection and Display Criteria

You are now going to modify how the list of employees appears in the table gadget of the `employdb` application. You want to display only the employees whose employee number is less than 8, in alphabetical order by name and without their salary information.

1. Make sure you are in Selection editing mode ↖ . Click the EMP_DS data source gadget (or double-click it if no inspector panel is open).

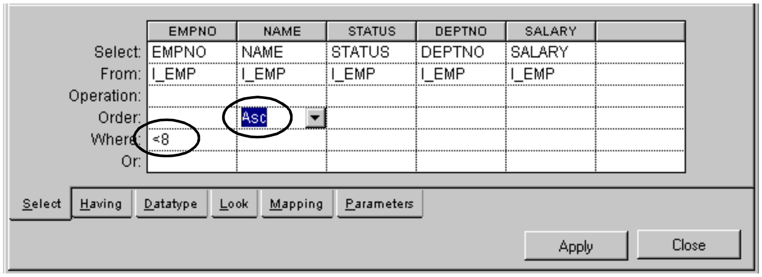
   The SQL Data Source inspector appears.

2. On the Select page:

   - Type `<8` in the Where cell of the EMPNO column to list only those employees whose employee number is less than 8.

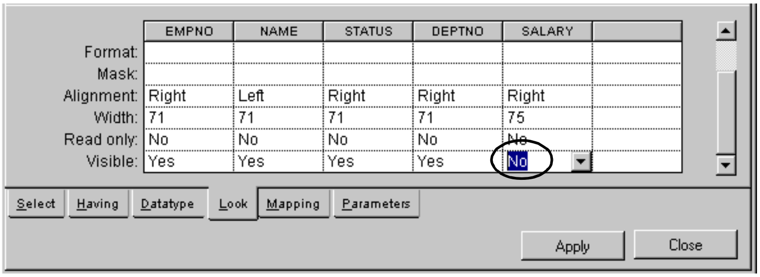
- Choose `Asc` (ascending) in the Order cell of the NAME column to sort the entries in alphabetical order by name.

> ***Note:** Selecting `No` from the Order cell combo box menu leaves the cell empty. If the cell is left empty, the default is used, which is a random order.*



	EMPNO	NAME	STATUS	DEPTNO	SALARY	
Select:	EMPNO	NAME	STATUS	DEPTNO	SALARY	
From:	I_EMP	I_EMP	I_EMP	I_EMP	I_EMP	
Operation:						
Order:		Asc				
Where:	<8					
Or:						

3. Click the Look page and change `Yes` to `No` in the Visible cell of the Salary column to prevent the SALARY column from being displayed. You may have to use the scroll bar on the left to move down through the rows of the table so that you can see the Visible row.



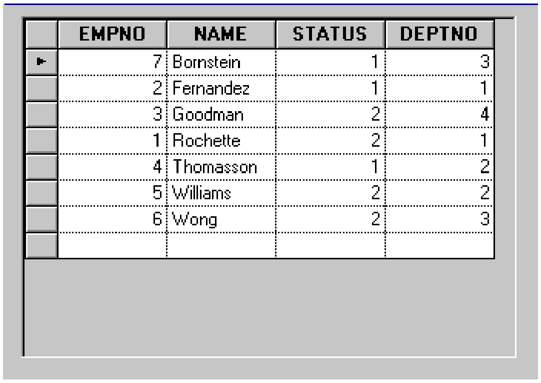
	EMPNO	NAME	STATUS	DEPTNO	SALARY	
Format:						
Mask:						
Alignment:	Right	Left	Right	Right	Right	
Width:	71	71	71	71	75	
Read only:	No	No	No	No	No	
Visible:	Yes	Yes	Yes	Yes	No	

4. Click Apply in the EMP_DS SQL Data Source inspector.

   The salary column disappears from the table gadget in the Main window.

5. Click the Active button in the Main window, then press the F9 key to update the data source.

   The table gadget changes to display only employees whose employee numbers are from 1 to 7, in alphabetical order by name.



	EMPNO	NAME	STATUS	DEPTNO
►	7	Bornstein	1	3
	2	Fernandez	1	1
	3	Goodman	2	4
	1	Rochette	2	1
	4	Thomasson	1	2
	5	Williams	2	2
	6	Wong	2	3

Additional Information about Selection Criteria

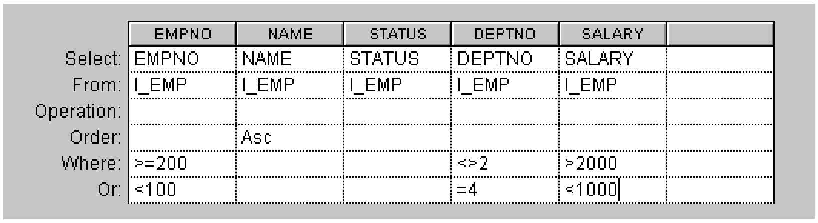
You can use the following operators in the SQL Data Source inspector to define selection criteria used by the data source to retrieve values from the database tables. Use the Select page when no Operation is defined in any of the columns; otherwise, use the Having page. The following expressions can be used with most database management systems. You can also use any expression supported by your particular system.

`=    4`	Equals
`<>   ‘Smith'`	Different than
`<    56`	Less than
`<=   8979`	Less than or equal to
`>    78`	Greater than
`>=   876`	Greater than or equal to
`LIKE 'ABCD%'`	(all words beginning with ABCD)
`LIKE 'A_B'`	(all words where “_” equals any one character)
`between 1 AND 5`	(all numbers between 1 and 5)

Where row The criteria typed in the Where cell of each column is added with an SQL logical AND operator to WHERE criteria in other columns to further restrict the selection. If a cell is left empty, the column is not taken into account for defining the selection criteria.

Or rows When you add criteria in a Where cell and validate it, a new row labeled “Or” appears below the Where row. The cells in the Or row are used just like those in the Where row. An SQL logical AND operator is used with each cell to further restrict the criteria. To provide further choices, an SQL logical OR operator is used between each of the Where and Or rows. Multiple Or rows are possible.

Here is an example of using the Where and Or rows:



	EMPNO	NAME	STATUS	DEPTNO	SALARY	
Select:	EMPNO	NAME	STATUS	DEPTNO	SALARY	
From:	I_EMP	I_EMP	I_EMP	I_EMP	I_EMP	
Operation:						
Order:		Asc				
Where:	>=200			<>2	>2000	
Or:	<100			=4	<1000	

The Where and Or rows are read as follows:

Select all the employees whose employee number is greater than or equal to 200,

AND whose department number is not 2,

AND whose salary is greater than 2000;

OR

all the employees whose employee number is less than 100,

AND whose department number equals 4,

AND whose salary is less than 1000.

Since the Name cell is left blank, any employee name can be selected if all the other criteria are met.

Saving the Window

If you have not already saved the window, do so now by choosing Save As... from the File menu in the Main Window. Give it the name `employee.ilv` and save it in the directory in which you saved the `.iva` application file *Creating the Application File* on page 50.

For information on saving window, see the section *Saving and Opening Windows* on page 53.

Defining the STATUS_MDS Memory Data Source

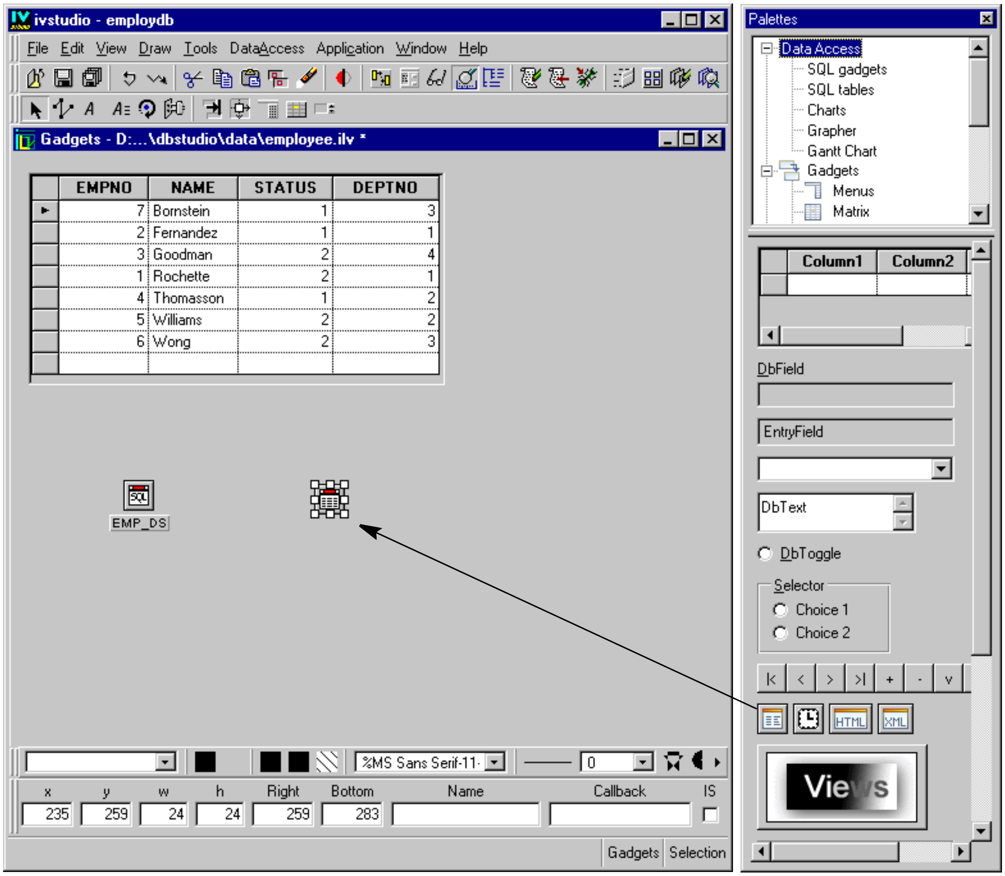
In our example, the STATUS column of the I_EMP table contains either the number `1` or `2` for each employee. It is preferable, however, to give a name to these numbers for display purposes by replacing `1` with `Permanent` and `2` with `Temporary`.

By creating a memory data source containing the numbers and their corresponding names and connecting it to a table gadget, you can temporarily change the display of the I_EMP data source table through *mapping.*

You use the Memory Data Source gadget, dragged from the Data Access palette of the Palettes panel, to create a table in memory. This table is used for making data available on a temporary basis, which does not need to be stored in a database. Of course, any table you create in a memory data source could also be created in a database. However, the memory data source provides a simple solution when you only need a temporary table and do not want to modify the database schema. You cannot use the SQL query language with the memory data source database, but you can *map* cells between data source and memory data source tables.

To define a memory data source, follow these steps:

1. In Selection editing mode ↖ , drag the Memory Data Source gadget from the Data Access palette of the Palettes panel to the Main window.



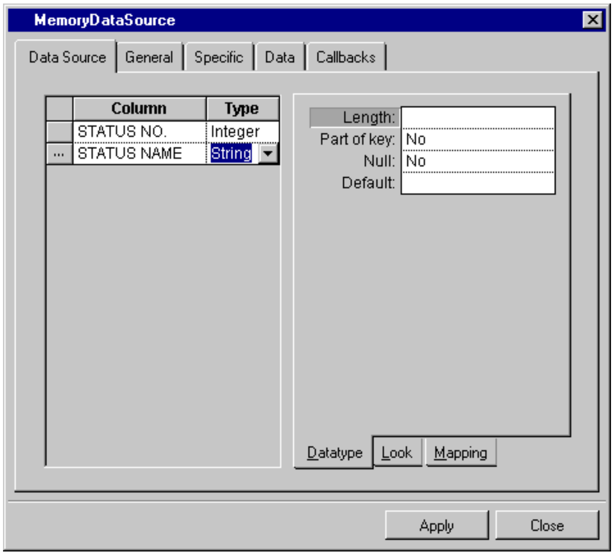
EMPNO	NAME	STATUS	DEPTNO
7	Bornstein	1	3
2	Fernandez	1	1
3	Goodman	2	4
1	Rochette	2	1
4	Thomasson	1	2
5	Williams	2	2
6	Wong	2	3

x	y	w	h	Right	Bottom	Name	Callback	IS
235	259	24	24	259	283			□

2. Click the Memory Data Source gadget (or double-click it if no inspector panel is open).

   The Memory Data Source inspector appears.

3. In the Memory Data Source inspector, the Data Source notebook page should be displayed. Fill in the Column and Type columns of the table on the left as shown below.

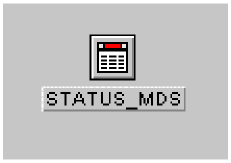


Column	Type
STATUS NO.	Integer
STATUS NAME	String

Length:
Part of key: No
Null: No
Default:

4. Go to the General notebook page of the Memory Data Source inspector. Type `STATUS_MDS` in the Name field. Click Apply.

   The Memory Data Source inspector closes and `STATUS_MDS` appears under the Memory Data Source gadget in the Main window.



You have now defined a memory data source.

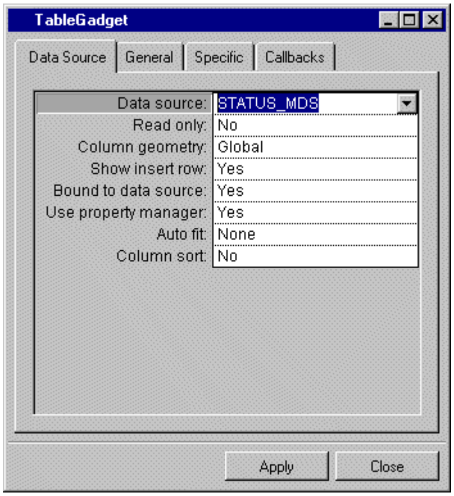
Connecting a Table Gadget to the STATUS_MDS Data Source

To connect a table gadget to the STATUS_MDS memory data source, do the following:

1. Drag a table gadget from the Data Access palette to the Main window.

   The Table Gadget inspector appears in place of the Memory Data Source panel.

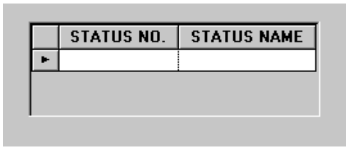
2. In the Table Gadget inspector, use the combo box menu to enter `STATUS_MDS` in the Data source field. This connects the table gadget to the memory data source. Click Apply.



Data source:	STATUS_MDS
Read only:	No
Column geometry:	Global
Show insert row:	Yes
Bound to data source:	Yes
Use property manager:	Yes
Auto fit:	None
Column sort:	No

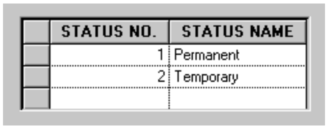
The column headers in the table gadget in the Main window become the same as those in the STATUS_MDS memory data source.

3. In the Main window, click the Active editing mode button . Resize the table gadget columns so that the headers will fit.



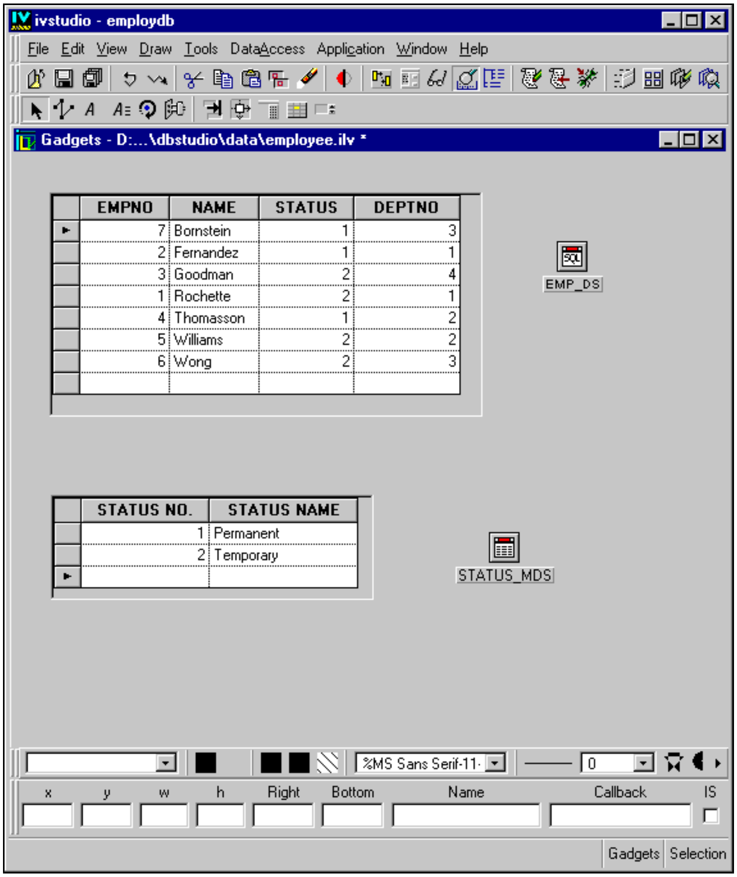
STATUS NO.	STATUS NAME

4. Fill in the table so it looks like the following figure. Once you have entered the data, validate by pressing Enter.



STATUS NO.	STATUS NAME
1	Permanent
2	Temporary

You have now connected a table gadget to the STATUS_MDS memory data source and added data to it. The Main window should now look like this:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
EMP_DS
STATUS_MDS
%MS Sans Serif-11-
0
x
y
w
h
Right
Bottom
Name
Callback
IS
Gadgets
Selection

EMPNO	NAME	STATUS	DEPTNO
7	Bornstein	1	3
2	Fernandez	1	1
3	Goodman	2	4
1	Rochette	2	1
4	Thomasson	1	2
5	Williams	2	2
6	Wong	2	3

STATUS NO.	STATUS NAME
1	Permanent
2	Temporary

Mapping Cells Between Data Sources

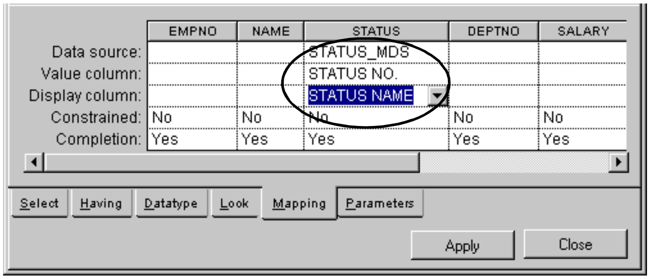
You are now going to map the status numbers `1` and `2` in the EMP_DS SQL data source to the names `Permanent` and `Temporary` in the STATUS_MDS memory data source.

1. Click the Selection mode button ↖. Click the EMP_DS SQL data source gadget in the Main window.

   The SQL Data Source inspector appears.

2. Click the Mapping tab of the SQL Data Source inspector so the Mapping page becomes active.

   You are now going to fill in the cells of the STATUS column so it looks like this:



	EMPNO	NAME	STATUS	DEPTNO	SALARY
Data source:			STATUS_MDS		
Value column:			STATUS NO.		
Display column:			STATUS NAME		
Constrained:	No	No	No	No	No
Completion:	Yes	Yes	Yes	Yes	Yes

- Enter `STATUS_MDS` in the Data source cell of the STATUS column.

  This is the data source from which the STATUS cell will get the information to display.

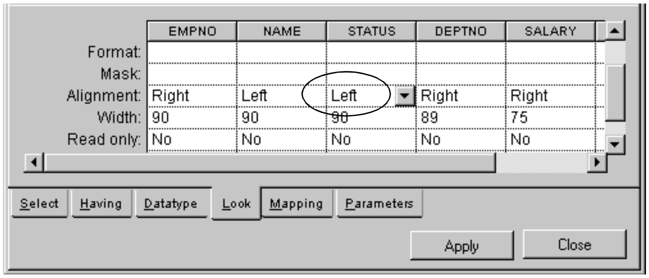
- Enter `STATUS NO.` in the Value column cell of the STATUS column.

  This is the column in the STATUS_MDS memory data source to which the value of the STATUS cell for the EMP_DS data source is matched.

- Enter `STATUS NAME` in the Display column cell of the STATUS column.

  This is the column in the STATUS_MDS data source from which the name corresponding to the value in the Value column is taken. This name is displayed in the STATUS cell of each employee.

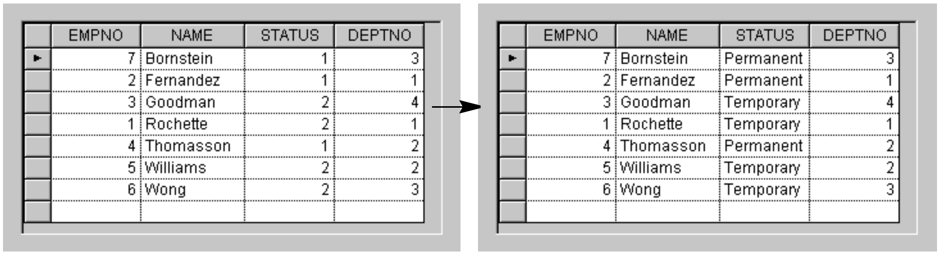
3. Because numbers are aligned right and text is aligned left, you need to change the alignment of the STATUS column. Click the Look page to activate it. Then change the Alignment cell of the STATUS column to `Left`.



	EMPNO	NAME	STATUS	DEPTNO	SALARY
Format:					
Mask:					
Alignment:	Right	Left	Left	Right	Right
Width:	90	90	90	89	75
Read only:	No	No	No	No	No

4. Click Apply.

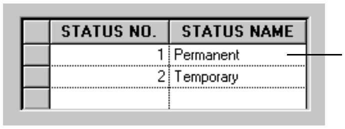
   In the Main window, the numbers in the STATUS column of the I_EMP table gadget are replaced by the names mapped from the STATUS_MDS memory data source.



EMPNO	NAME	STATUS	DEPTNO
7	Bornstein	1	3
2	Fernandez	1	1
3	Goodman	2	4
1	Rochette	2	1
4	Thomasson	1	2
5	Williams	2	2
6	Wong	2	3

EMPNO	NAME	STATUS	DEPTNO
7	Bornstein	Permanent	3
2	Fernandez	Permanent	1
3	Goodman	Temporary	4
1	Rochette	Temporary	1
4	Thomasson	Permanent	2
5	Williams	Temporary	2
6	Wong	Temporary	3

5. Since you do not need to keep the table in the window, delete the table gadget connected to the STATUS_MDS data source. In Selection mode, click the table gadget and press the Delete key. The data source is not affected.



STATUS NO.	STATUS NAME
1	Permanent
2	Temporary

Delete this table gadget

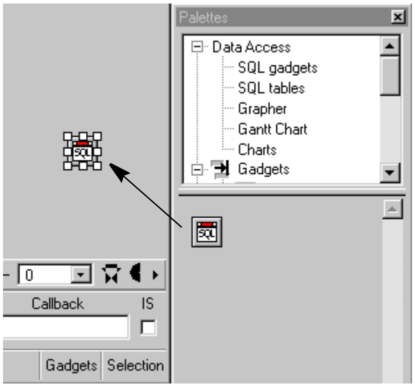
The table gadget disappears.

6. Save the Employee window by clicking the Save button in the Main window.

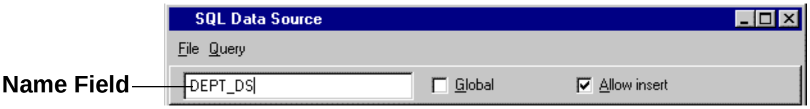
Defining the DEPT_DS SQL Data Source

You are now going to define another data source, this one based on the I_DEPT table in the Employee database.

1. Drag the SQL Data Source gadget from the SQL gadgets palette to the Main window.



2. Double-click the SQL Data Source gadget you just dragged to open the SQL Data Source inspector. Type `DEPT_DS` in the Name field at the top of the SQL Data Source inspector.



3. Choose Add Tables... from the Query menu of the SQL Data Source inspector panel.

   The Connect panel appears.

4. Select `Main` from the Name combo box of the Connect panel as you did for the EMP_DS data source in the section *Defining the EMP_DS SQL Data Source* on page 57. The Connect panel changes to show only this name.

5. Click OK.

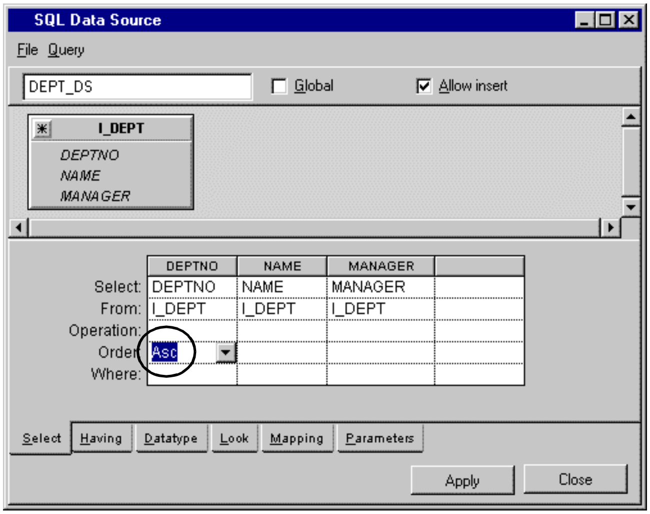
   The Select Tables panel appears.

6. Select the name I_DEPT in the list on the left of the panel and click the -> button to add it to the right list. Then click OK.

   The I_DEPT table appears in the FROM section of the SQL Data Source inspector.

7. Create three columns in the SELECT section by dragging from the asterisk in the top left corner of the database table in the FROM section to the column in the SELECT section.

8. On the Select page, choose `Asc` from the combo box menu in the Order cell of the DEPTNO column.



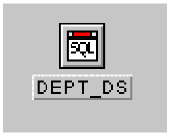
DEPT_DS ☐ Global ☑ Allow insert

I_DEPT
DEPTNO
NAME
MANAGER

	DEPTNO	NAME	MANAGER	
Select:	DEPTNO	NAME	MANAGER	
From:	I_DEPT	I_DEPT	I_DEPT	
Operation:				
Order:	Asc			
Where:				

9. Click Apply in the SQL Data Source inspector.

   `DEPT_DS` appears under the SQL Data Source gadget in the Main window.



You have now defined the DEPT_DS data source.

Connecting a Table Gadget to the DEPT_DS Data Source

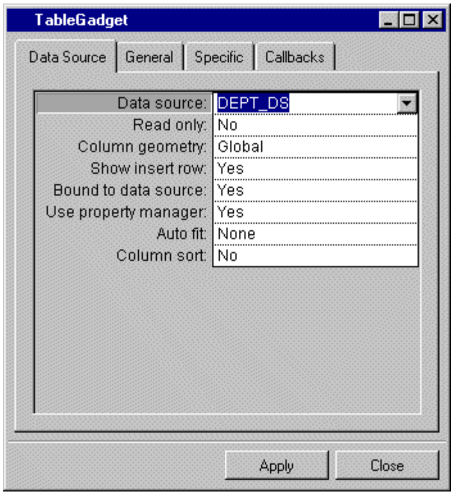
To connect a table gadget to the DEPT_DS data source, do the following:

1. Drag a table gadget from the Data Access palette to the Main window.

2. Double-click the table gadget.

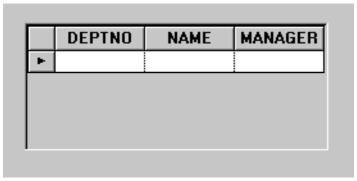
   The Table Gadget inspector appears.

3. Use the combo box to enter DEPT_DS in the Data source cell of the Table Gadget inspector. Click Apply.



Data source: DEPT_DS
Read only: No
Column geometry: Global
Show insert row: Yes
Bound to data source: Yes
Use property manager: Yes
Auto fit: None
Column sort: No

The columns in the table gadget in the Main window change to match those of the DEPT_DS data source table.

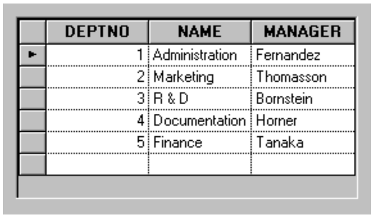


	DEPTNO	NAME	MANAGER
►			

4. In the Main window, click the Active button in the Editing Mode toolbar at the top.

   You can now edit the table gadget.

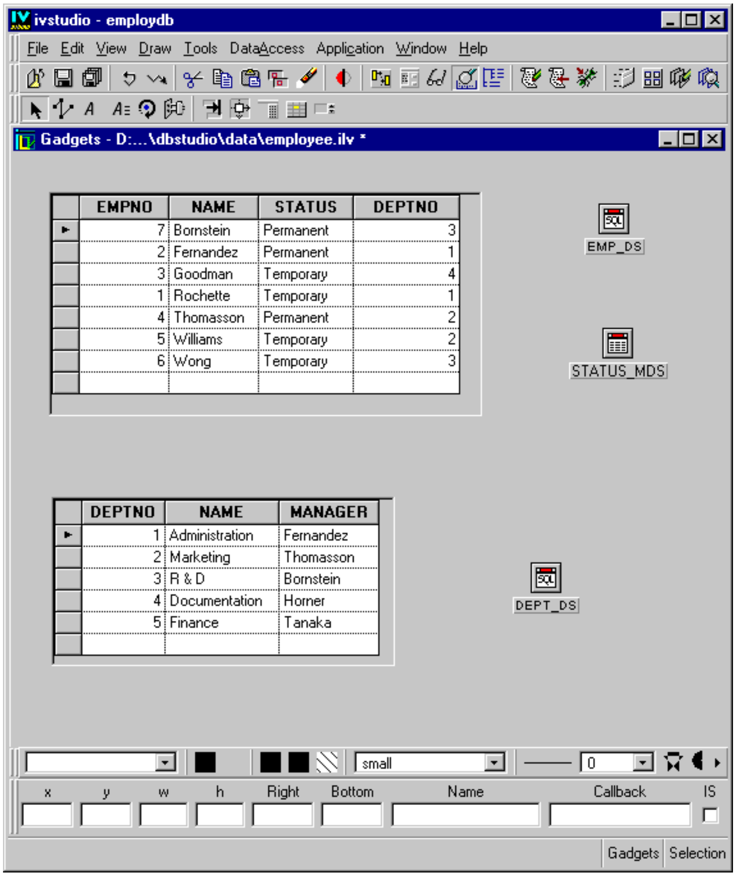
5. Fill in the table gadget cells as shown in the following figure. The data is automatically entered in the database. For information on editing tables, see *Additional Information about Editing Tables* on page 67.



	DEPTNO	NAME	MANAGER
►	1	Administration	Fernandez
	2	Marketing	Thomasson
	3	R & D	Bornstein
	4	Documentation	Horner
	5	Finance	Tanaka

Leave the table gadget as it is, as you will be using it in the next section.

Your Main window should now be similar to this:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
EMP_DS
STATUS_MDS
DEPT_DS
small
0
x y w h Right Bottom Name Callback IS
Gadgets Selection

	EMPNO	NAME	STATUS	DEPTNO
►	7	Bornstein	Permanent	3
	2	Fernandez	Permanent	1
	3	Goodman	Temporary	4
	1	Rochette	Temporary	1
	4	Thomasson	Permanent	2
	5	Williams	Temporary	2
	6	Wong	Temporary	3

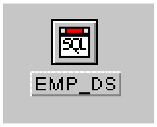
	DEPTNO	NAME	MANAGER
►	1	Administration	Fernandez
	2	Marketing	Thomasson
	3	R & D	Bornstein
	4	Documentation	Horner
	5	Finance	Tanaka

Mapping and Formatting the Cells

In this example, each employee works in a department that has a given number. However, you would rather display the department name instead of its number. You want to replace the DEPTNO column numbers with names.

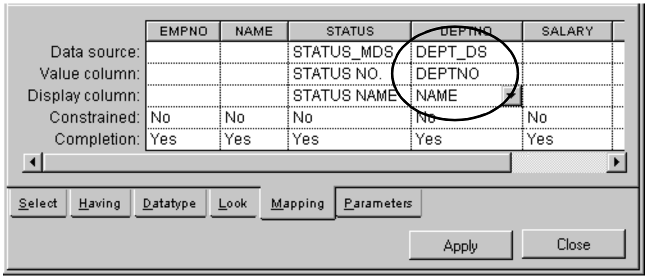
In this section, you are going to map the cells of the DEPTNO column of the EMP_DS data source to the NAME cells of the DEPT_DS data source. To do this, follow these steps:

1. In Selection mode ↖ , double-click the EMP_DS SQL Data Source gadget to open its inspector.



2. Select the Mapping page of the inspector.

3. Fill in the DEPTNO column cells as shown below:

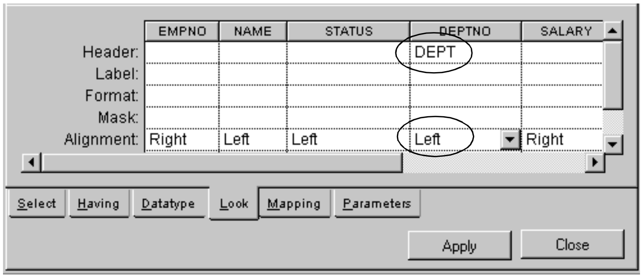


	EMPNO	NAME	STATUS	DEPTNO	SALARY
Data source:			STATUS_MDS	DEPT_DS	
Value column:			STATUS NO.	DEPTNO	
Display column:			STATUS NAME	NAME	
Constrained:	No	No	No	No	No
Completion:	Yes	Yes	Yes	Yes	Yes

4. Because you are going to display the department name instead of the number, you need to change the column header. Select the Look page in the SQL Data Source inspector and change the Header cell of the DEPTNO column from `DEPTNO` to `DEPT`.

5. Because numbers are aligned right and text is aligned left, change the Align cell in the DEPTNO column from `Right` to `Left`.

   The SELECT section of the Look page in the SQL Data Source inspector panel now looks like this:

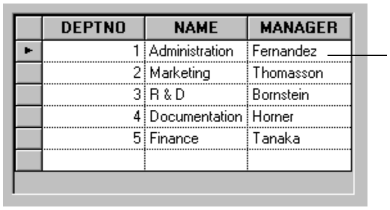


	EMPNO	NAME	STATUS	DEPTNO	SALARY
Header:				DEPT	
Label:					
Format:					
Mask:					
Alignment:	Right	Left	Left	Left	Right

6. Click Apply in the SQL Data Source inspector.

   The header of the DEPTNO column of the Employee table gadget in the Main window changes to DEPT, while the numbers in the DEPT cells change to their corresponding names and are aligned left. If necessary, adjust the size of the columns.

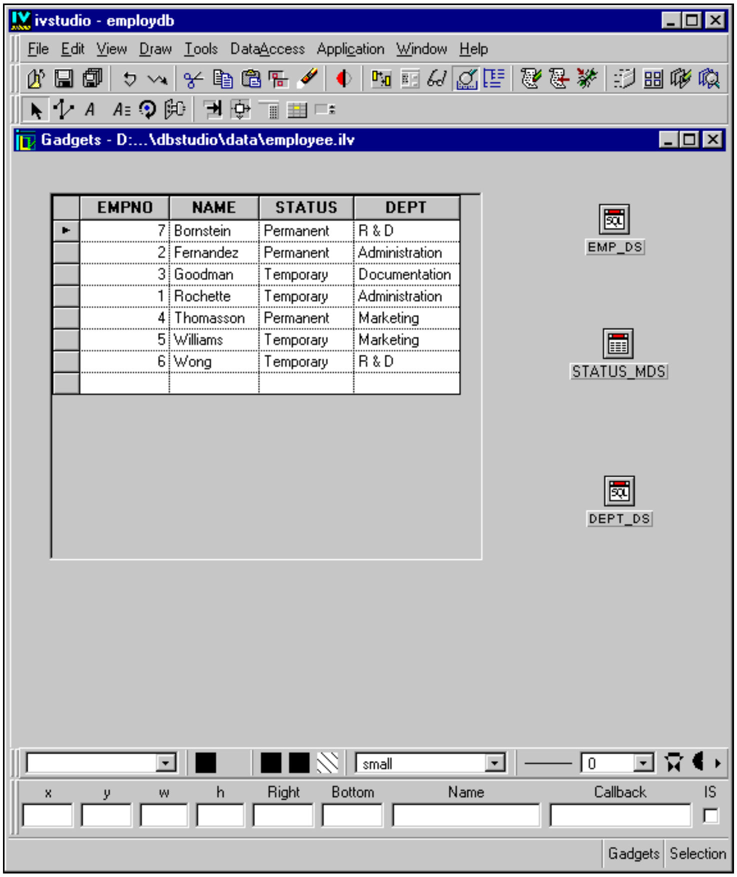
7. As you will not need it anymore, delete the table gadget connected to the DEPT_DS data source. Click the table gadget and press the Delete key.



	DEPTNO	NAME	MANAGER
►	1	Administration	Fernandez
	2	Marketing	Thomasson
	3	R & D	Bornstein
	4	Documentation	Horner
	5	Finance	Tanaka

Delete this table gadget

The Main window now looks like this:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv
EMP_DS
STATUS_MDS
DEPT_DS
small
0
x
y
w
h
Right
Bottom
Name
Callback
IS
Gadgets
Selection

	EMPNO	NAME	STATUS	DEPT
►	7	Bornstein	Permanent	R & D
	2	Fernandez	Permanent	Administration
	3	Goodman	Temporary	Documentation
	1	Rochette	Temporary	Administration
	4	Thomasson	Permanent	Marketing
	5	Williams	Temporary	Marketing
	6	Wong	Temporary	R & D

***Note:** A combo box is now available for the DEPT and STATUS columns of the EMP table gadget. This allows the user to change the department name or status for an employee by selecting an item from a list.*

8. Save the Employee window by clicking the Save button in the Main window.

Defining a Form

In this section, you are going to learn what a form is and how to define and connect it to the DEPT_DS data source.

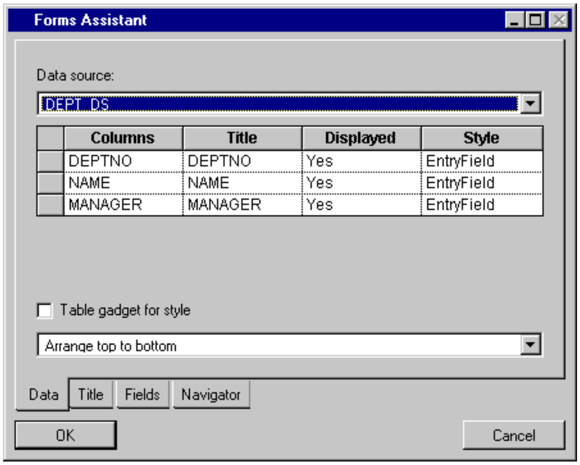
How to Define a Form

To define a form and connect it to the DEPT_DS data source, follow these steps:

1. In Selection mode , choose Forms Assistant from the Data Access menu of the Main window.

   The Forms Assistant panel appears.

2. Select the DEPT_DS data source from the combo box menu and click OK. The Forms Assistant Panel closes.



Data source: DEPT_DS

Columns	Title	Displayed	Style
DEPTNO	DEPTNO	Yes	EntryField
NAME	NAME	Yes	EntryField
MANAGER	MANAGER	Yes	EntryField

☐ Table gadget for style

Arrange top to bottom

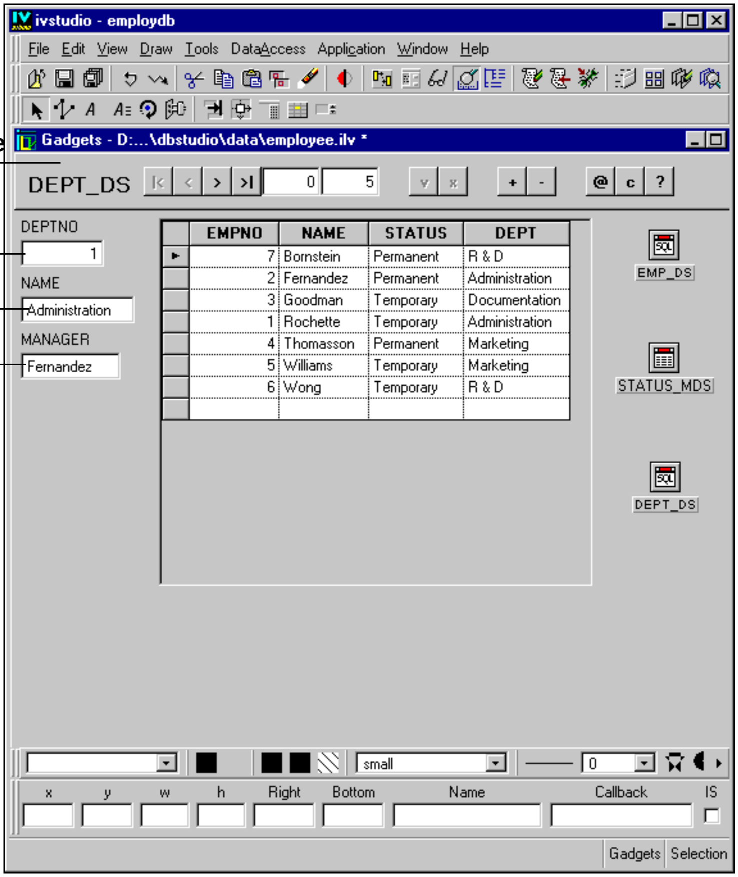
   The I_DEPT form appears in the Gadgets buffer of the Main window.

   There is a text field gadget for each of the columns in the data source: DEPTNO, NAME, and MANAGER.

   There is also a rectangle at the top of the screen that contains the data source name DEPT_DS as a caption and a group of 10 buttons. (See *Additional Information about Forms* on page 88 for a description of the buttons.) In addition, there are two fields that show you the current row of the data source table that is being displayed and the total number of rows in the data source table.

3. If necessary, rearrange the objects in the work space of the Main window.

   The work space should now show a table gadget connected to the EMP_DS data source and a form connected to the DEPT_DS data source.


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
EMP_DS
STATUS_MDS
DEPT_DS
small
x y w h Right Bottom Name Callback IS
Gadgets Selection

Form Rectangle with Caption and Buttons

Form Text Fields

DEPT_DS |< < > >| 0 5 v x + - @ c ?

DEPTNO: 1

NAME: Administration

MANAGER: Fernandez

	EMPNO	NAME	STATUS	DEPT
►	7	Bornstein	Permanent	R & D
	2	Fernandez	Permanent	Administration
	3	Goodman	Temporary	Documentation
	1	Rochette	Temporary	Administration
	4	Thomasson	Permanent	Marketing
	5	Williams	Temporary	Marketing
	6	Wong	Temporary	R & D

4. In Active mode , click the “@” button to refresh the data source and go to the first record.

The DEPTNO, NAME and MANAGER text field gadgets of the I_DEPT form display the data for the first record in the DEPT_DS data source.

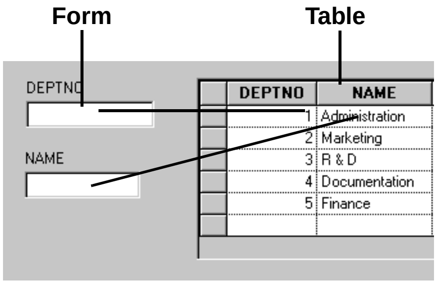
5. Click the “+” button.

The DEPTNO, NAME and MANAGER fields of the I_DEPT form become empty and ready for editing. At this point, you could enter a new department if you wanted, but leave the table as it is for now.

Additional Information about Forms

A form is a set of gadgets that are used to show one data source row (record) at a time in a more flexible way than is possible with a table gadget.

You use a form by connecting each text field gadget of a form to a column in a particular data source. The column cells of the current record can then be displayed in the gadgets placed in the window as you wish.



DEPTNO	NAME
1	Administration
2	Marketing
3	R & D
4	Documentation
5	Finance

To access the Forms feature, choose Forms Assistant... from the Data Access menu in the Main window. When the Forms Assistant panel appears and you select the data source to which the form is to be connected, the Main window appears as follows:

Text Field Form

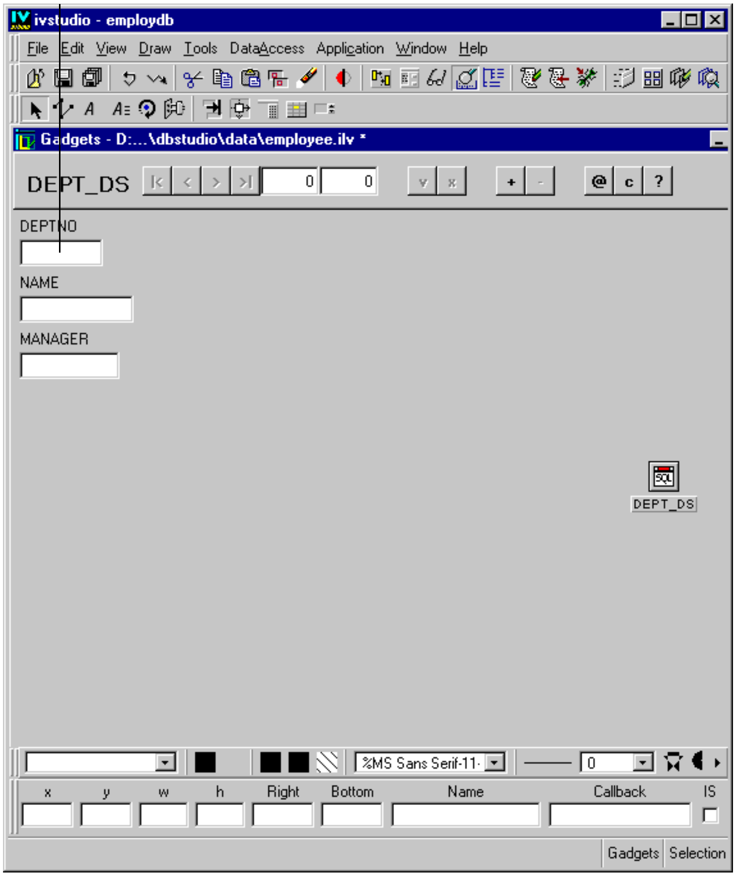
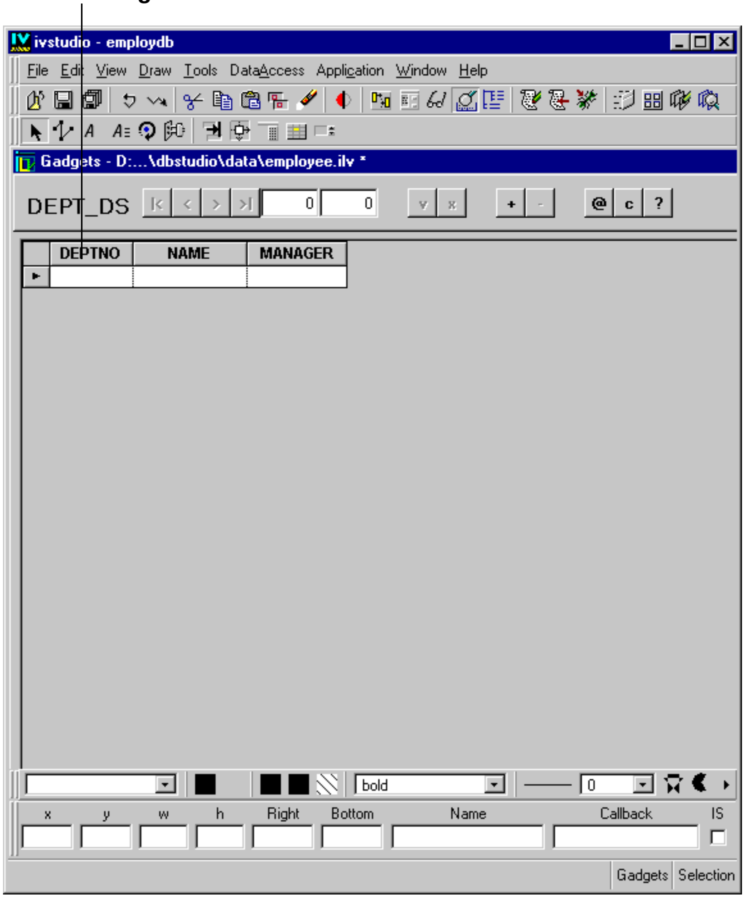


Table Gadget Form


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
DEPT_DS
|< < > >| 0 0 v x + - @ c ?
DEPTNO NAME MANAGER
bold
0
x y w h Right Bottom Name Callback IS
Gadgets Selection

A form consists of the following elements:

- A text field for each image in the data source table or a table gadget (if you selected the Table gadget for style check box in the Forms Assistant panel).
- A rectangle at the top of the window. This rectangle contains several buttons and fields:
  - A caption on the left side of the rectangle.
  - 10 buttons (and one not visible) on the rectangle, that are:

|<	Go to first record.
<	Go to previous record.
>	Go to next record.
>|	Go to last record.

`v`	Validate entry (create in database) or apply query mode.
`x`	Cancel entry (revert to previous state) or apply query mode.
`+`	If Allow Insert was activated in the SQL Data Source inspector panel, go to the insert row to add a new record.
`-`	Delete current record.
`@`	Clear the data source cache, update it by querying the database, then refresh the display (same as F9 for tables).
`c`	Clear the data source cache.
`?`	Change to Query mode (not displayed by default)

- 2 fields on the rectangle (that represent the current position and the number of lines)

All the above elements can be moved and resized in Selection editing mode.

> ***Note:*** *You cannot type a value in a form field to display the record to which it belongs. You must use the edit buttons to move around the database. Data should be typed in a form field only when you want to edit or add data.*

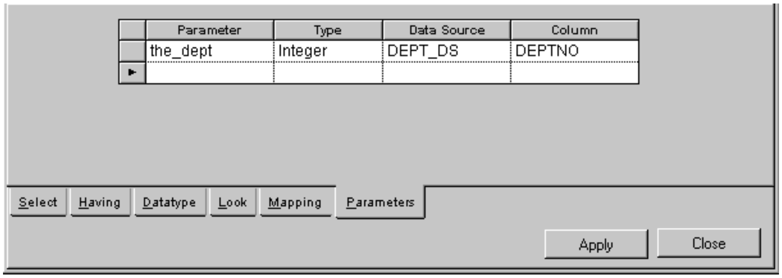
Using Parameters to Link Data Sources

You are now going to use the Parameters notebook page to link the EMP_DS and DEPT_DS data sources so that only employees for the current department are listed. To do this, follow these steps:

1. In Selection mode , double-click the EMP_DS SQL data source gadget in the Main window.

   The SQL Data Source inspector for the EMP_DS data source appears.

2. Click the Parameters page to make it active. Fill in the table cells as shown below:

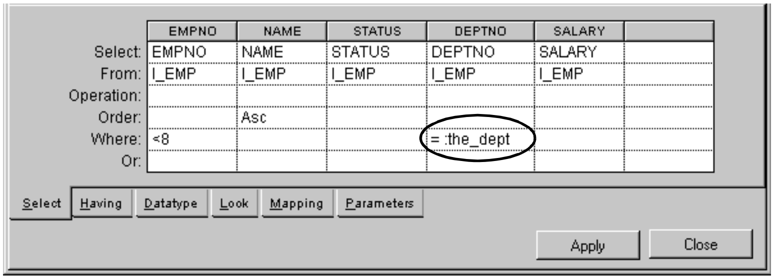


Parameter	Type	Data Source	Column
the_dept	Integer	DEPT_DS	DEPTNO

The table can be read as follows: the parameter `the_dept` of type `integer` matches the `DEPTNO` column of the `DEPT_DS` data source.

You can now use the parameter `the_dept` as criteria for selecting employees from the I_EMP table.

3. Click the Select page in the SQL Data Source inspector. Type the parameter in the Where row of the DEPTNO column like this: `= :the_dept`



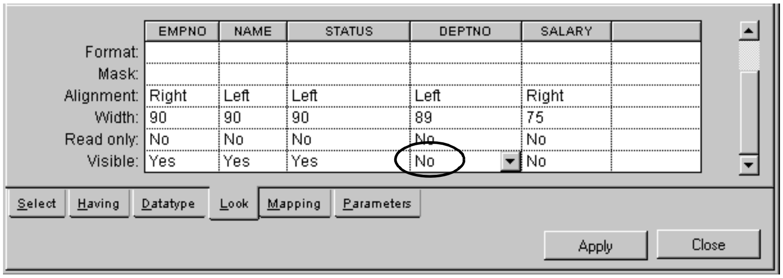
	EMPNO	NAME	STATUS	DEPTNO	SALARY	
Select:	EMPNO	NAME	STATUS	DEPTNO	SALARY	
From:	I_EMP	I_EMP	I_EMP	I_EMP	I_EMP	
Operation:						
Order:		Asc				
Where:	<8			= :the_dept		
Or:						

Note: *Always preface a parameter with a colon without a space.*

4. If necessary, adjust the size of the DEPTNO column.

5. You will no longer need to have the DEPTNO column visible in the I_EMP table since all the employees selected will be in the current department displayed in the form field.

   Click the Look page in the SQL Data Source inspector. Change the Visible cell of the DEPTNO column to `No`.



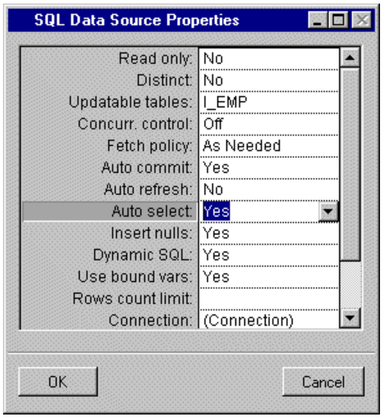
	EMPNO	NAME	STATUS	DEPTNO	SALARY
Format:					
Mask:					
Alignment:	Right	Left	Left	Left	Right
Width:	90	90	90	89	75
Read only:	No	No	No	No	No
Visible:	Yes	Yes	Yes	No	No

6. For this operation to work, you need to set the automatic selection of items for data sources using parameters to refer to items in other data sources.

   Choose Properties... in the File menu of the SQL Data Source inspector.

   The SQL Data Source Properties panel appears.

7. In the SQL Data Source Properties panel, change the Auto select field from `No` to `Yes`. Click OK.



Read only:	No
Distinct:	No
Updatable tables:	I_EMP
Concurr. control:	Off
Fetch policy:	As Needed
Auto commit:	Yes
Auto refresh:	No
Auto select:	Yes
Insert nulls:	Yes
Dynamic SQL:	Yes
Use bound vars:	Yes
Rows count limit:	
Connection:	(Connection)

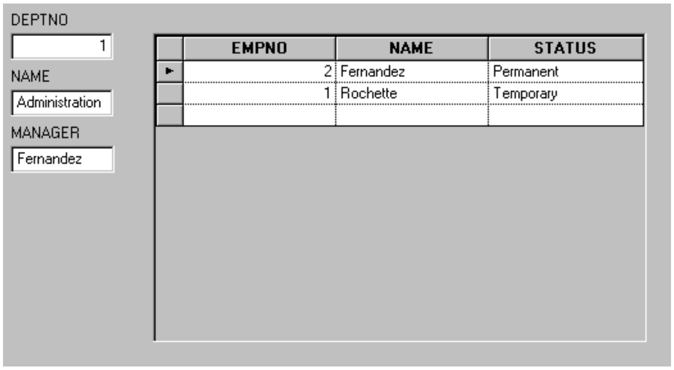
   The SQL Data Source Properties panel disappears.

8. Click Apply in the SQL Data Source inspector. Click Close.

   The SQL Data Source inspector disappears.

9. In Active mode , click the @ button in the Main window to go to the first record (Department 1) of the DEPT_DS data source.

   Only those employees in Department 1 are displayed in the I_EMP table.



DEPTNO

1

NAME

Administration

MANAGER

Fernandez

	EMPNO	NAME	STATUS
►	2	Fernandez	Permanent
	1	Rochette	Temporary

10. Click the **>** button to go the next record in the DEPT_DS data source.

Only those employees in Department 2 are displayed in the I_EMP table.

11. Save the Employee window by clicking the Save button in the Main window. Give it the name `employee.ilv` and save it in the directory in which you saved the `.iva` application file.

CHAPTER

4

Designing Windows

This chapter shows you how to set up the appearance of your window by using the various GUI features that Data Access provides.

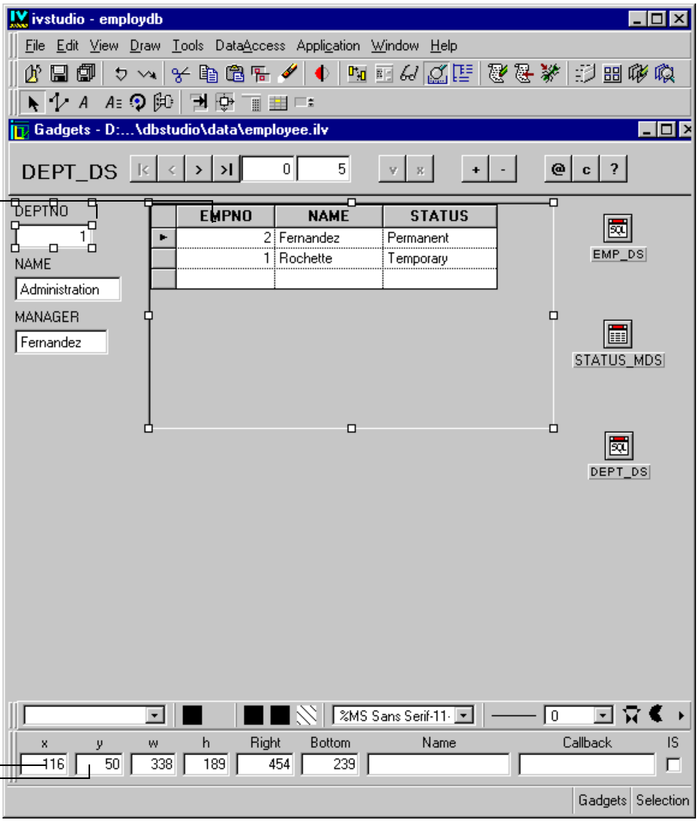
You will finish the Employee window by performing the following procedures:

- *Aligning Objects*
- *Using Resources*
- *Checking the Focus Chain*
- *Setting the Size of the Window*
- *Setting the Background of the Window*
- *Setting Attachments*
- *Testing Attachments*
- *Adding the Employee Window to the Application*

Aligning Objects

You are now going to align objects in the Employee window, as shown in the following figure:

Align tops of objects


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv
DEPT_DS
EMP_DS
STATUS_MDS
DEPT_DS
%MS Sans Serif-11-
Gadgets
Selection

DEPT_DS 0 5

DEPTNO 1

NAME Administration

MANAGER Fernandez

EMPNO	NAME	STATUS
2	Fernandez	Permanent
1	Rochette	Temporary

x	y	w	h	Right	Bottom	Name	Callback	IS
116	50	338	189	454	239			☐

X and Y Fields

1. If you closed the Employee window (`employee.ilv`) at the end of the previous chapter, open it again by clicking the Open button in the toolbar.

2. To line up the tops of the Employee table and the DEPTNO text field gadget, first click the DEPTNO text field gadget. Then shift-click the Employee table gadget to select both items.

> ***Note:*** *An object changes its characteristics to match that of the first item selected.*

3. From the Draw menu, choose Align/Distribute > Align Top.

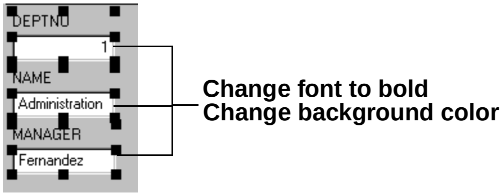
   The Employee table moves so that it is at the same height as the DEPTNO text field gadget.

> ***Note:*** *You can also use the X, Y, Right and Bottom fields in the Generic Inspector area to align objects. For example, if you put the same number in the X field for different objects, they will all line up with their left borders at the distance from the left margin you entered. In the same way, you can use the Y field to align the top borders of objects. The Right and Bottom fields can be used to line up the right and bottom borders of objects.*

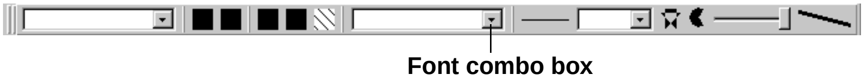
Using Resources

You are now going to change some fonts and colors of objects in the Employee window using the Resources editor.

1. In Selection editing mode, select the three form text field gadgets in the Main window.

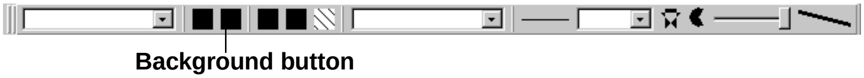


2. In the Resources editor, choose `bold` from the Font combo box menu.

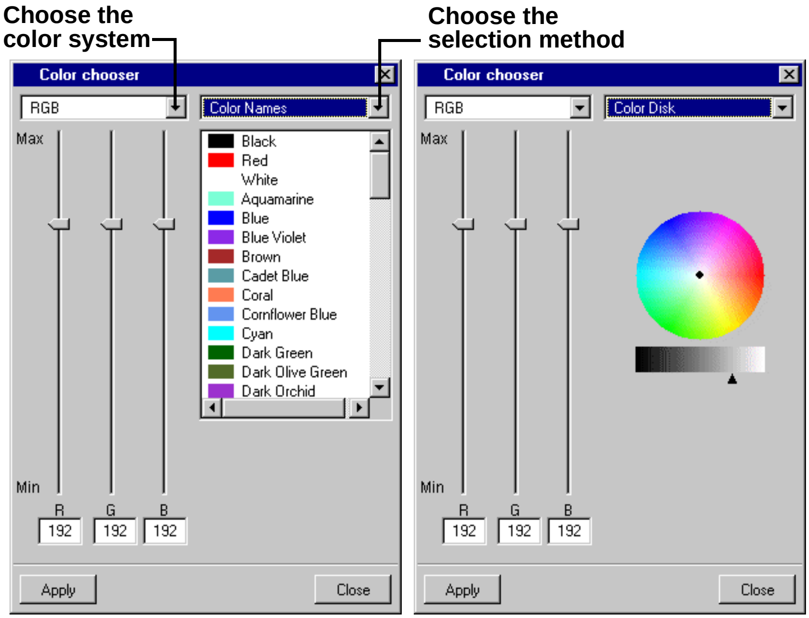


The text will now appear in bold type in the text fields. (You can verify this by clicking the Active editing mode button in the Main window and typing in the text field.)

3. Click the Background button in the Resource editor, then click the top-left colored filled circle.



The Color Chooser panel appears.


Choose the color system

Choose the selection method

4. At the top of the panel, choose the color system and/or selection method. Select the color you want. Use the RGB/HSV values and/or the color wheel to define your own colors or use the Color Names option to use predefined colors.

   RGB= Red, Green, Blue

   HSV = Hue, Saturation, Value

   The color selected appears in the lower-right rectangle of the Color Chooser panel.

5. Click Apply in the Color Chooser panel.

   The background of the text fields in the work space changes to the color you chose.

Checking the Focus Chain

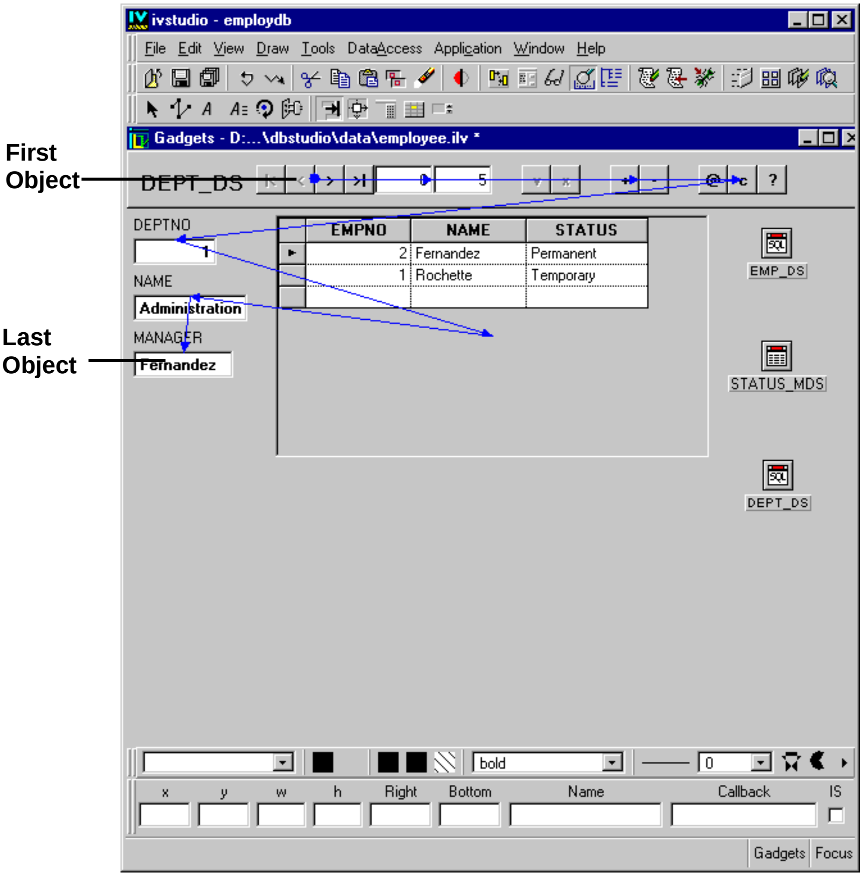
The focus chain is the order in which objects in the window own the keyboard focus as the Tab key is pressed repeatedly. When you click the Focus button in the Editing Modes toolbar, a default focus chain path appears in the Main window.

You can edit the focus chain so that the objects own the keyboard focus in any order you wish.

You are now going to change the focus chain so that the order goes through the three form fields from top to bottom, then back to the top.

1. Click the Focus button in the Main window.

   A default focus chain appears linking the objects in the work space.


First Object
Last Object
ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
DEPT_DS
EMP_DS
STATUS_MDS
DEPT_DS
bold
x y w h Right Bottom Name Callback IS
Gadgets Focus

DEPTNO 1

NAME Administration

MANAGER Fernandez

	EMPNO	NAME	STATUS
►	2	Fernandez	Permanent
	1	Rochette	Temporary

> ***Note:*** *Your default focus chain may not appear exactly as above due to how you manipulated your objects.*

2. Delete the arrows from each object by dragging a line from the object and releasing the mouse when the tip of the line is in the work space (but not on another object).

3. Drag a line from anywhere in the work space (but not from another object) to the DEPTNO text field object. When the object is highlighted, release the mouse button.

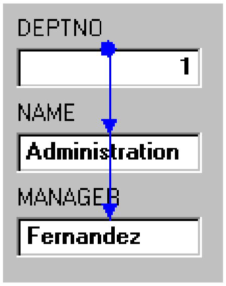
   A filled-in circle appears in the object, signifying that it has become the first focus object.

4. Drag a line from the DEPTNO text field object to the NAME text field object. When the NAME object becomes highlighted, release the mouse button.

   An arrow appears going from the DEPTNO object to the NAME object.

5. Drag a line from the NAME object to the MANAGER object.

   An arrow appears going from the NAME to MANAGER objects. You have now completed the focus chain, which should look like this:



DEPTNO
1
NAME
Administration
MANAGER
Fernandez

6. To test the focus chain, click the Test button in the toolbar.

   The Test panel appears.

7. Press the Tab key a few times in succession.

   The objects become selected in the order of the focus chain you created.

8. Click the Test button again to close the Test panel.

9. Click the Selection mode button in the Main window.

Setting the Size of the Window

There are two methods for resizing a window:

- **Fit to Contents... command** in the Edit menu of the Main window. (You will use this method in this section.)

- **Grid panel** accessed through the View Inspector command from the Tools menu in the Main window (see *Setting the Background of the Window* on page 104).

To adjust the size of the Employee window to fit around the objects, do the following:

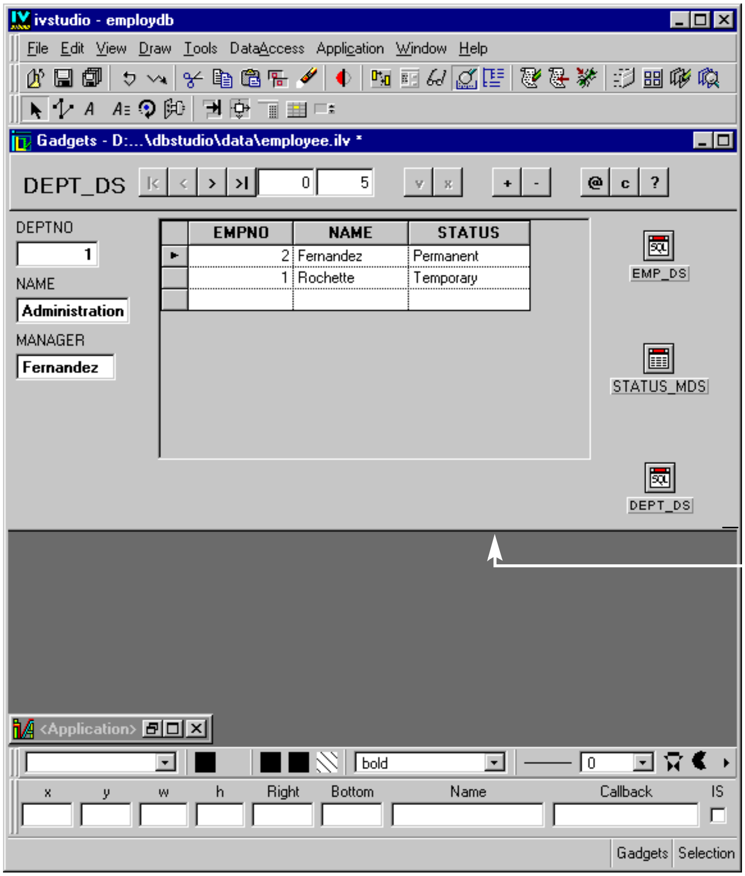
1. Since the data source and memory data source gadgets will not be displayed in your final window, make sure they are placed within the perimeter made up of the other objects so that they will not be taken into account when using the Fit to Contents... command.

2. Choose Fit to Contents... from the Edit menu in the Main window.

   A Prompt String panel appears requesting the size of the window margin.

3. Type `10` in the Margin field of the Prompt String panel, then click Apply.

   The size of the window adjusts to fit around the object with the margin you just specified as shown in the following figure:



DEPT_DS 0 5

DEPTNO 1

NAME Administration

MANAGER Fernandez

EMPNO	NAME	STATUS
2	Fernandez	Permanent
1	Rochette	Temporary

bold 0

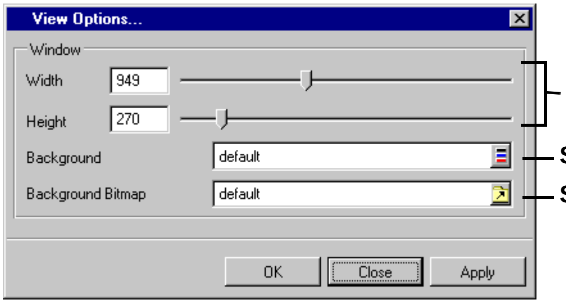
**Window fits aroun
objects with marg
specified in Promp
String panel**

> ***Note:*** *There is no right margin from the top form rectangle, because by default this object automatically takes on the width of the window.*

Setting the Background of the Window

You are now going to set the color or bitmap used as the background for the window. Choose View Inspector in the Tools menu of the Main window.

The View Options panel appears.



Width 949

Height 270

Background default

Background Bitmap default

Resize panel

Select panel background color

Select panel background bitmap

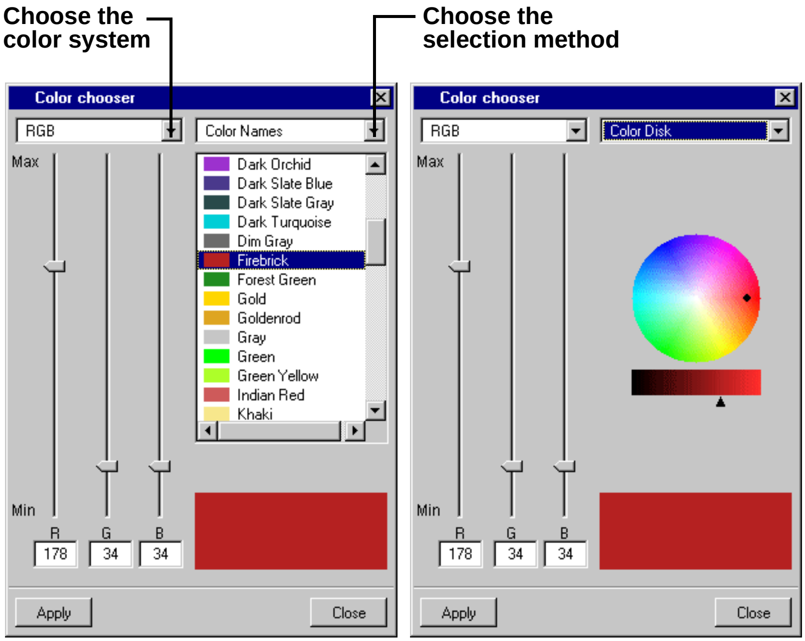
When you specify a bitmap for the panel background, the background color is not used.

> ***Note:*** *The View Options panel is also used to create a grid and to set a size for the window being built.*

Changing the Background Color

1. Click the Select button in the Background row.

The Color chooser panel appears.


Choose the color system

Choose the selection method

2. At the top of the panel, choose the color system and/or selection method. Then select the color you want. Use the RGB/HSV values and/or the color wheel to define your own colors or use the Color Names option to use predefined colors.

   The color selected appears in the lower-right rectangle of the Color chooser panel.

3. Click Apply in the Color chooser panel.

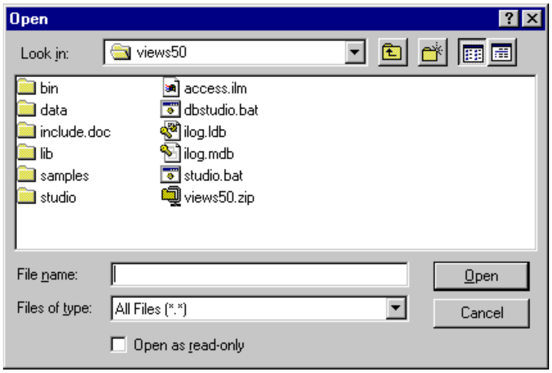
   The background of the window you are building in the work space changes to the color you selected.

4. Click Close in the Color chooser panel.

5. Click Close in the View Options panel.

Setting a Background Bitmap

1. Click the combo box button next to Background Bitmap in the Grid panel. The Open dialog box appears.



Look in: views50

File name:

Files of type: All Files (*.*)

Use the dialog box to choose a file containing a bitmap image. Click Open in the Open dialog box.

The background of the window you are building in the work space changes to the bitmap you selected.

2. Click Close in the View Options panel.

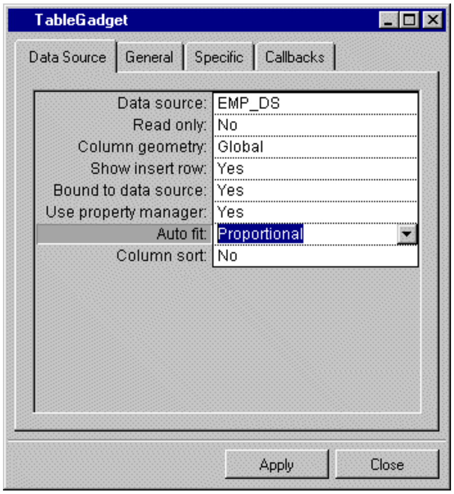
Setting Attachments

The attachments function allows you to specify how objects in the window change when the window is resized. To specify the attachments for the table gadget in the Employee window so that it will change size proportionally with the window, do the following:

1. Before actually setting the attachments, you need to specify that the columns within the table gadget can also be resized proportionally. In Selection editing mode ↖ , double-click the table gadget in the Main window.

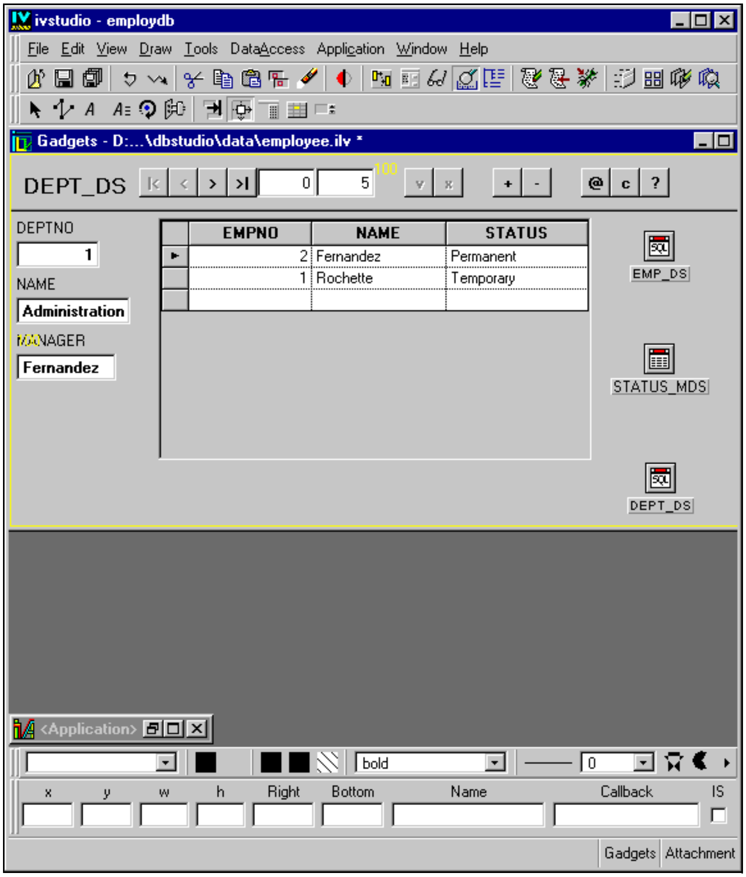
   The Table Gadget inspector appears.

2. When the table gadget is resized, its columns need to be resized proportionally with it. In the Table Gadget inspector, use the combo box menu to change the Auto Fit field to `Proportional`. Click Apply.



Data source:	EMP_DS
Read only:	No
Column geometry:	Global
Show insert row:	Yes
Bound to data source:	Yes
Use property manager:	Yes
Auto fit:	Proportional
Column sort:	No

3. Click the Attachments button in the Editing Modes toolbar of the Main window. Guides appear on the window borders and numbers on the top and left.


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
DEPT_DS
100
EMP_DS
STATUS_MDS
DEPT_DS
<Application>
bold
Gadgets
Attachment

DEPT_DS 0 5

DEPTNO: 1

NAME: Administration

MANAGER: Fernandez

EMPNO	NAME	STATUS
2	Fernandez	Permanent
1	Rochette	Temporary

x	y	w	h	Right	Bottom	Name	Callback	IS
								□

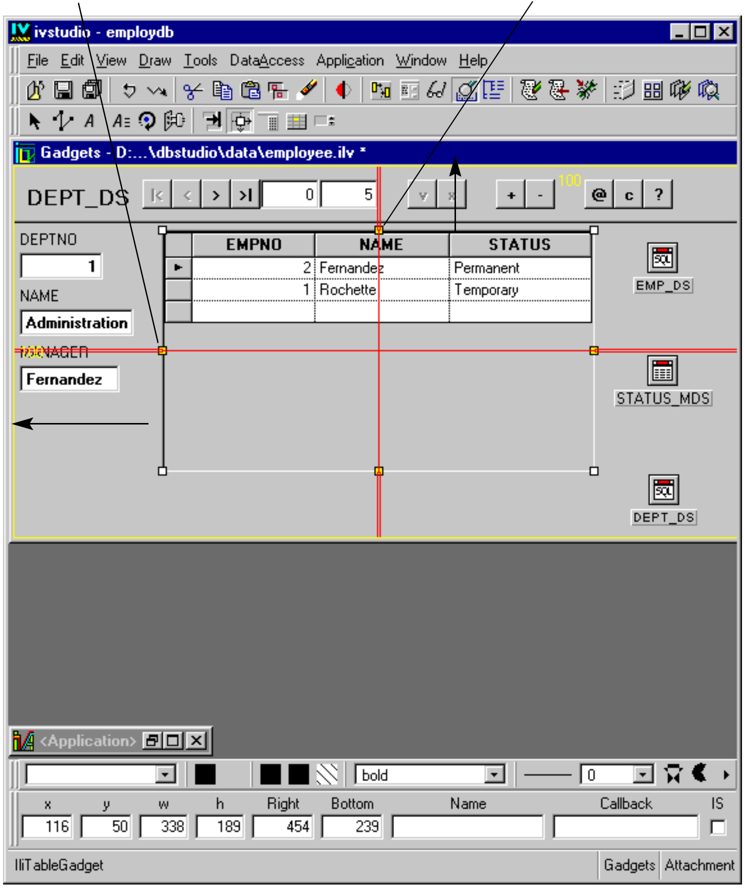
The numbers correspond to the weight of the section delimited by the guide. You can ignore these numbers for this example.

4. Set the attachments for the Employee table:

- Drag a line from the left middle handle to the guide on the left border. Release the mouse button when the guide becomes highlighted.
- Drag a line from the top middle handle to the guide on the top border. Release the mouse button when the guide becomes highlighted.

Drag from this handle to guide at left border

Drag from this handle to guide at top border


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
DEPT_DS
100
EMP_DS
STATUS_MDS
DEPT_DS
<Application>
bold
IliTableGadget
Gadgets
Attachment

DEPTNO: 1

NAME: Administration

MANAGER: Fernandez

	EMPNO	NAME	STATUS
►	2	Fernandez	Permanent
	1	Rochette	Temporary

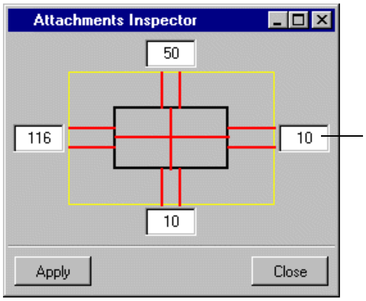
x	y	w	h	Right	Bottom	Name	Callback	IS
116	50	338	189	454	239			□

All the other attachments are made by default. This attachment (fixed, elastic, fixed) resizes the gadget proportionally while keeping it at the same distance from the borders.

5. Double-click the Employee table gadget.

The Attachments Inspector panel for the object appears when in Attachments editing mode.

6. To keep the table gadget 10 pixels from the right border, type `10` in the right box of the Attachments Inspector panel as shown in the following figure, then click Apply.



Type 10 here

Leave the other boxes with the numbers that appear in them. The table gadget will remain 10 pixels from the right border of the window and from the other borders according to the number of pixels appearing in their respective boxes.

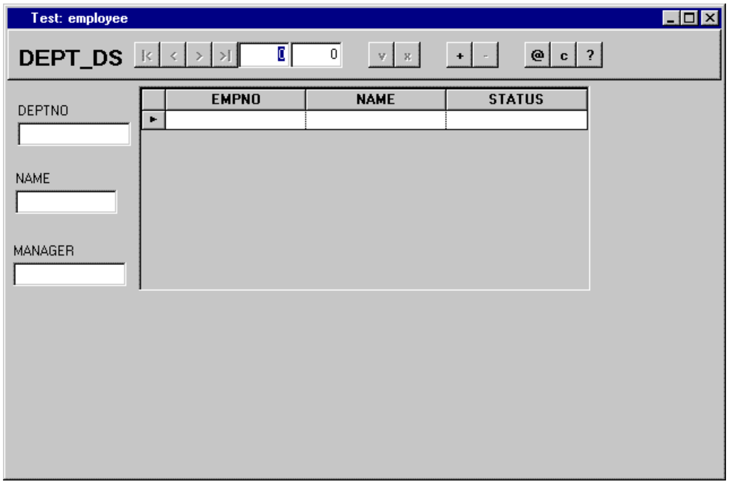
> ***Note:*** *In the Attachments Inspector, you can also click the attachment lines to change their type. They toggle between fixed and elastic when clicked.*

For more information about setting attachements, see the *IBM ILOG Views Controls User’s Manual*.

Testing Attachments

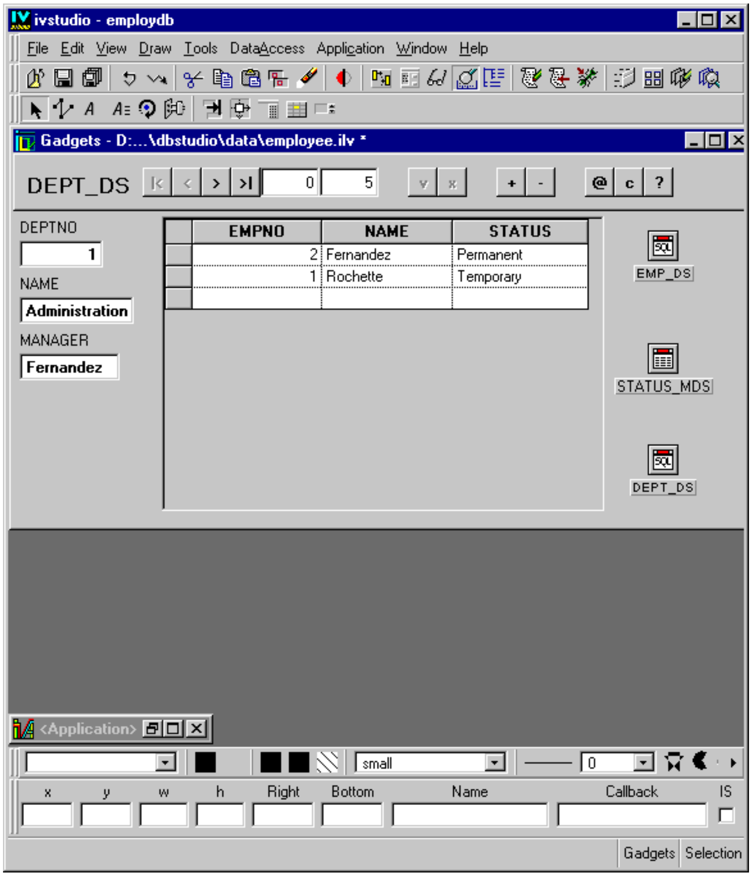
To test the attachments you applied to the objects, do the following:

1. Click the Test button in the toolbar of the Main window. The following Test window appears:



2. Change the window size by using the windowing system. You can see the behavior of the objects as the window changes size. If you need to make changes you can click the Test button again to close the Test window and make your adjustments in Attachments mode. You can go back and forth between Attachments and Test modes until you have what you want.

   You have now finished creating the Employee window. It should appear similar to the following:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - D:...\dbstudio\data\employee.ilv *
DEPT_DS
0
5
EMP_DS
STATUS_MDS
DEPT_DS
<Application>
small
0
x y w h Right Bottom Name Callback IS
Gadgets Selection

DEPTNO

1

NAME

Administration

MANAGER

Fernandez

EMPNO	NAME	STATUS
2	Fernandez	Permanent
1	Rochette	Temporary

3. Save the Employee window by clicking the Save button in the Main window.

Adding the Employee Window to the Application

To add the Employee window to the `employdb` application, do the following:

1. Make sure the Employee window (`employee.ilv`) is the current buffer in the work space of the Main window.

2. Click the Panel Class Palette button in the toolbar of the Main window.

   The Panel Class Palette is displayed in the bottom half of the Main window.

3. In the Panel Class Palette toolbar, click the New Panel Class button.

The employee panel appears in the palette as an icon with the name `Employee`.

4. Maximize the `employdb` application buffer window. (You minimized this window at the beginning of the tutorial.) If the `employdb.iva` file is no longer open, open the file by selecting Open from the File menu.

   Choose Edit Application from the Window menu. The current buffer changes to the `employdb` application buffer and the Panel Class Palette remains displayed.

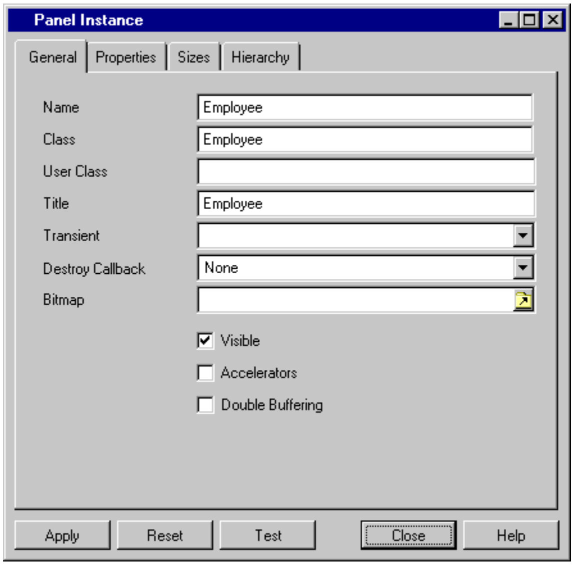
5. Drag the `Employee` icon from the Panel Class Palette to the application buffer work space.

   The `Employee` panel appears in the application buffer. This means that an instance of the employee panel has been successfully added to the application.

6. Double-click the `Employee` panel title bar or choose Panel Inspector from the Employee panel pop-up menu within the application buffer.

   The Panel Instance inspector appears for the `Employee` panel.

7. In the Panel Instance inspector, click the Visible toggle switch to deactivate it.



Name	Employee
Class	Employee
User Class	
Title	Employee
Transient	
Destroy Callback	None
Bitmap	

☑ Visible
☐ Accelerators
☐ Double Buffering

The panel will now be invisible when the application is launched. You want to do this because you are going to use a Control window to display this window.

8. Click Apply. Then click Close in the Panel Instance inspector.

9. Click the Save button in the Main window.

The application is saved with the new Employee window.

CHAPTER

5

Additional Features

In this final chapter of the tutorial, you will create three small panels, practicing what you have learned and getting acquainted with other features of Data Access.

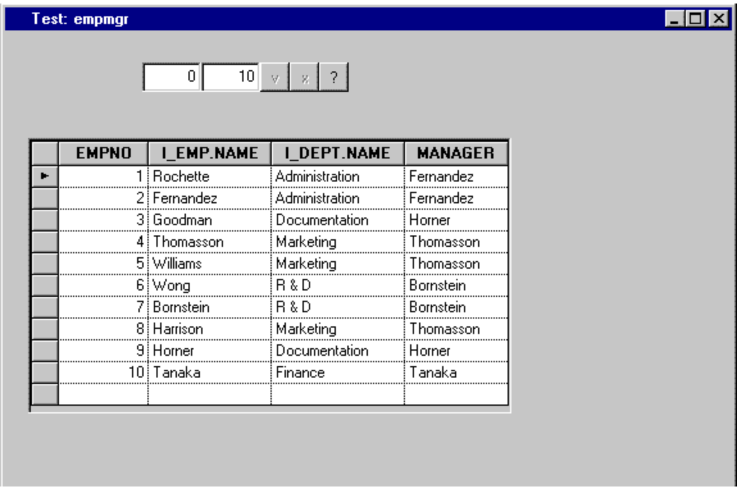
You will perform the following operations:

- *Creating the Employee Manager Window*
- *Creating the Department Maximum Salary Window*
- *Creating the Control Window*

Each window shows you different features of Data Access.

Creating the Employee Manager Window

You are now going to create a new buffer containing an Employee Manager window that will look like this when finished:



	EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
►	1	Rochette	Administration	Fernandez
	2	Fernandez	Administration	Fernandez
	3	Goodman	Documentation	Horner
	4	Thomasson	Marketing	Thomasson
	5	Williams	Marketing	Thomasson
	6	Wong	R & D	Bornstein
	7	Bornstein	R & D	Bornstein
	8	Harrison	Marketing	Thomasson
	9	Horner	Documentation	Horner
	10	Tanaka	Finance	Tanaka

Figure 5.1 *Employee Manager Window*

This window will have one data source, EMPMGR_DS. When you save the file, give it the name `empmgr.ilv`.

What You Are Going To Do

In this section, you will do the following tasks:

- *Defining the EMPMGR_DS SQL Data Source*
- *Connecting a Table Gadget to the EMPMGR_DS Data Source*
- *Joining Two Tables*
- *Finishing the Employee Manager Window*
- *Saving the Employee Manager Window*
- *Adding the Employee Manager Window to the Application*

Defining the EMPMGR_DS SQL Data Source

To define the EMPMGR_DS SQL data source, do the following:

1. Choose New > Gadgets from the File menu.

   A new gadgets buffer is opened ready for you to edit.

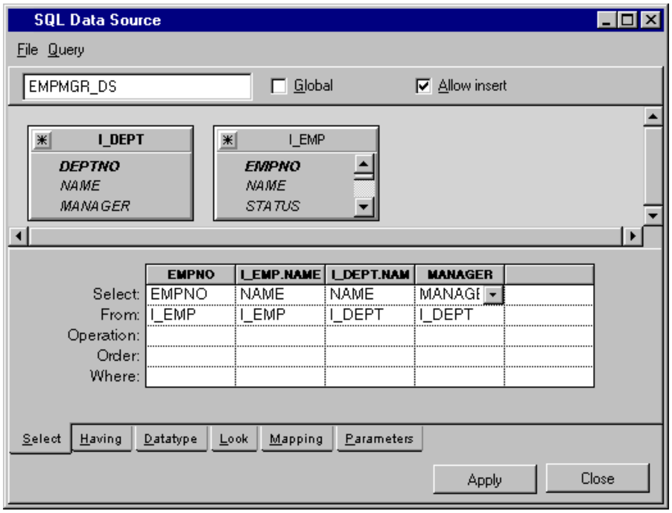
> ***Note:*** *If your* `employee.ilv` *buffer is still open, minimize it in the Main Window.*

2. Define the EMPMGR_DS data source like you did for the EMP_DS data source *Defining the EMP_DS SQL Data Source* on page 57.

- Type the name `EMPMGR_DS` in the Name field of the SQL Data Source inspector.
- In this case, you need to add both the I_EMP and I_DEPT tables to the data source. Select Add Tables... from the Query menu of the SQL Data Source inspector. When the Select Tables panel appears, select each table in the left list and move it to the right list by clicking the -> button. When both tables appear in the right list, click OK and then Close.

3. Click the Select page of the SQL Data Source inspector. From the I_EMP table, add the EMPNO and NAME columns (but not the DEPTNO, STATUS nor SALARY columns) to the EMPMGR_DS data source by dragging lines from the tables in the FROM section to the SELECT section.

4. From the I_DEPT table, drag lines to add the NAME and MANAGER columns (but not the DEPTNO column). The EMPMGR_DS SQL Data Source inspector now looks like this:



EMPMGR_DS ☐ Global ☑ Allow insert

I_DEPT: *DEPTNO*, *NAME*, *MANAGER*

I_EMP: *EMPNO*, *NAME*, *STATUS*

	EMPNO	I_EMP.NAME	I_DEPT.NAM	MANAGER	
Select:	EMPNO	NAME	NAME	MANAGI	
From:	I_EMP	I_EMP	I_DEPT	I_DEPT	
Operation:					
Order:					
Where:					

5. Choose Properties from the File menu of the SQL Data Source inspector. Ensure that the Updatable Table property is set to I_EMP.

> ***Note:*** *By default, only the first table added to a data source can be updated. In the FROM section of the SQL Data Source inspector, the title of the table that can be updated is shown in bold font. You can change which tables can be updated by double-clicking on each table title and changing the "Updatable" check box according to your needs.*

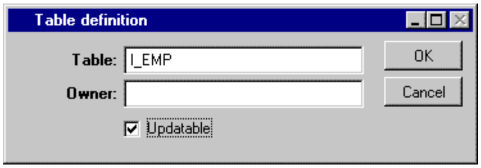


Table: I_EMP

Owner:

☑ Updatable

6. Click Apply in the SQL Data Source inspector.

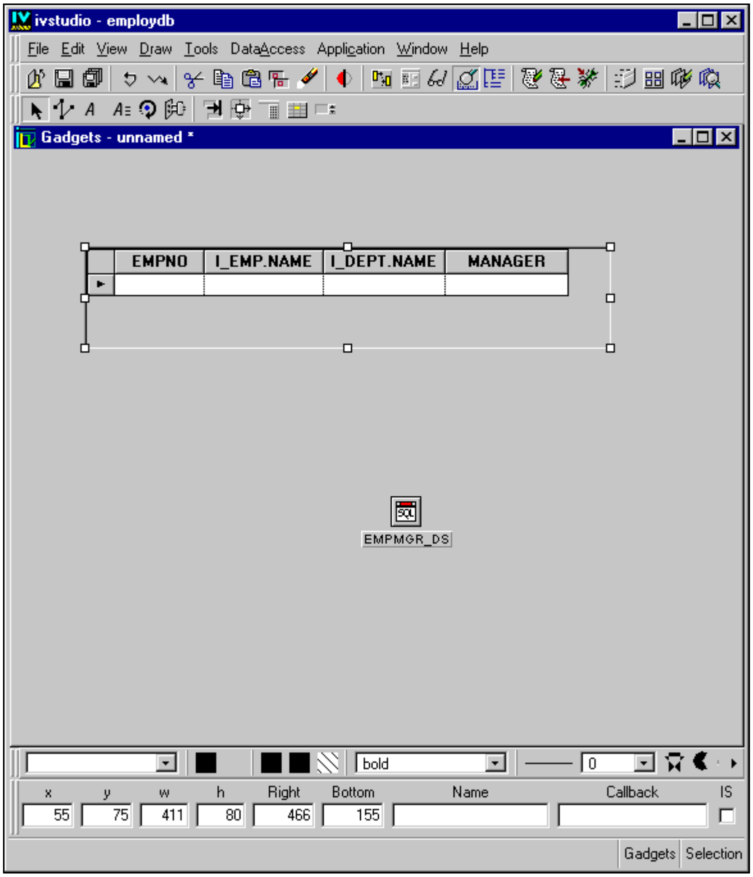
   `EMPMGR_DS` appears under the data source icon in the Main window.

Connecting a Table Gadget to the EMPMGR_DS Data Source

You are now going to connect a table gadget to the data source.

1. Drag a table gadget to the Main window and connect it to the EMPMGR_DS data source as you did in the procedure in *Connecting a Table Gadget to the EMP_DS Data Source* on page 64. Do not press the F9 key yet to add the data to your table.
2. Click Apply in the Table Gadget inspector panel.

   The Main window now looks like this:



bold 0

x	y	w	h	Right	Bottom	Name	Callback	IS
55	75	411	80	466	155			□

Joining Two Tables

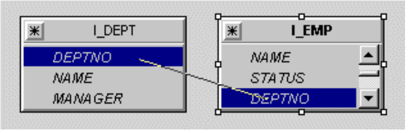
You are now going to do a *join* operation, which allows the display of data in one or more columns from multiple tables. A join operation connects two columns in different tables. Depending on the type of join operation used, you can specify the way in which rows are displayed from the two tables.

The following procedure will join the I_EMP and I_DEPT tables so that the DEPTNO cell of the I_EMP table is matched to the DEPTNO cell of the I_DEPT table. By default, an inner join is created. This means that for each employee who is in a department that is referenced in the I_DEPT table, you will display:

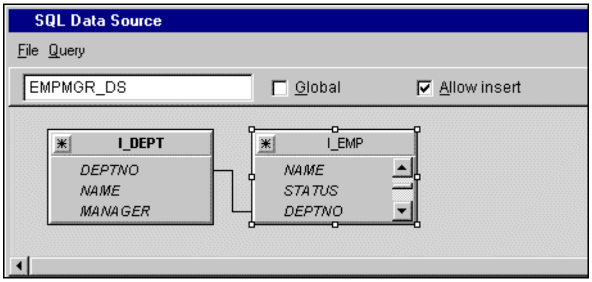
- From the I_EMP table: the employee number, name
- From the I_DEPT table: the employee's department name and manager

To join the DEPTNO columns of the I_EMP and I_DEPT tables, do the following:

1. In Selection mode ↖ , click the EMPMGR_DS SQL data source gadget to make the SQL Data Source inspector appear.
2. In the FROM section of the SQL Data Source inspector panel, drag a line from the DEPTNO label of the I_EMP table to the DEPTNO label of the I_DEPT table:



The FROM section now looks like this:



EMPMGR_DS

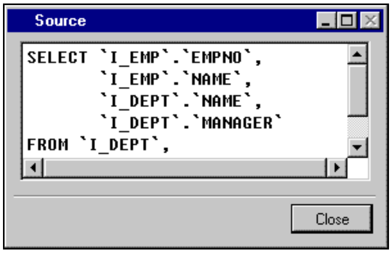
□ Global

☑ Allow insert

If you made a mistake, you can click the join line to select it, then choose Delete Join... from the Query menu of the EMP_DS SQL Data Source inspector. Alternatively you can use the Delete key to remove the selected join.

By default, an inner join is created. This is the type of join that we require. However, if you wish to check the join type, double-click the join line and the Edit Join panel is displayed.

3. Click Apply in the EMPMGR_DS SQL Data Source inspector panel to validate the join operation.
4. To view the SQL source file that reflects the operation you just performed, choose View Source... in the File menu of the EMPMGR_DS SQL Data Source inspector:



```
SELECT `I_EMP`.`EMPNO`,
       `I_EMP`.`NAME`,
       `I_DEPT`.`NAME`,
       `I_DEPT`.`MANAGER`
FROM `I_DEPT`,
```

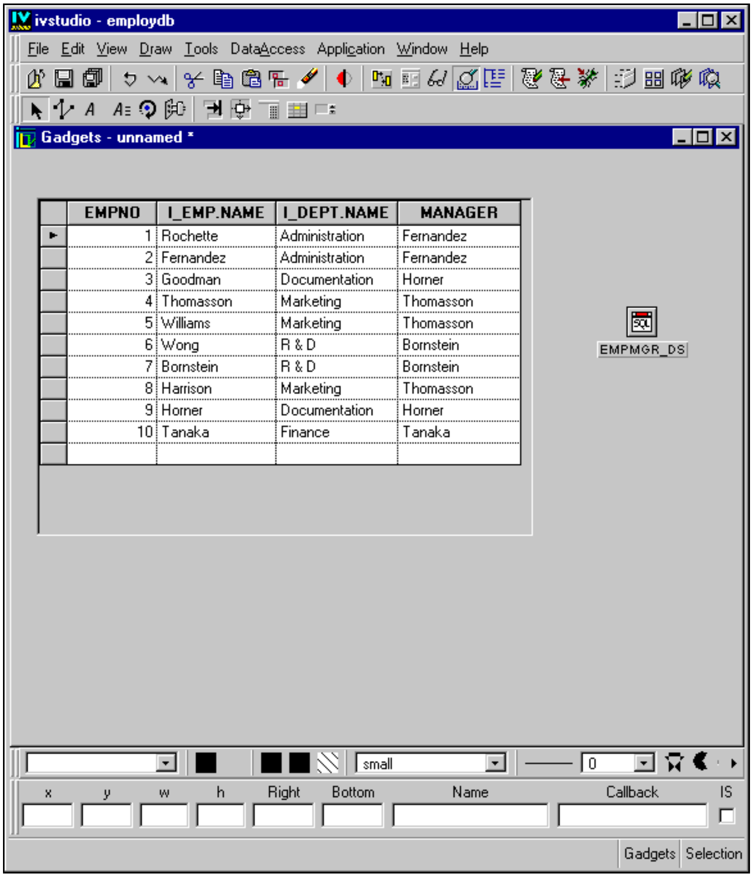
Click Close in the Source panel when finished with it.

5. Click the Active button in the Main window, select the I_EMP table gadget and press the F9 key.

   The table gadget is updated.

6. Adjust the size of the window. Resize the columns by placing the cursor on the line in the header row between the columns to be resized and dragging.

   The Employee Manager window now looks like this:


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - unnamed *
EMPMGR_DS
small
0
x y w h Right Bottom Name Callback IS
Gadgets Selection

EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
1	Rochette	Administration	Fernandez
2	Fernandez	Administration	Fernandez
3	Goodman	Documentation	Horner
4	Thomasson	Marketing	Thomasson
5	Williams	Marketing	Thomasson
6	Wong	R & D	Bornstein
7	Bornstein	R & D	Bornstein
8	Harrison	Marketing	Thomasson
9	Horner	Documentation	Horner
10	Tanaka	Finance	Tanaka

> ***Note:*** *Since a column called NAME appears in both tables, Data Access automatically adds the title of the table from which these columns came as a prefix to the column headers, thus distinguishing them from one another.*

Additional Information about Table Joins

A join operation between the columns of two different database tables enables you to specify a relationship between data that is common to the two tables. By specifying a join, you can set up a table with columns that originate from different database tables and indicate how the rows are to be displayed.

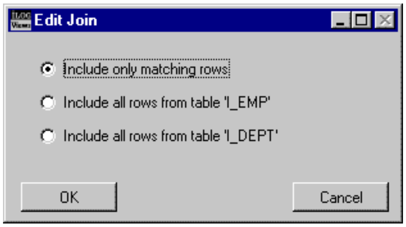
You can specify three different types of joins between the columns of database tables:

- **Inner Join:** Data is only displayed for rows that have the same value in the connected columns. This is the join that is created by default. In the Edit Join panel, it can be selected by clicking on the Include only matching rows radio button.
- **Outer Joins (Left or Right):** In addition to rows that have matching values in the joined columns, the rows of one of the tables that do not have a matching value in the column of the other table are displayed. These additional rows have empty cells in the columns where they do not have a value. A join like this is shown with a small black dot that indicates the table whose extra rows will be displayed. To select an outer join, select the appropriate radio button in the Edit Join panel.

To set a join, do the following:

1. In the FROM section of the SQL Data Source inspector, drag a line from the column to be joined in one of the tables to the column to be joined in the other table. A link appears between the joined columns in the two tables. By default, the join is an inner join.
2. To change the join type, double-click the join line.

   An Edit Join panel appears.



Edit Join

● Include only matching rows

○ Include all rows from table 'I_EMP'

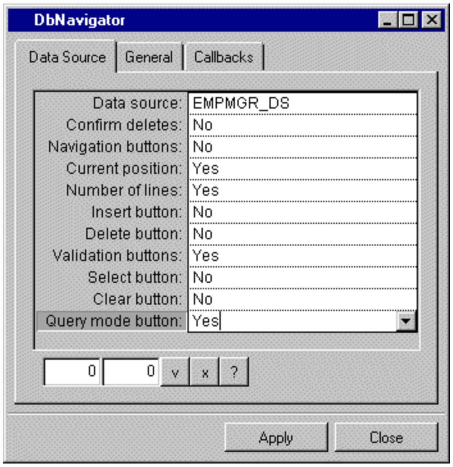
○ Include all rows from table 'I_DEPT'

   Use the radio buttons provided in this panel to select the type of join required. The join shown in the data source inspector changes accordingly.

Using Query Mode

You are now going to use query mode to display those employees whose employee number is less than 5 and whose manager is Thomasson, or whose employee number is greater than or equal 8 and whose manager starts with the letter T.

1. Click the Selection button in the Main window. Drag a DbNavigator gadget from the Data Access palette of the Palettes panel to the Gadgets buffer. Connect it to the EMPMGR_DS data source using the DbNavigator inspector.
2. You want your DbNavigator gadget to have the following buttons: Current Position, Number of Lines, Validate, Cancel, and Query Mode. Use the settings as shown in the following DbNavigator inspector to modify the DbNavigator gadget:



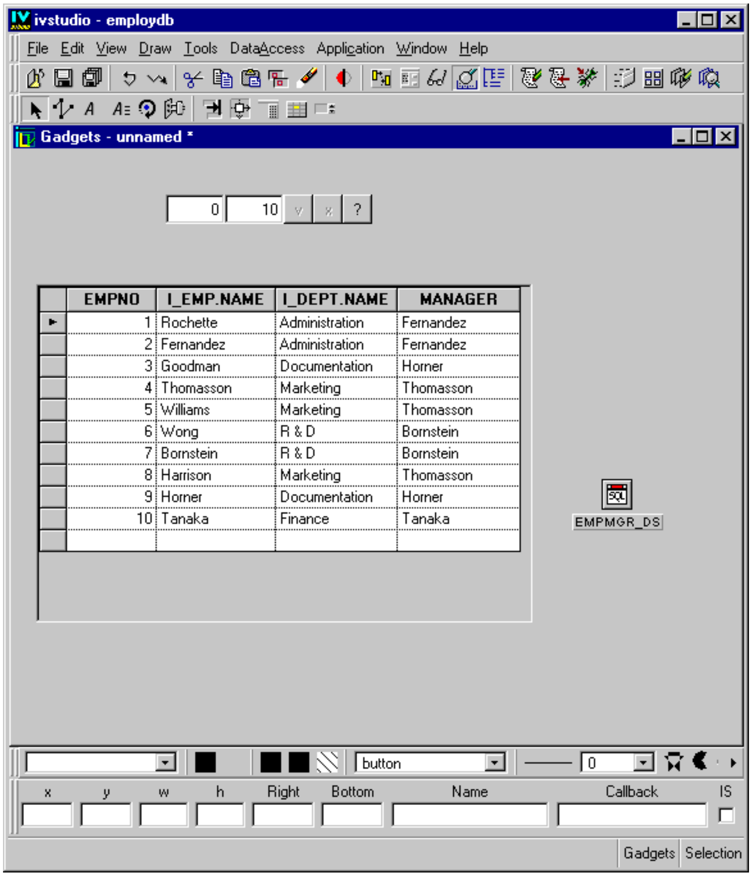
Data source:	EMPMGR_DS
Confirm deletes:	No
Navigation buttons:	No
Current position:	Yes
Number of lines:	Yes
Insert button:	No
Delete button:	No
Validation buttons:	Yes
Select button:	No
Clear button:	No
Query mode button:	Yes

0 0

3. Click Apply in the DbNavigator inspector.

4. Click the Active button in the Main window. Click the table gadget. Press the F9 key to refresh the data source. All data source data are displayed.

   The Main window now looks like this:



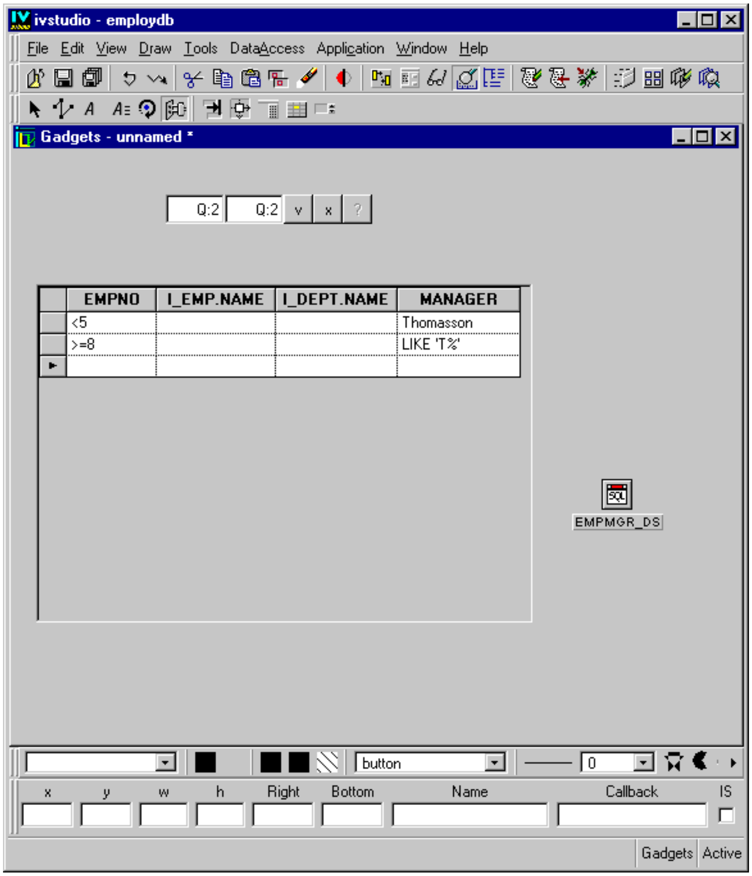
	EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
►	1	Rochette	Administration	Fernandez
	2	Fernandez	Administration	Fernandez
	3	Goodman	Documentation	Horner
	4	Thomasson	Marketing	Thomasson
	5	Williams	Marketing	Thomasson
	6	Wong	R & D	Bornstein
	7	Bornstein	R & D	Bornstein
	8	Harrison	Marketing	Thomasson
	9	Horner	Documentation	Horner
	10	Tanaka	Finance	Tanaka

5. While in Active mode, click the ? button of the DbNavigator gadget to switch to query mode.

The data source substitutes a memory table for the SQL table. This memory table has the same number of columns as the SQL table, but it differs in that all columns in the memory table have a String type.

6. In the table gadget in the Main window, do the following:

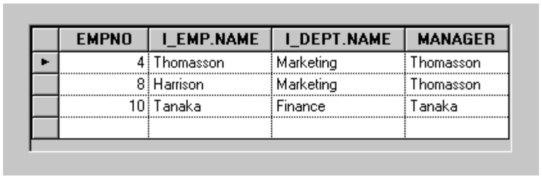
- On the first line, type `< 5` in the EMPNO column and `Thomasson` in the MANAGER column to list only those employees whose employee number is less than 5 and whose manager is Thomasson.
- On the second line, type `>= 8` in the EMPNO column and `LIKE 'T%'` in the MANAGER column to list only those employees whose employee number is greater than or equal to 8 and whose manager starts with the letter T.


ivstudio - employdb
File Edit View Draw Tools DataAccess Application Window Help
Gadgets - unnamed *
Q:2 Q:2 v x ?
EMPMGR_DS
button
0
x y w h Right Bottom Name Callback IS
Gadgets Active

	EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
	<5			Thomasson
	>=8			LIKE 'T%'
►				

7. Click the v button of the DbNavigator gadget to apply the query.

The table gadget displays only employees whose employee number is less than 5 and whose manager is Thomasson, or whose employee number is greater than or equal to 8 and whose manager starts with the letter T.



	EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
►	4	Thomasson	Marketing	Thomasson
	8	Harrison	Marketing	Thomasson
	10	Tanaka	Finance	Tanaka

Additional Information about Query Mode

Query mode lets you specify interactively a query to apply to the SQL table.

To use query mode, you must first add a DbNavigator gadget that displays the query mode button ? and validation buttons v x and is connected to the data source.

Switching to Query Mode

To enter query mode, click the query mode button ? of the DbNavigator gadget.

When query mode is entered, the data source substitutes a memory table for the SQL table. This memory table has the same number of columns as the SQL table, but differs in that all columns in the memory table have a String type.

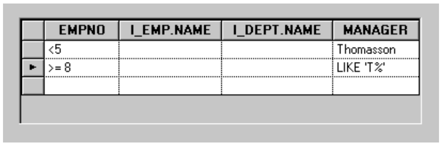
The user can then edit the contents of the memory table through the same set of gadgets that are used to edit the SQL table in regular (nonquery) mode.

Entering the Query

Each column of the memory table can contain:

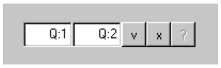
- A literal value (implying the = relational operator)
- A value containing the SQL wildcard character % or _ (implying the LIKE SQL operator)
- An SQL condition (such as NULL, NOT NULL, LIKE ‘ABC’, BETWEEN 1 AND 2, =, <>, and so on)

The memory table can contain more than one row. All conditions that appear on the same line will be combined with an AND operator when the query is applied. Conditions that appear on different lines will be combined with an OR operator.



	EMPNO	I_EMP.NAME	I_DEPT.NAME	MANAGER
	<5			Thomasson
►	>=8			LIKE 'T%'

If the DbNavigator displays the current position and/or the number of lines, while in query mode, these fields will display Q: in front of the value to show that query mode is active.



Q:1 Q:2

Applying and Canceling the Query

To apply the query, click the validate button v of the DbNavigator gadget. The portion of the WHERE clause based on the contents of the memory table will be synthesized. It will then be assigned to the SQL table query conjunct property and the data source will revert to using the SQL table instead of the memory table so that all connected gadgets show the contents of the SQL table.

Alternatively, the user can cancel the query by clicking the cancel button x of the DbNavigator gadget. This simply reverts to using the SQL table instead of the memory table so that all connected gadgets show the contents of the SQL table.

Finishing the Employee Manager Window

For more practice, you can do the following:

1. Apply resources to the table gadget as described in the section *Using Resources* on page 99.
2. Resize the window in relation to the table gadget as described in the section *Setting the Size of the Window* on page 102.
3. Attach the table gadget to the window as described in the section *Setting Attachments* on page 106.
4. Change the background of the window as described in the section *Setting the Background of the Window* on page 104.

Saving the Employee Manager Window

If you have not already saved the window, do so now by choosing Save As... from the File menu in the Main window. Give it the name `empmgr.ilv` and save it in the directory in which you saved the `.iva` application file.

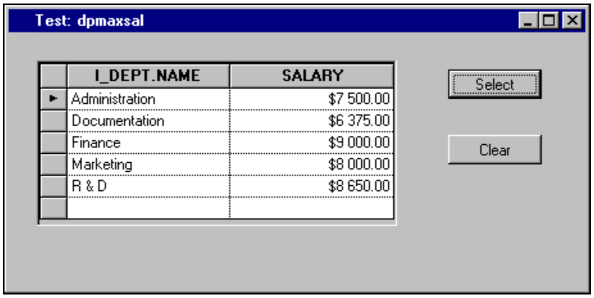
For information on saving windows, see the section *Saving and Opening Windows* on page 53.

Adding the Employee Manager Window to the Application

Add the Employee Manager window to the `employdb` application by following the procedure described in the section *Adding the Employee Window to the Application* on page 112.

Creating the Department Maximum Salary Window

You are now going to create the Department Maximum Salary window that will show the maximum salary in each department. When finished, the window will look like this:



Test: dpmaxsal

	I_DEPT.NAME	SALARY
►	Administration	$7 500.00
	Documentation	$6 375.00
	Finance	$9 000.00
	Marketing	$8 000.00
	R & D	$8 650.00

Figure 5.2 *Department Maximum Salary Window*

This window will have one data source, DEPTMAXSAL_DS. When you save it, give it the name `dpmaxsal.ilv`.

What You Are Going To Do

In this section, you will do the following tasks:

- *Defining the DEPTMAXSAL_DS SQL Data Source*
- *Specifying Selection Criteria Operations*
- *Connecting a Table Gadget to the DEPTMAXSAL_DS Data Source*
- *Saving the Department Maximum Salary Window*
- *Using Predefined Callbacks*
- *Finishing the Department Maximum Salary Window*
- *Saving the Department Maximum Salary Window*
- *Adding the Department Maximum Salary Window to the Application*

Defining the DEPTMAXSAL_DS SQL Data Source

Begin to define the DEPTMAXSAL_DS SQL data source by doing the following:

1. Choose New > Gadgets from the File menu in the Main window.

A new Gadgets buffer is opened ready for editing.

> ***Note:*** *If your* `empmgr.ilv` *buffer is still open, minimize it in the Main Window.*

2. In Selection mode , drag an SQL Data Source gadget from the Data Access palette to the Main window.

3. Double-click the SQL Data Source gadget.

   The SQL Data Source inspector appears.

4. In the SQL Data Source inspector, type `DEPTMAXSAL_DS` in the top-left text field.

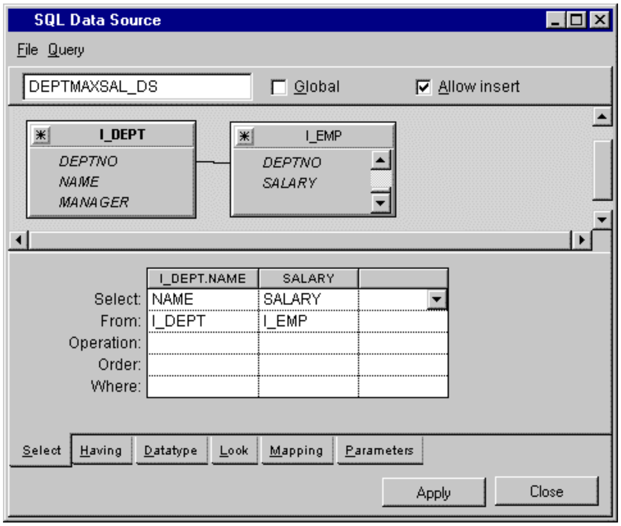
5. In the SQL Data Source inspector, choose Add Table... from the Query menu to add the I_EMP and I_DEPT tables to the DEPTMAXSAL_DS data source. This operation is described in the section *Defining the EMP_DS SQL Data Source* on page 57.

6. Do a join operation by dragging a line between the DEPTNO columns of the I_EMP and I_DEPT tables as you did in the section *Joining Two Tables* on page 119.

7. Add the following columns to the DEPTMAXSAL_DS data source table by dragging lines to the SELECT section from the:

   - NAME column in the I_DEPT table
   - SALARY column in the I_EMP table

   The DEPTMAXSAL_DS SQL Data Source inspector panel now looks like this:



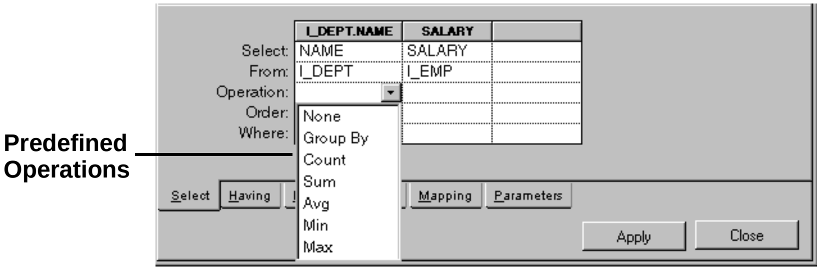
DEPTMAXSAL_DS □ Global ☑ Allow insert

	I_DEPT.NAME	SALARY	
Select:	NAME	SALARY	
From:	I_DEPT	I_EMP	
Operation:			
Order:			
Where:			

> ***Note:*** *The header of the NAME column taken from the I_DEPT table changes from NAME to I_DEPT.NAME to distinguish it from the NAME column in the I_EMP table.*

Specifying Selection Criteria Operations

The Operation row of the SELECT section in the SQL Data Source inspector contains a number of operations that you can choose to establish selection criteria. The combo box menu shows this list of operations.



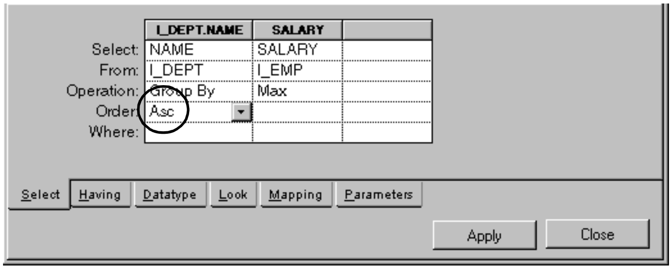
	I_DEPT.NAME	SALARY	
Select:	NAME	SALARY	
From:	I_DEPT	I_EMP	
Operation:			
Order:			
Where:			

Predefined Operations

You are going to continue to define the data source by using a Group By operation to display the highest salary in each department. Do the following:

1. In the Select page of the SQL Data Source inspector, choose the following from the combo boxes in the table:
   - `Group By` in the Operation row of the I_DEPT.NAME column.
   - `Max` in the Operation row of the SALARY column.
2. To make the departments appear in alphabetical order, choose `Asc` from the combo box menu in the Order row of the I_DEPT.NAME column.

   The Select page of the SQL Data Source inspector panel now looks like this:



	I_DEPT.NAME	SALARY	
Select:	NAME	SALARY	
From:	I_DEPT	I_EMP	
Operation:	Group By	Max	
Order:	Asc		
Where:			

3. Click Apply.

   You have now finished defining the DEPTMAXSAL_DS SQL data source.

Connecting a Table Gadget to the DEPTMAXSAL_DS Data Source

You are now going to connect a table gadget to the data source.

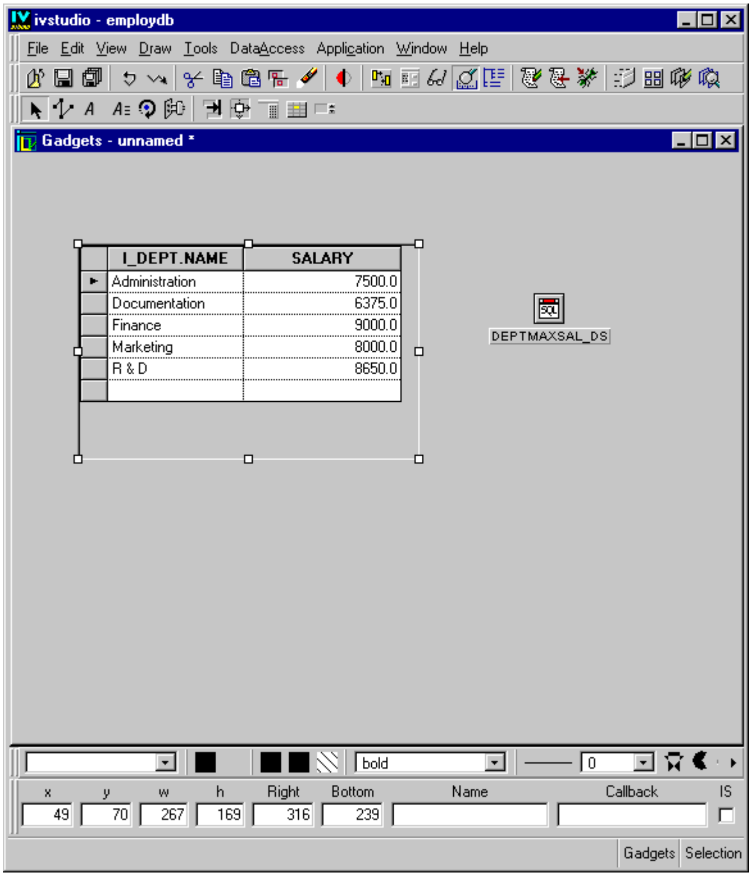
1. Drag a table gadget to the work space and connect it to the DEPTMAXSAL_DS data source as you did for the EMP_DS data source in the procedure described *Connecting a Table Gadget to the EMP_DS Data Source* on page 64.

   The DEPTMAXSAL_DS SQL data source is now defined and connected to a table gadget.

2. Click the Active mode button in the Main window. Click in the table gadget and press the F9 key to refresh the data source.

   The I_EMP table gadget in the Gadgets buffer changes to display the name of each department along with the highest salary in that department.

   The Main window now looks like this:



	I_DEPT.NAME	SALARY
►	Administration	7500.0
	Documentation	6375.0
	Finance	9000.0
	Marketing	8000.0
	R & D	8650.0

x	y	w	h	Right	Bottom	Name	Callback	IS
49	70	267	169	316	239			□

Keep the window as it is, since you will need it to continue the tutorial.

To change the format of the salary column in the table gadget (see the figure above) so that it shows the type of currency and has two digits after the decimal point, do the following:

1. Choose Application Properties from the Data Access menu.

The Application Properties panel appears.

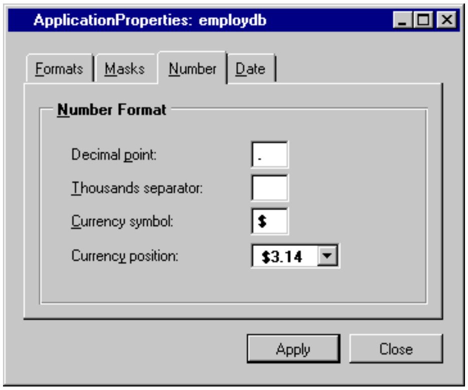
2. Click the Number page and set the format like this:

Decimal point: `.`

Thousands separator: no separator

Currency symbol: `$`

Currency position: `$3.14`



Decimal point: .

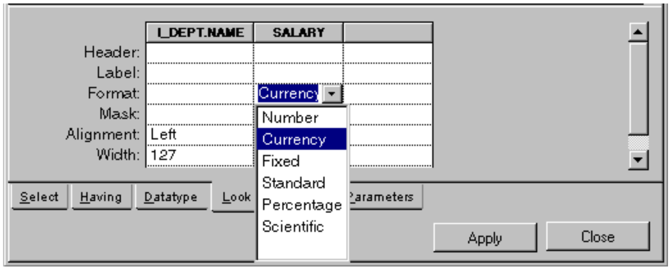
Thousands separator:

Currency symbol: $

Currency position: $3.14

3. Click Apply, then Close.
4. In Selection mode , double-click the DEPTMAXSAL_DS SQL Data Source gadget in the Gadgets buffer.

   The SQL Data Source inspector appears.
5. On the Look page, choose `Currency` in the combo box menu of the Format cell of the Salary column.

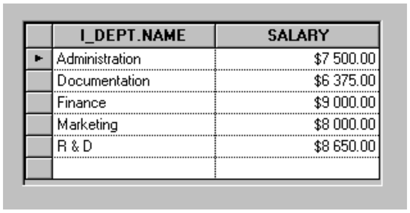


	I_DEPT.NAME	SALARY	
Header:			
Label:			
Format:		Currency	
Mask:			
Alignment:	Left		
Width:	127		

The word `Currency`, which is a predefined system format, appears in the cell.

6. Click Apply in the SQL Data Source inspector.

   In the Gadgets buffer, the format of the Salary column of the table gadget changes:

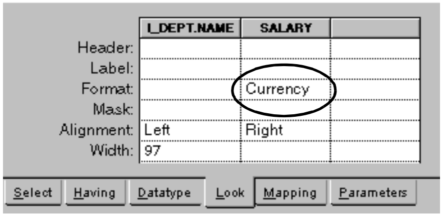


	I_DEPT.NAME	SALARY
►	Administration	$7 500.00
	Documentation	$6 375.00
	Finance	$9 000.00
	Marketing	$8 000.00
	R & D	$8 650.00

Keep the window as it is as you will need it to continue the tutorial.

Additional Information about Using Data Formats

You can create and edit data formats for your data source tables in various ways. Whatever method used, you always apply the format in a Format cell of a data source column in the SELECT section of the SQL Data Source inspector Look page.



	I_DEPT.NAME	SALARY	
Header:			
Label:			
Format:		Currency	
Mask:			
Alignment:	Left	Right	
Width:	97		

You can either choose a predefined named format from the cell combo box menu or you can define your own string, number or date format using the symbols and formulas found in Appendix B, *Format Syntax*. For example, instead of choosing the predefined system format `Currency` from the menu as in the figure, you could create your own format such as this: `$ #,##0.00`. See Appendix B, *Format Syntax*.

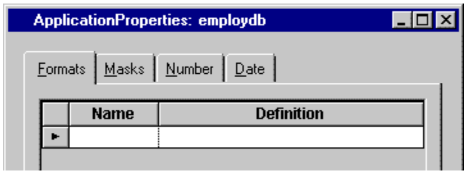
Predefined System Formats

Data Access provides numerous predefined formats that you can choose from the combo box menu of the Format cell. Whether system formats appear in the menu depends on the data type for the column; that is, what is entered in the Type cell on the Datatype page of the SQL Data Source inspector.

Predefined User Formats

You can create your own predefined format by using the symbols and formulas described in Appendix B and then giving the format a name. This name then appears in the combo box menu of all Format cells whose data type is appropriate for the format. To create a user predefined format, do this:

1. Choose Application Properties from the Data Access menu of the Application Editor panel. The Application Properties panel appears.



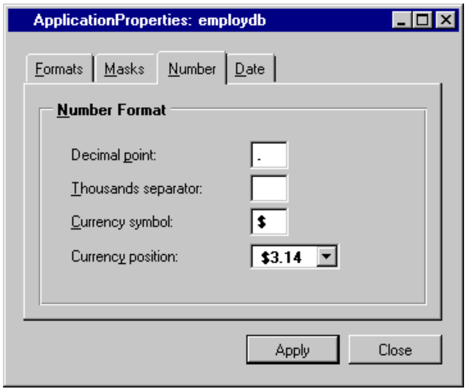
2. Click the Format page. On this page, type a name for the format in the Name field and define a string, number, or date format in the Definition field (see Appendix B, *Format Syntax*).

3. Click Apply. The format is defined and the name appears in the appropriate combo box menus of the Format cells.

Application-Wide Settings

You can apply application-wide setting as follows:

Numbers: Use the Number page in the Application inspector panel to define settings for the decimal point and thousands separator as well as a currency symbol and its placement. These settings are always applied when they are appropriate for the value. The settings also apply to number formulas typed in the Format cell and with predefined user number formats.



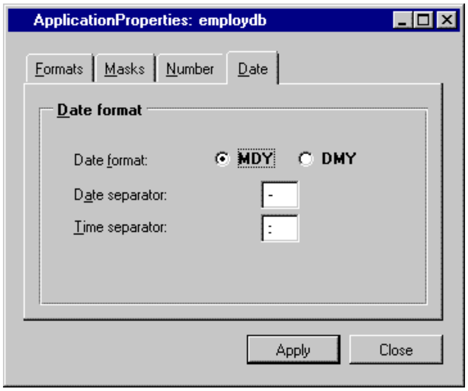
Decimal point: .

Thousands separator:

Currency symbol: $

Currency position: $3.14

Dates: Use the Date page in the Application inspector panel to define the order of the month and date and the date and time separator symbols. These settings are applied with predefined date and time system formats that appear in the Format cell combo box menu when `Date` is used in the Datatype cell. The settings also apply to date formulas typed in the Format cell and with predefined user date formats.



Date format

Date format: ● MDY ○ DMY

Date separator: -

Time separator: :

Saving the Department Maximum Salary Window

If you have not already saved the window, do so now by choosing Save As... from the File menu in the Main window. Give it the name `dpmaxsal.ilv` and save it in the directory in which you saved the `.iva` application file *Creating the Application File* on page 50.

Using Predefined Callbacks

Various predefined callbacks are provided with Data Access as well as the means for defining new callbacks. You can define new callbacks using:

- The C++ language (see the *IBM ILOG Views Data Access Reference Manual* and *IBM ILOG Views Controls User's Manual*)

This section describes the predefined callbacks that can be used without having to write any code. These callbacks can be attached to any gadget or menu item, whether it is taken from the Data Access palettes or from other IBM ILOG Views palettes.

The following predefined callbacks are currently available:

- `@Quit()`
- `@ShowPanel(panelName)`
- `@HidePanel(panelName)`
- `@Validate(dataSourceName)`
- `@Cancel(dataSourceName)`
- `@Select(dataSourceName)`
- `@Clear(dataSourceName)`
- `@StartInsert(dataSourceName)`
- `@Commit(sessionName)`

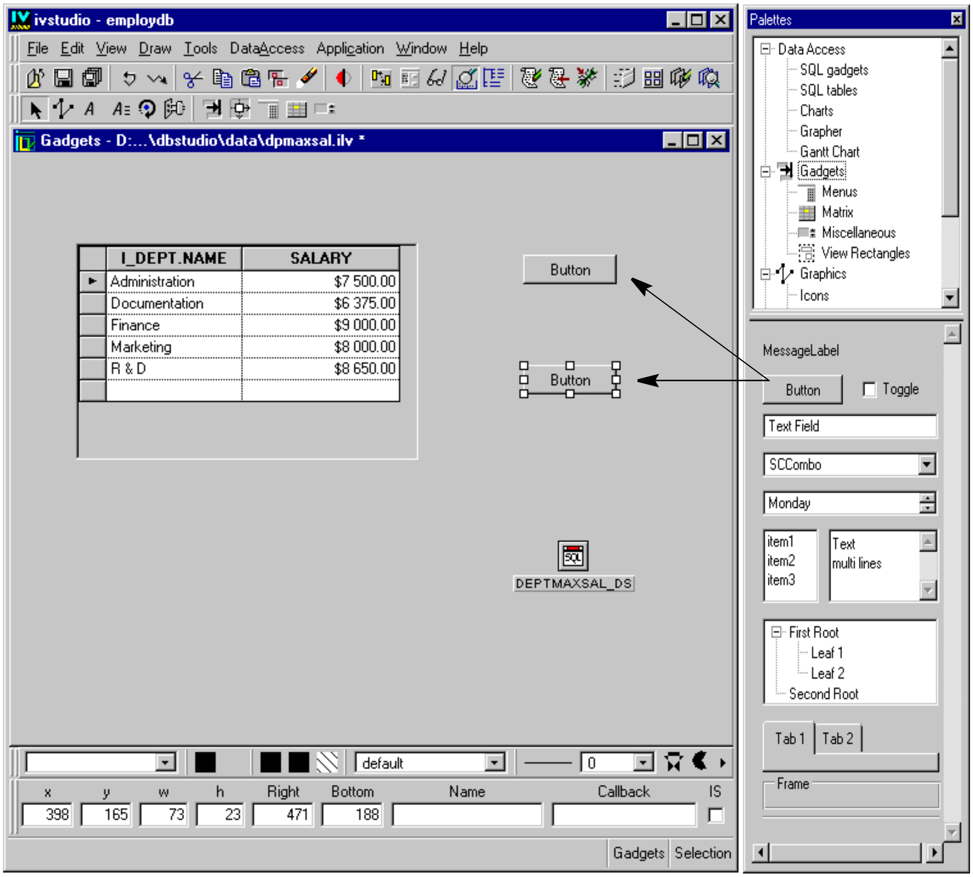
- `@Rollback(sessionName)`
- `@Connect(sessionName)`
- `@QueryConnect(sessionName)`
- `@Disconnect(sessionName)`

You are now going to create two button gadgets that make use of the `@Select` and `@Clear` callbacks. The first is used for updating the data source table and the second for clearing it. To do this, follow these steps:

1. In the Palettes panel, choose Gadgets from the tree structure in the upper pane of the window.

   The lower pane of the window changes, making the set of IBM ILOG Views gadgets available.

2. In Selection mode , drag two `IlvButton` gadgets from the Gadgets palette to the Main window and place them one above the other as in the figure that follows.



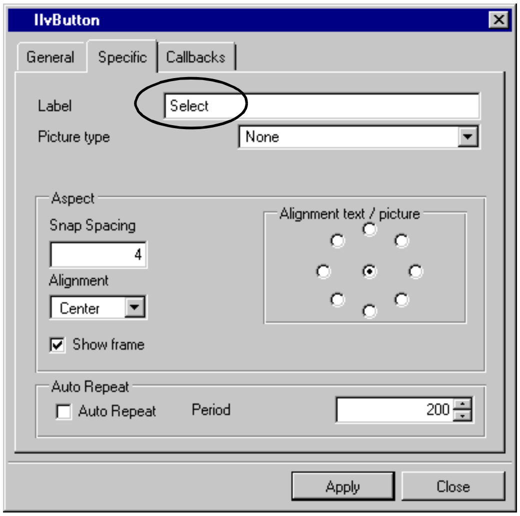
	I_DEPT.NAME	SALARY
►	Administration	$7 500.00
	Documentation	$6 375.00
	Finance	$9 000.00
	Marketing	$8 000.00
	R & D	$8 650.00

x	y	w	h	Right	Bottom	Name	Callback	IS
398	165	73	23	471	188			☐

3. Double-click the top button gadget.

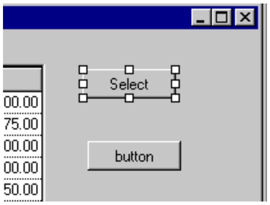
The IlvButton inspector appears.

4. In the IlvButton inspector, click the Specific tab to make that notebook page appear. Change the Label field to `Select`. Click Apply.

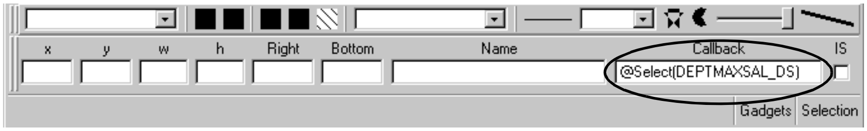


Label: Select
Picture type: None
Snap Spacing: 4
Alignment: Center
☑ Show frame
□ Auto Repeat
Period: 200

The top button gadget label changes to `Select` in the Main window.



5. With the top button gadget selected, enter `@Select(DEPTMAXSAL_DS)` in the Callback field of the Generic inspector area in the Main window. If the IS button of the Generic inspector is checked, uncheck it.



Callback: @Select(DEPTMAXSAL_DS)
IS: □

> ***Note:*** *If a gadget is not already connected to the data source, you must enter the data source name as a parameter in the Callback field (in this case,* `DEPTMAXSAL_DS`*). If you used IBM ILOG Views Data Access gadgets that you already connected to the data source (in the Data Source cell of the gadget inspector panel), you would just enter the Callback name without the data source parameter, for example:* `@Select`.

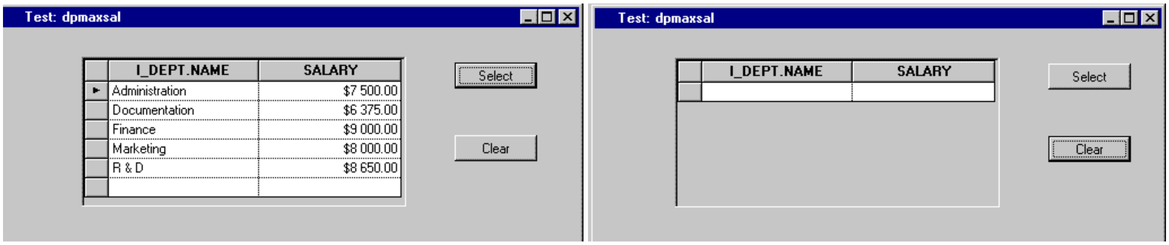
6. Repeat steps 2-4 for the second button gadget, but change the Label field to `Clear` and enter `@Clear(DEPTMAXSAL_DS)` in the Callback field of the Generic inspector.

7. To test the callback, click the Test button in the toolbar of the Main window.

The Test window appears.

> ***Note:*** *You cannot test predefined callbacks in Active editing mode.*

8. Click the Select and Clear buttons successively to see how the callbacks attached to the button gadgets update and clear the table gadget connected to the DEPTMAXSAL_DS data source.



I_DEPT.NAME	SALARY
Administration	$7 500.00
Documentation	$6 375.00
Finance	$9 000.00
Marketing	$8 000.00
R & D	$8 650.00

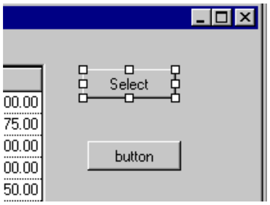
9. Click the Test button again to close the Test window.

> ***Note:*** *Some predefined callbacks, such as* `@ShowPanel` *and* `@HidePanel`*, do not work when tested with the Test window before being integrated with the application. First, you have to integrate them into the application and test them when the current buffer in the Main window is the Application buffer.*

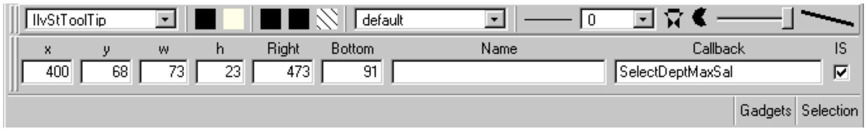
Additional Information About Using IBM ILOG Script Callbacks

Instead of using predefined callbacks such as "@Select", you can code an IBM ILOG Script callback. By doing this, you have access to the extensive Data Access scripting API as well as to IBM ILOG Script intrinsic language features (such as loops, conditions, and so on). To define an IBM ILOG Script callback, do the following:

1. Select the top button gadget in the buffer window:



2. Enter `SelectDeptMaxSal` in the Callback field of the Generic inspector area in the Main Window.

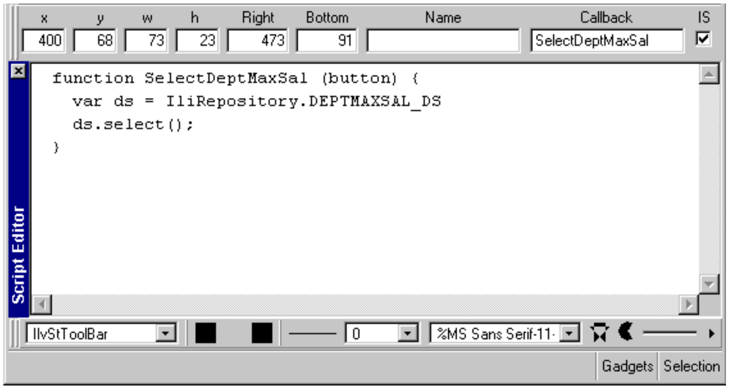


x	y	w	h	Right	Bottom	Name	Callback	IS
400	68	73	23	473	91		SelectDeptMaxSal	☑

3. Make sure that the IS toggle is checked.

4. Click the Script Editor button in the toolbar. The script editor appears in the bottom part of IBM ILOG Views Studio main panel.

5. Type the following IBM ILOG Script code in this editor:



x	y	w	h	Right	Bottom	Name	Callback	IS
400	68	73	23	473	91		SelectDeptMaxSal	☑

```
function SelectDeptMaxSal (button) {
  var ds = IliRepository.DEPTMAXSAL_DS
  ds.select();
}
```

The IBM ILOG Script callback is now defined. The code in the script editor will be written in the `.ilv` file the next time you save the buffer.

As an alternative to storing IBM ILOG Script code in the `.ilv` file of the panel, you can edit, with any text editor of your choice, an IBM ILOG Script source file having the same name as the `.ilv` file but with the `.js` extension instead. If present in the same directory, this file will be read when you start test mode or when you run the application with the “runner” bin.

We recommend that you test the application instead of testing the individual panels when you use IBM ILOG Script callbacks. To test the application, ensure that the current buffer is the <Application> buffer before you click the Test button in the toolbar.

Finishing the Department Maximum Salary Window

For more practice, you can do the following:

1. Apply resources to the gadgets as described in the section *Using Resources* on page 99.
2. Align the gadgets with each other as described in the section *Aligning Objects* on page 97.
3. Resize the window in relation to the gadgets as described in the section *Setting the Size of the Window* on page 102.
4. Attach the gadgets to the window as described in the section *Setting Attachments* on page 106.
5. Change the background of the window as described in the section *Setting the Background of the Window* on page 104.

Saving the Department Maximum Salary Window

Save the Department Maximum Salary window by clicking the Save button 💾 in the Main Window.

Adding the Department Maximum Salary Window to the Application

Add the Department Maximum Salary window to the `employdb` application by following the procedure described in the section *Adding the Employee Window to the Application* on page 112.

Creating the Control Window

The last window you are going to create is the Control window.

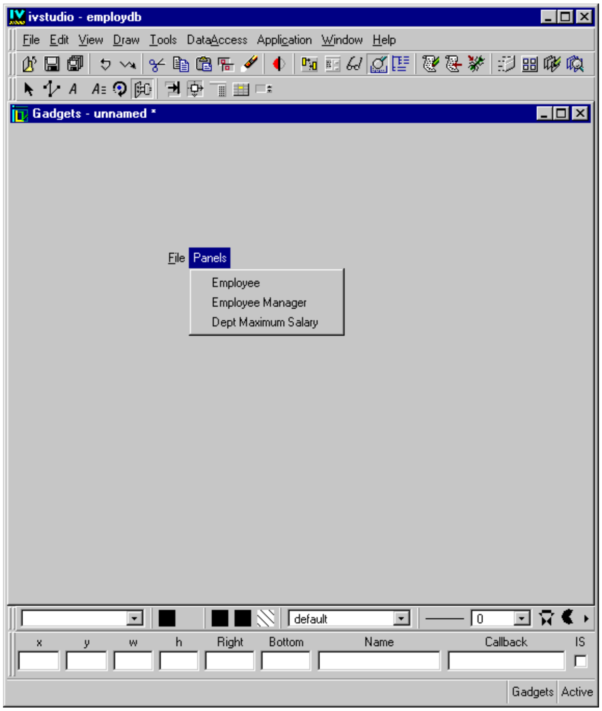


Figure 5.3 Control Window

This window will consist only of menus. When you save it, give it the name `control.ilv`.

What You Are Going To Do

In this section, you will perform the following procedures to create the Control window:

- *Adding a Menu Bar to the Window*
- *Creating the File Pop-up Menu*
- *Attaching the File Pop-up Menu to the File Menu Item*
- *Creating and Attaching the Panels Pop-up Menu*
- *Finishing the Control Window*
- *Using the Browser*

Adding a Menu Bar to the Window

To add a menu bar to the Control window, do the following:

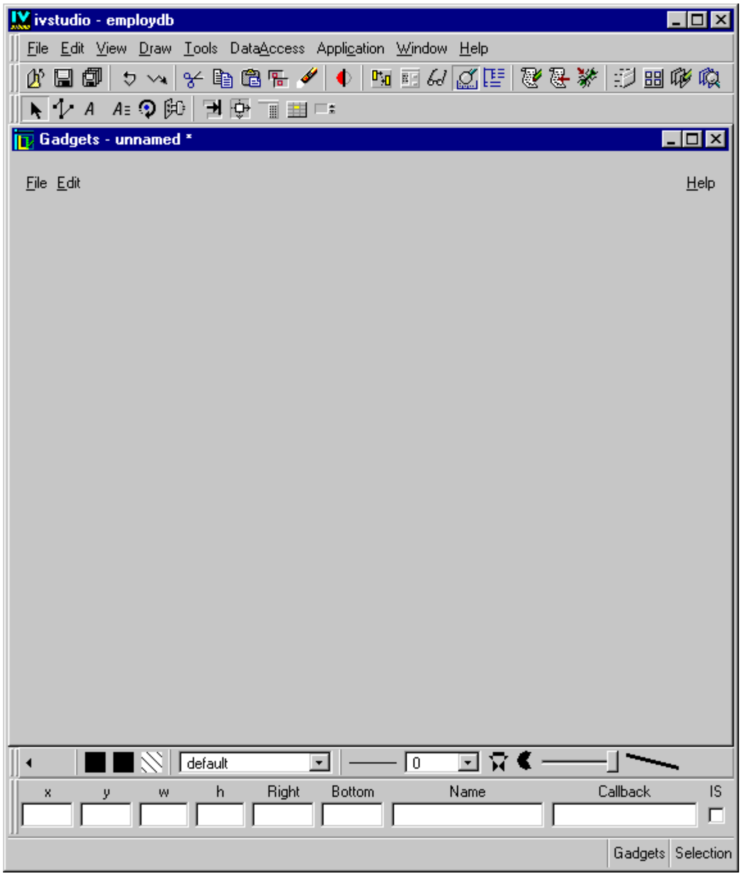
1. Choose New > Gadgets from the File menu in the Main window.

   An empty Gadgets buffer window is opened ready for editing.

> ***Note:*** *If your* `dpmaxsal.ilv` *buffer is still open, minimize it in the Main Window.*

2. In the upper pane of the Palettes panel, click Menus under Gadgets.
3. Drag the menu bar gadget from the Gadgets palette and drop it in the Gadgets buffer window in the Main window, leaving it selected.

   The menu bar automatically positions itself at the top while taking on the width of the window with default attachments.
4. Use the mouse to resize the panel so that its size corresponds more or less to that of the following figure:

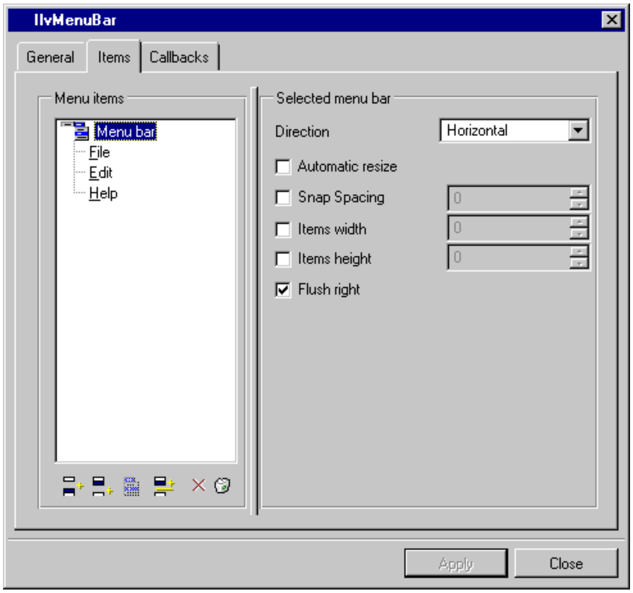


5. Double-click the menu bar gadget you just placed in the work space.

   The IlvMenuBar inspector appears.

> ***Note:*** *Until you click Apply in the inspector panel, your actions do not affect the menu bar in the work space.*

6. Go to the Items notebook page. Click the Flush right toggle switch to deactivate it.



Menu items

- Menu bar
  - File
  - Edit
  - Help

Selected menu bar

Direction: Horizontal

□ Automatic resize

□ Snap Spacing 0

□ Items width 0

□ Items height 0

☑ Flush right

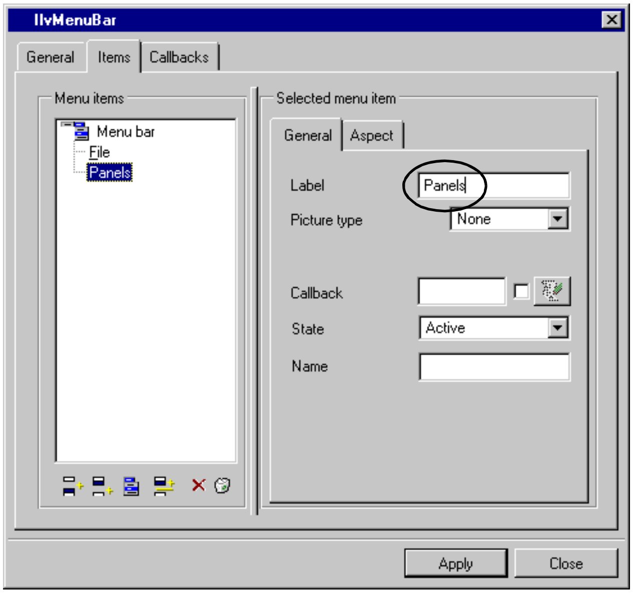
The Help menu item moves to the left. When activated, the Flush right toggle switch places the right-most item on the menu bar at the right margin, which you do not need in this example.

7. In the Menu Items tree gadget on the left side of the page, click Help to select it. Then click Remove button ✕ under the tree gadget.

The Help item is removed from the menu bar.

> ***Note:*** *When you want to insert a new item, select the item in the menu bar of the inspector panel before which you want the new item to be inserted and then click Insert. If no item is selected, the item is added to the bottom of the list of items in the tree gadget in the Menu Items box.*

8. In the Menu Items tree gadget, click Edit to select it. You are going to change the label of this menu item from Edit to Panels. In the Selected menu item box on the right side of the Items page, type `Panels` in the Label field. Notice that the label in the tree gadget on the left changes as you type.



Label: Panels

Picture type: None

State: Active

Leave the File menu item as it is.

> ***Note:*** *If you use an ampersand (*`&file`*,* `&menu_file`*), you make the item a variable for multilingual applications. By using the Message Editor, you can localize your window into different languages. You access this panel by choosing Messages Editor from the Tools menu. You will not be using this feature in the example.*

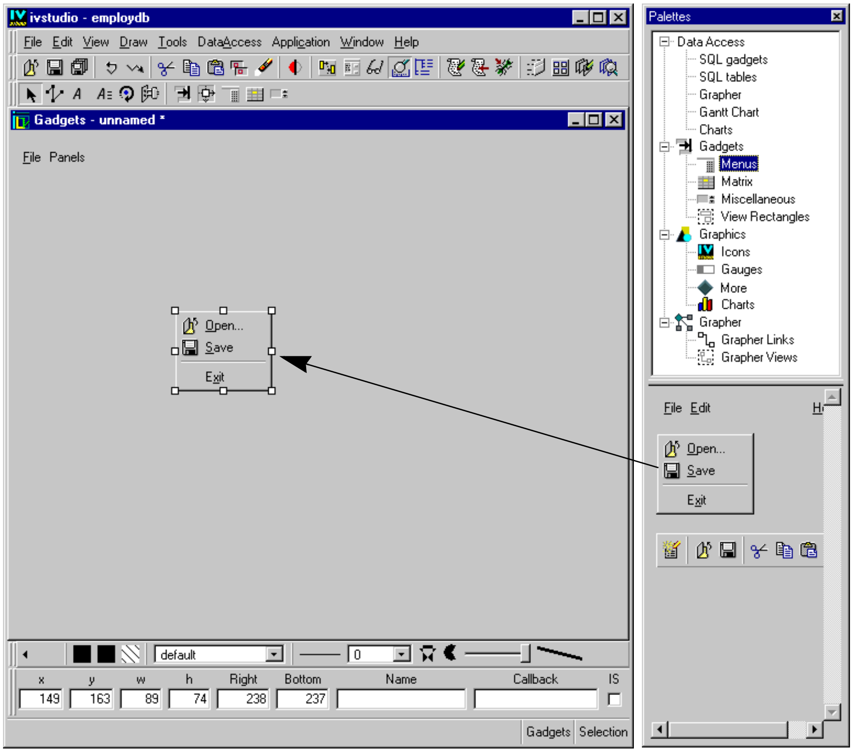
9. Click Apply.

   The menu bar in the Gadgets buffer changes. It now has the two menu items you want for your application, File and Panels.

Creating the File Pop-up Menu

To create the pop-up menu for the File item in the menu bar, follow these steps:

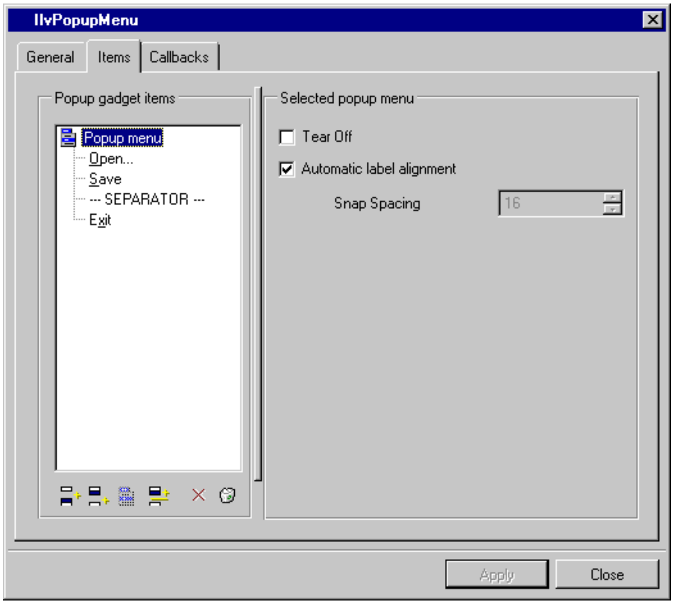
1. In the Palettes panel, select the Menus palette under Gadgets. Drag a pop-up menu gadget to the Gadgets buffer, leaving it selected.



x	y	w	h	Right	Bottom	Name	Callback	IS
149	163	89	74	238	237			□

2. Double-click the pop-up menu gadget in the buffer window to display its inspector.

   The IlvPopupMenu inspector appears.



☐ Tear Off

☑ Automatic label alignment

Snap Spacing 16

> ***Note:*** *Your actions do not affect the pop-up menu in the buffer window until you click Apply in the inspector.*

3. Make sure you are on the Items notebook page. In the Popup gadget items tree on the left side of the page, click `---SEPARATOR---` to select it.

Click the Remove button ✕ under the Popup gadget items tree.

The `---SEPARATOR---` item is removed from the tree gadget in the inspector.

4. Click `Open` in the inspector panel to select it. Then click the Remove button ✕.

The Open menu item is removed from the tree gadget in the inspector.

5. Click `Save` in the inspector panel to select it. Then click the Remove button ✕.

The Save menu item is removed from the tree gadget in the inspector.

6. Click `Exit` in the inspector to select it.

7. In the General notebook page on the right side of the inspector, change the Label field.

Replace the existing text with `^Quit`. (The “^” symbol makes the letter following it underlined, thus highlighting the keyboard equivalent of the command).

The remaining pop-up menu item in the tree gadget becomes `Quit`.

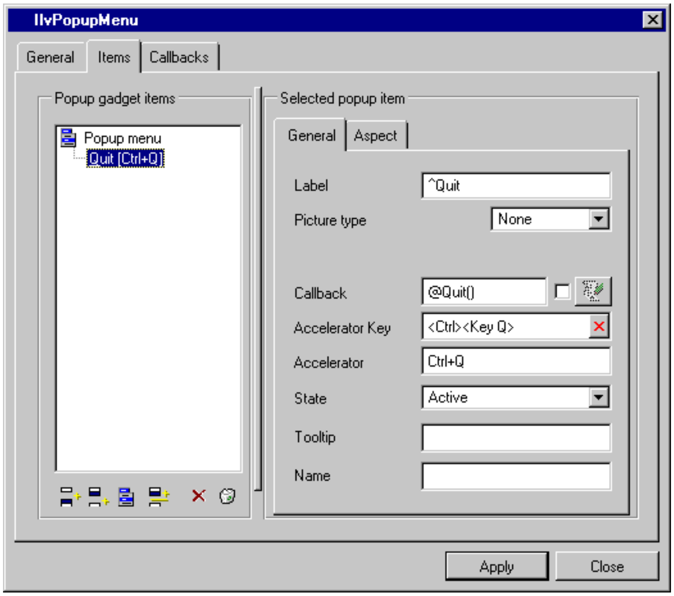
8. Type `@Quit()` in the Callback field.

By using this predefined callback, you can quit the application without coding. For more information, see the section *Using Predefined Callbacks* on page 137.

9. Go to the Accelerator Key field. Press the CTRL key and the Q key at the same time.

When you do this, you will see that `<Ctrl><Key Q>` appears in the Accelerator Key field, `Ctrl+Q` appears in the Accelerator field, and `Active` appears in the State field. These are the actual keys used to execute the accelerator.

The IlvPopupMenu inspector now looks like this:



Label: ^Quit

Picture type: None

Callback: @Quit() □

Accelerator Key: <Ctrl><Key Q>

Accelerator: Ctrl+Q

State: Active

Tooltip:

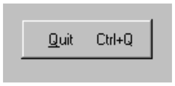
Name:

Notes:

1. *An accelerator allows you to immediately call a callback function (in this case, to quit the application) by pressing the keys defined in the Accelerator Key field. An accelerator is a mechanism that establishes a connection between an event and a function. When a predefined event takes place, such as pressing a key, Data Access recognizes the event and calls the associated function*

2. *The Picture type field allows you to load and display an icon in place of a label for the menu item. When you select Bitmap from the combo box, you can specify the name of a file containing the icon to be displayed. You will not use this feature in the tutorial.*

10. Click Apply in the IlvPopupMenu inspector to have your changes take effect in the current Gadgets buffer window. The button should now appear like this:



11. This step will allow you to see other options you can use for your menu items. It is provided for your information only. For this tutorial, you will want to return the options in their default state once you have tried them to see how they work.

    Select `Quit(Ctrl+Q)` in the Popup gadgets items on the left side of the inspector panel. You can then use the notebook pages of the Selected popup item box to specify other options.

    - In the General page, the State combo box contains a list of options that you can use with the selected item:

      Active When you select this option, the menu item can be activated and appears in normal font in the menu. This is the default behavior.

      None selectionable When you select this option, the menu item cannot be activated, but it is not grayed. The menu item cannot be chosen. You can use this option to make a title within a menu.

      Grayed When you select this option, the menu item is dimmed and cannot be chosen.

      You can try out these options by selecting the option and then clicking Apply in the inspector panel. Then go to the Main window to see the results. To see the results of the None selectionable option, you will have to be in Active mode. When you have finished, make sure you return the State setting to Active.

    - The Aspect page contains several options that you can use for the display of the item:

      Toggle and **Radio** These buttons let you choose the type of check mark that appears when the item is checked. The behavior depends on the look and feel. With Windows and Windows 95, nothing is displayed when the item is not checked. With Motif, a check box is displayed even if the item is not checked and that check box is inverted when the item is checked.

      Checked When you select this option, a symbol appears next to the item in the menu.

      Again, you can try out these options, by selecting them and clicking Apply in the inspector panel. Make sure you return them to their original setting before continuing with the tutorial.

You have now finished creating the File popup menu and will attach it to the File menu in the menu bar.

Attaching the File Pop-up Menu to the File Menu Item

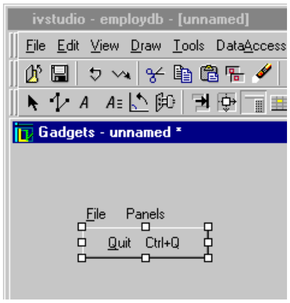
To attach the File pop-up menu to the File menu item, do the following:

1. Click the Menu mode button in the Editing Modes toolbar of the Main window.
2. In the Main window, drag a line from the File pop-up menu you just created to the File menu item in the menu bar.

   As you drag, a line stretches from the pop-up menu to the File menu item, which becomes highlighted. When you release the mouse button, the pop-up menu disappears and is attached to the File menu.

3. For your information only, to make the newly attached pop-up menu reappear, double-click the File label in the menu bar while you are still in Menu editing mode. If you do this, the menu is no longer attached to the menu bar and will need to be reattached (see step 2).

   The newly attached pop-up menu reappears.


ivstudio - employdb - [unnamed]
File Edit View Draw Tools DataAccess
Gadgets - unnamed *
File Panels
Quit Ctrl+Q

> ***Note:*** *Click the Selection editing mode button and double-click the pop-up menu to bring back the IlvPopupMenu inspector for editing if you have closed it.*

4. Click the Active mode button . Then click the Quit menu item to verify its functioning.
5. Click the Selection editing mode button .

Creating and Attaching the Panels Pop-up Menu

Create and attach the pop-up menu for the Panels menu item like you did for the File menu item. The pop-up menu will have three items, each item being the name of one of the windows. Enter these names in the Label field of the IlvPopupMenu inspector:

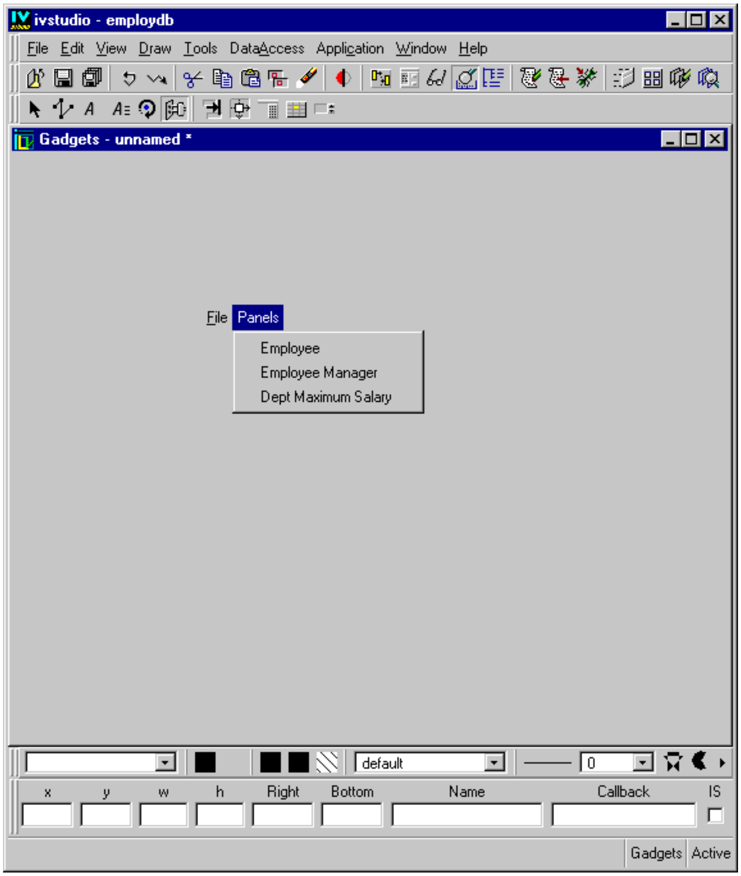
- `Employee`
- `Employee Manager`
- `Dept Maximum Salary`

In the Callback field of the IlvPopupMenu inspector, enter the `@ShowPanel` callback for each menu item, using these panel names:

- `(Employee)`
- `(Empmgr)`
- `(Dpmaxsal)`

Example: `@ShowPanel(Employee)`

When the pop-up menu is finished, it should look like this:



You can test that the pop-up menus appear by clicking the Active button and then selecting the menu bar items. To test the pop-up menu items themselves, you need to test the application (because of the use of certain predefined callbacks). See the section *Testing the Application* on page 158.

Additional Information About Adding Items to a Pop-Up Menu

You can use the buttons on the Items notebook page of the IlvPopupMenu inspector to insert or remove items from a pop-up menu gadget. Note that your changes will not take effect in the pop-up menu gadget in the Main window until you press the Apply button.

Insert You create a new item (labeled `item`) in the pop-up menu list each time you press the Insert button. Items are created in the list before a selected item, or at the bottom if no item is selected. If you make a mistake, you can select an item and click the Remove button to delete it.

Add Use this button to create a new item after the selected item in your list.

Popup menu Use this button to add another level of menus under the selected item in the pop-up menu list. A new pop-up menu item (labeled `Popup menu`) with a new item (labeled `Item`) under it are added to the tree structure.

Separator This button allows you to insert a separator line between pop-up menu items.

Remove Use this button to remove a selected item from the pop-up menu tree gadget.

Clean This button allows you to remove all the items in the pop-up menu list. The Popup menu item at the top of the tree gadget is the only one that remains after you press the Clean button.

Finishing the Control Window

For more practice, you can apply resources to the menu gadgets as described in the section *Using Resources* on page 99.

> ***Note:*** *You probably would not want to use the Fit to View... command, since the window will automatically take the size of the menu bar, which is too long for the number of menu items. See the section Setting the Size of the Window on page 102.*

6. Attach the menu bar gadget to the window as described in the section *Setting Attachments* on page 106.

7. Change the background color of the window as described in the section *Setting the Background of the Window* on page 104.

Saving the Control Window

If you have not already saved the window, do so now by choosing Save As... from the File menu in the Main window. Give it the name `control.ilv` and save it in the directory in which you saved the `.iva` application file in the section *Creating the Application File* on page 50.

Adding the Control Window to the Application

Add the Control window to the `employdb` application by following the procedure described in the section *Adding the Employee Window to the Application* on page 112. Do not do steps 6 and 7 since you want the Control window to be visible when the application is launched.

Using the Browser

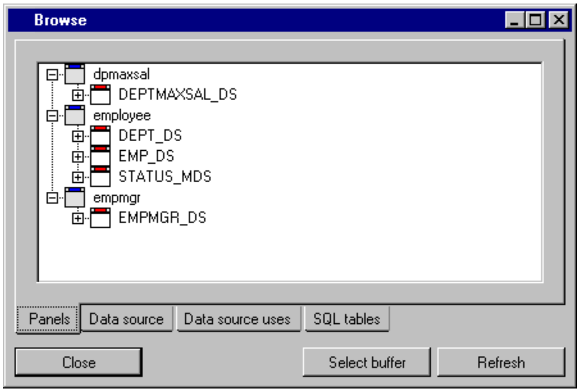
You can use the Browser to get a closer look at the structure of your application. To open the Browser, select Browse... from the Data Access menu of the Main window.

The Browser shows how the objects (data sources, gadgets, and so on) of the application are related to one another.

There are four notebook pages in this window:

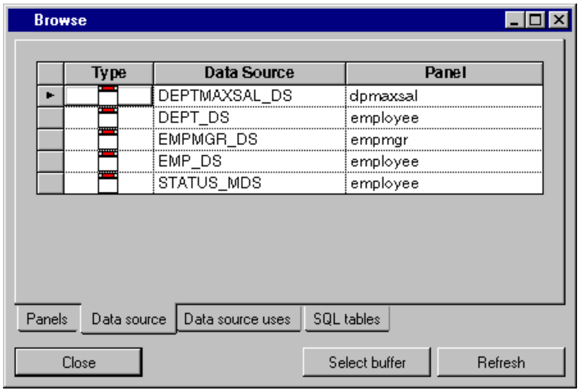
◆ Panels

For each panel in the application, it shows the data sources contained in the panel.



◆ Data source

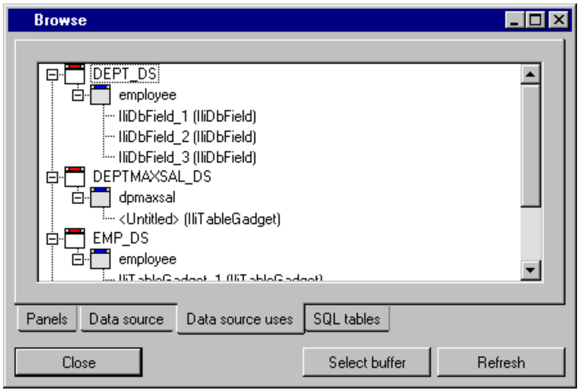
For each data source in the application, it shows the panel in which the data source is located.



Type	Data Source	Panel
	DEPTMAXSAL_DS	dpmaxsal
	DEPT_DS	employee
	EMPMGR_DS	empmgr
	EMP_DS	employee
	STATUS_MDS	employee

- Data source uses

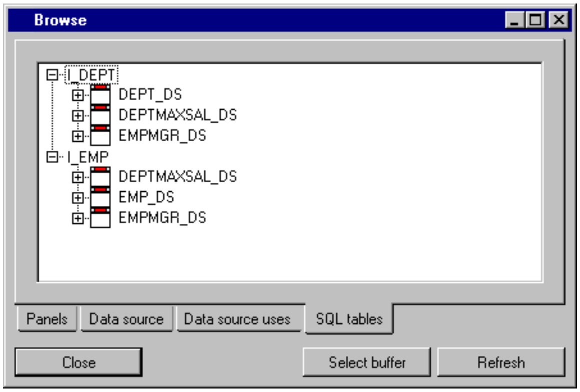
  For each data source in the application, it shows the gadgets that are connected to the data source. The gadgets are grouped by panels.



DEPT_DS
employee
IliDbField_1 (IliDbField)
IliDbField_2 (IliDbField)
IliDbField_3 (IliDbField)
DEPTMAXSAL_DS
dpmaxsal
<Untitled> (IliTableGadget)
EMP_DS
employee
IliTableGadget_1 (IliTableGadget)

- SQL tables

  For each SQL data source in the application, it shows the database table used by the data source (in its FROM clause).



> ***Note:*** *Only the data sources located in panel instances of the application are shown in the Browser. Any data source that would be located in a panel not directly instantiated in the application (such as a notebook page) is not shown in the Browser.*

Testing the Application

The final operation is to test the application that you have created. To do so, follow these steps:

1. Choose <Application> from the Window menu in the Main window. The application buffer is displayed.
2. Click the Test button in the Main window toolbar.

   The Control window you created appears. The other windows do not appear, since you made them invisible upon launching when you added each of them to the application.
3. Choose a window from the Panels menu in the test Control window.

   The window you chose appears.
4. Choose Quit in the File menu or use the accelerator keys Ctrl+Q.

   The application test windows disappear.

Generating the Application

Depending on the language(s) used to code the application logic, there are two ways to deliver a running application:

- If the application uses only IBM ILOG Script and predefined callbacks, the application runner can be used to run the application. It is located at:

  ```
  $ILVHOME/bin/runner/<system>/<database>/runner
  ```

  This program accepts the name of an application `.iva` file as an argument:

  ```
  runner <application>.iva
  ```

  By default it shows, along with the visible panels of the application, a small window containing an “exit” button. The `-noexitpanel` option can be used to prevent this window from being shown:

  ```
  runner -noexitpanel <application>.iva
  ```

- If the application also contains C++ callbacks, you need to generate C++ source files for the application and for each panel class. Then compile and link the application program using a C++ compiler. For more information see the chapter on generating code in the *IBM ILOG Views Controls User's Manual*.

Summary

You have now completed the main part of the Data Access tutorial. You have learned the basic procedures for creating SQL database applications using Data Access and should be able to create your own applications.

If you find that you need more information as you create your application, do not hesitate to consult the other IBM ILOG documents released with your IBM® ILOG Views package or to make use of IBM support services.

If you are using an Oracle database in your application, the next chapter describes how to use the nested tables feature of Data Access. The nested tables feature is currently only available for users of Oracle 9i or later.

CHAPTER

6

Nested Tables

This chapter is a supplement to the tutorial for those users interested in the nested tables feature of Data Access.

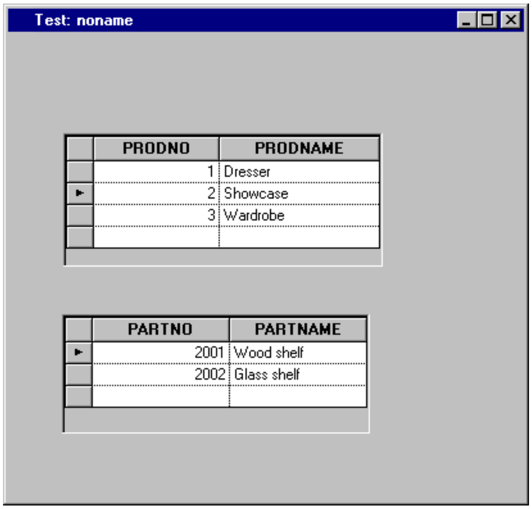
> ***Note:*** *Currently, the nested tables feature of Data Access is only supported for Oracle 9i or later.*

What You Are Going To Do

In this chapter, you are going to create a new buffer containing a Product window by performing the following operations:

- *Setting Up the Database*
- *Defining the PRODUCT_DS SQL Data Source*
- *Defining the PARTS_DS SQL Data Source*
- *Creating the Product Window*
- *Saving the Product Window*

When finished, it will look like this:



PRODNO	PRODNAME
1	Dresser
2	Showcase
3	Wardrobe

PARTNO	PARTNAME
2001	Wood shelf
2002	Glass shelf

Setting Up the Database

This tutorial uses a database schema with a single table called PRODUCT. Use your own database system tools to create the table, using the following SQL statements as a model. The statements are based on Oracle syntax.

- PARTS_T Type

```
CREATE TYPE PARTS_T AS OBJECT (
PARTNO    INTEGER      NOT NULL,
PARTNAME  VARCHAR(20)  NOT NULL)
```

- PARTS_TABLE_T Type

```
CREATE TYPE PARTS_TABLE_T AS TABLE OF PART_T
```

- PRODUCT Table

```
CREATE TABLE PRODUCT (
PRODNO    INTEGER      NOT NULL,
PRODNAME  VARCHAR(50)  NOT NULL,
PARTS     PARTS_TABLE_T,
PRIMARY KEY (PRODNO))
```

The column PARTS in the PRODUCT table contains the nested tables.

- The PRODUCT table contains the following data:

PRODNO	PRODNAME	PARTS	
		PARTNO	PARTNAME
1	Dresser	1001 1002	Drawer Handles
2	Showcase	2001 2002	Wood shelf Glass shelf
3	Wardrobe	3001 3002 3003	1 meter shelf Rod Basket

Defining the PRODUCT_DS SQL Data Source

Define the PRODUCT_DS data source by doing the following:

1. Choose New > Gadgets from the File menu.

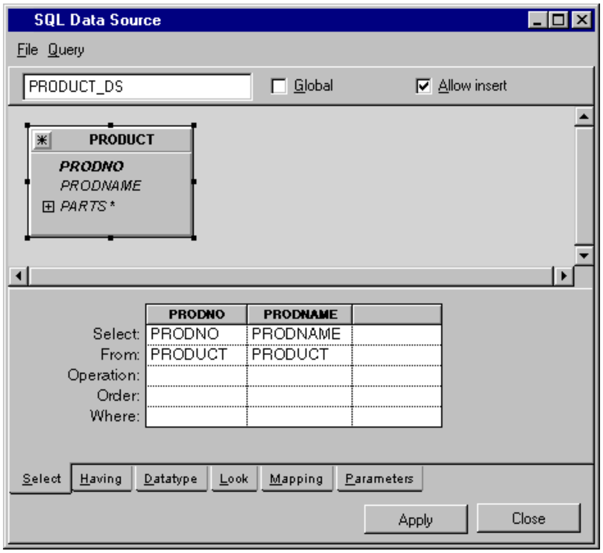
   A new Gadgets buffer window is opened ready for you to edit.

2. Define the PRODUCT_DS data source. Drag an SQL data source gadget from the Palettes panel to the Main Window. Double-click the SQL data source gadget to display the SQL Data Source inspector.

   Type the name `PRODUCT_DS` in the Name field of the SQL Data Source inspector.

   Select Add Tables... from the Query menu of the SQL Data Source inspector. When the Select Tables panel appears, select the PRODUCT table in the left list and move it to the right list by clicking the -> button. Click OK.

3. Add the PRODNO and PRODNAME columns (but not the PARTS column) to the PRODUCT_DS data source by dragging lines from the data source in the FROM section to the SELECT section. Click Apply.



PRODUCT_DS

□ Global

☑ Allow insert

	PRODNO	PRODNAME	
Select:	PRODNO	PRODNAME	
From:	PRODUCT	PRODUCT	
Operation:			
Order:			
Where:			

Defining the PARTS_DS SQL Data Source

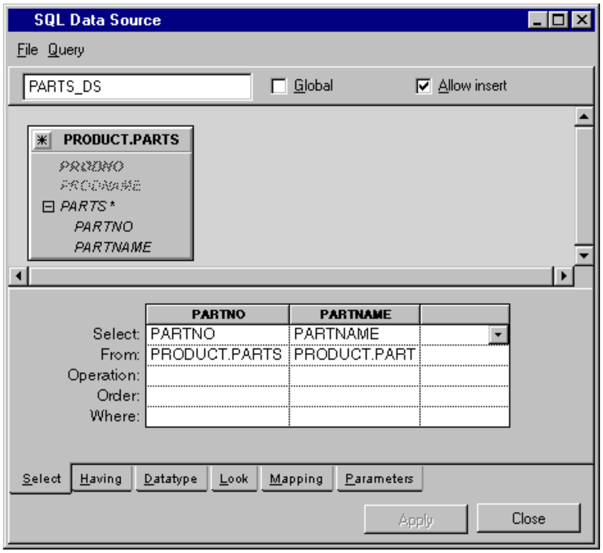
Define the PARTS_DS data source by doing the following:

1. Drag an SQL data source gadget from the Palettes panel to the Gadgets buffer.

   Type the name `PARTS_DS` in the Name field of the SQL Data Source inspector.

   Select Add Tables... from the Query menu of the SQL Data Source inspector panel. When the Select Tables panel appears, select the PRODUCT table in the left list and move it to the right list by clicking the -> button. Click OK.

2. Click the + sign to the left of the PARTS column. Select the nested columns PARTNO and PARTNAME and add them to the PARTS_DS data source by dragging lines to the SELECT section:



PARTS_DS □ Global ☑ Allow insert

	PARTNO	PARTNAME	
Select:	PARTNO	PARTNAME	
From:	PRODUCT.PARTS	PRODUCT.PART	
Operation:			
Order:			
Where:			

3. Double-click the table PRODUCT.PARTS in the FROM section of the inspector panel or select the Edit Table from the Query menu.

   Edit the Parent and Alias items in the dialog box as follows:

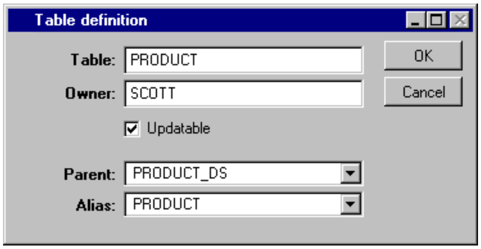
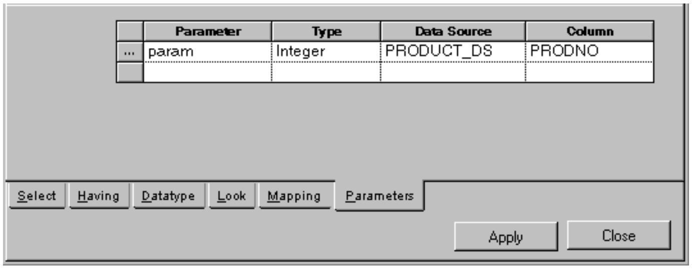


Table: PRODUCT
Owner: SCOTT
☑ Updatable
Parent: PRODUCT_DS
Alias: PRODUCT

   Click OK. As a result, the PARTS_DS data source holds a table that edits the nested PARTS table of the PRODUCT designated in the current row of the PRODUCT_DS data source.

4. In the SQL Data Source inspector, click the Parameters page to make it active. Fill in the parameters table cells as shown below. Click Apply.



	Parameter	Type	Data Source	Column
...	param	Integer	PRODUCT_DS	PRODNO

5. Select Properties from the File menu in the SQL Data Source inspector panel.

   The SQL Data Source Properties panel appears.

6. In the SQL Data Source Properties panel, change the Auto select property from `No` to `Yes`. Click OK.

   The combination of the parameter and the auto select mode is used so that PARTS_DS rows are selected as soon as you change row in PRODUCT_DS data source.

7. Click Apply in the SQL Data Source inspector.

Creating the Product Window

You are now going to connect table gadgets to the data sources.

1. Drag a table gadget to the Gadgets buffer and connect it to the PRODUCT_DS data source. In the Table Gadget inspector, enter `PRODUCT_DS` in the Data source field. Click Apply.

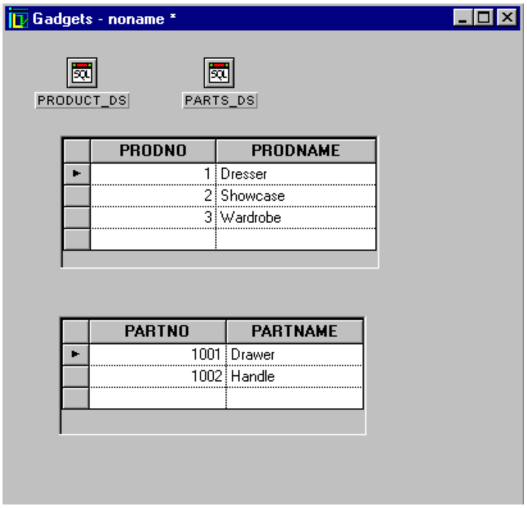
2. Drag a table gadget to the Main window and connect it to the PARTS_DS data source. Enter `PARTS_DS` in the Data source field of the Table Gadget inspector. Click Apply.

3. Click the Active mode button in the Main window. Click in the first table gadget (PRODUCT_DS) and press the F9 key to refresh the data sources.

   The PRODUCT_DS table gadget in the Gadgets buffer changes to display the products number and name as the PARTS_DS table gadget changes to display the parts of the product selected in the PRODUCT_DS table gadget.

   Changing row in the PRODUCT_DS table gadget selects the product parts in the PARTS_DS table gadget.

   The Products window now looks like this:



PRODNO	PRODNAME
1	Dresser
2	Showcase
3	Wardrobe

PARTNO	PARTNAME
1001	Drawer
1002	Handle

Saving the Product Window

If you have not already saved the window, do so now by choosing Save As 💾 from the File menu in the Main window. Give it the name `product.ilv`.

You have now finished this brief tutorial supplement on using the nested tables feature of Data Access.

APPENDIX

A

Setting Up Data Access

In order for Data Access to run, IBM® ILOG Views and IBM ILOG Views DB Link must be installed if you want to use the SQL gadgets.

This appendix provides instructions for setting up IBM ILOG Views, IBM ILOG Views DB Link, and Data Access.

The following topics are included:

- *Data Access Installation and Configuration*
- *Setting Up IBM ILOG Views Views, IBM ILOG Views DB Link, and Data Access on UNIX*
- *Setting Up IBM ILOG Views and Data Access on Windows XP and Vista*
- *Setting Up IBM ILOG Views and Data Access on Windows XP and Vista*
- *Setting Up the IBM ILOG Views DB Link Initialization File on Windows*

Data Access Installation and Configuration

This section explains how to install Data Access on various platforms. For the latest list of platforms on which you can use Data Access, see the table in the `README` file delivered with the product.

Installation Directories

We assume that you have already installed all needed packages and products, and have configured the IBM ILOG License Manager (ILM). These procedures are explained in the *Installation Guide* and the *IBM ILOG License Manager Reference Manual* provided with the Data Access package.

We also assume that the products have been installed in the following directories:

For UNIX users:

- IBM ILOG Views:`/usr/ilog/viewsXX` (where XX stands for the version number)
- IBM ILOG Views DB Link:`/usr/ilog/dblinkXX` (where XX stands for the version number)
- Data Access:`/usr/ilog/viewsXX` (where XX stands for the version number)

For Windows XP or Windows Vista users:

- IBM ILOG Views:`C:\ILOG\VIEWSXX` (where XX stands for the version number)
- IBM ILOG Views DB Link:`C:\ILOG\DBLINKXX` (where XX stands for the version number)
- Data Access:`C:\ILOG\VIEWSXX` (where XX stands for the version number)

If this is not the case, please replace subsequent references to these directories with your actual installation directory.

Please refer to the `README` file delivered with Data Access to determine which versions of IBM® ILOG Views DB Link and IBM ILOG Views are required with the version of Data Access you are using.

Data Access and the Packages

To use Data Access, you must install the “Foundation” and “Gadget” Views packages and IBM® ILOG Views DB Link product. To use other optional packages with Data Access, you must install the relevant IBM ILOG Views packages:

- IBM ILOG Views DB Link for SQL
- IBM ILOG Views Grapher package
- IBM ILOG Views Gantt package for Gantt charts
- IBM ILOG Views compatibility Package for old charts
- IBM ILOG Views Charts package

System Name

Before you continue reading, you should determine the name of your system. You can find it in the `README` file delivered with the product. For example, if you are using a Sun Ultra Sparc 32 bit Solaris 8.x with C++ 6.2, your system name is `ultrasparc32_8_6.2`.

In the sections that follow, replace the strings `<system>` and `<subsystem>` in commands and directory names with your actual system and subsystem names.

System Environments

The next three sections show how to install IBM® ILOG Views in these different environments:

- UNIX
- Windows XP
- Windows Vista

Since the sections are independent, you can go directly to the one that applies to you.

Setting Up IBM ILOG Views Views, IBM ILOG Views DB Link, and Data Access on UNIX

1. Set the `ILVHOME` and `DBLINK_HOME` variables.

   - If using `sh` or one of its derivatives, such as `ksh` or `bash`, type:

     ```
     $ ILVHOME=/usr/ilog/viewsXX
     $ DBLINK_HOME=/usr/ilog/dblinkXX
     $ export ILVHOME DBLINK_HOME
     ```

   - If using `csh` or `tcsh`, type:

     ```
     % setenv ILVHOME /usr/ilog/viewsXX
     % setenv DBLINK_HOME /usr/ilog/dblinkXX
     ```

   The setting up of these variables can be automated by putting the above commands in your shell startup file (either `.profile` if using `sh,` or `.login` if using `csh`).

2. If you are using Sun Ultra Sparc with Solaris 8.x, set up the shared library path variable.

   The directory containing the IBM® ILOG Views shared libraries must be added to the shell variable `LD_LIBRARY_PATH`.

   You should set the `LD_LIBRARY_PATH` variable.

- If using `sh` or one of its derivatives, type:

```
$ LD_LIBRARY_PATH=$ILVHOME/lib/<system>/shared:$LD_LIBRARY_PATH
```

```
$ LD_LIBRARY_PATH=$ILVHOME/studio/
<system>shared:$LD_LIBRARY_PATH
```

```
$ LD_LIBRARY_PATH=$DBLINK_HOME/lib/<system>/
shared:$LD_LIBRARY_PATH
```

```
$ export LD_LIBRARY_PATH
```

- If using `csh` or `tcsh`, and the variable `LD_LIBRARY_PATH` *is not already defined*, type:

```
% setenv LD_LIBRARY_PATH $ILVHOME/lib/<system>/shared
```

```
% setenv LD_LIBRARY_PATH $ILVHOME/studio/<system>/
shared:$LD_LIBRARY_PATH
```

```
% setenv LD_LIBRARY_PATH $DBLINK_HOME/lib/<system>/
shared:$LD_LIBRARY_PATH
```

- If using `csh` or `tcsh`, and the variable `LD_LIBRARY_PATH` *is already defined*, type:

```
% setenv LD_LIBRARY_PATH $ILVHOME/lib/<system>/
shared:$LD_LIBRARY_PATH
```

```
% setenv LD_LIBRARY_PATH $ILVHOME/studio/<system>/
shared:$LD_LIBRARY_PATH
```

```
% setenv LD_LIBRARY_PATH $DBLINK_HOME/lib/<system>/
shared:$LD_LIBRARY_PATH
```

The setting up of these variables can be automated by putting the above commands in your shell startup file (either `.profile` if using `sh`, or `.login` if using `csh`).

Setting Up IBM ILOG Views and Data Access on Windows XP and Vista

> ***Note:** For IBM ILOG Views DB Link, you do not need to set up environment variables except for compiling, which is described in the* `README` *file.*

1. Set the `ILVHOME` variable.

   The `ILVHOME` variable must be set to the IBM® ILOG Views installation directory. To do this, open the Configuration Panel in the Main group, then double-click on the System icon. This opens a panel where you can define new variables. Use this panel to define the `ILVHOME` variable as:

   ```
   C:\ILOG\VIEWSXX
   ```

Do not close the panel as you will need to use it again.

or add the following lines to the `VIEWS.INI` file:

```
[Ilog Views]
IlvHome=C:\ILOG\VIEWSXX
```

The file must be located in the directory where Windows NT is installed.

2. Set the `PATH` variable.

   The directory containing IBM ILOG Views DLLs must be added to the `PATH` variable. To do this, use the System panel to define the `PATH` variable as:

   ```
   C:\ILOG\VIEWSXX\LIB\<system>\<subsystem>;%PATH%
   C:\ILOG\DBLINKXX\LIB\<system>\<subsystem>;%PATH%
   ```

   or copy the DLLs into a directory of the path.

3. Click OK to validate your changes.

Setting Up the IBM ILOG Views DB Link Initialization File on Windows

For Windows XP and Windows Vista, the IBM® ILOG Views DB Link dynamic loader requires that the IBM ILOG Views DB Link initialization file `dblink.ini` be located:

- In each directory in which Data Access programs are located.

or

- In the Window system directory.

  Initially, the `dblink.ini` file is installed in the following location:

  ```
  %ILVHOME%\bin\<system>\mulitdb\dblink.ini
  ```

We recommend copying it from this location to the Windows system directory so that Data Access programs can be run indifferently from any location.

Setting Up Database-Specific Variables

Certain database-related environment variables also need to be set up. Depending on the database you are using, you will need to define environment variables that will enable Data Access to locate your database server. Here are the names of these variables for each supported database.

- DB2: DB2DIR
- Oracle: `ORACLE_HOME`
- Informix: `INFORMIXDIR` and `INFORMIXSERVER`

- Sybase: `SYBASE`

For more information, refer to the documentation that comes with your database software.

APPENDIX

B

Format Syntax

This appendix contains the symbols and formats you can use to create application-wide named formats and local formats for a particular field.

A format specification controls the way a value will be formatted for display. For each type of format specification, there is a set of special symbols, each having a specific meaning. You can find information on the following types of formats:

- *String Formats*
- *Number Formats*
- *Date Formats*
- *Literal Characters*

String Formats

String formats refer to the formatting of text. You can use the following symbols to specify a string format.

Symbol	Description
!	Formatting must proceed from right to left
<	Following characters will be converted to lowercase
>	Following characters will be converted to uppercase
@	Placeholder for a mandatory character
&	Placeholder for an optional character

Normally, character string formatting proceeds by scanning the character string value and the format specification from left to right. However, if the format specification contains a “!” symbol, then scanning of both the character string value and the format specification proceeds from right to left.

Examples

Value	Format	Result
forms	>	FORMS
FormS	<	forms
forms	<@>	fORMS
forms	<@@>	foRMS
forms	!>@<	FORMs
forms	!>@<@@@>	FoRMS
forms	&&&&&“data”	formsdata
forms	@@@@@“data”	forms data

The @ symbols are replaced by spaces when there is no corresponding character in the string value.

Number Formats

Number formats refer to the formatting of numbers, including currency amounts. You can use the following symbols to specify a number format.

Symbol	Description
0	Placeholder for a mandatory digit
#	Placeholder for an optional digit
.	Placeholder for the decimal point
,	Placeholder for the thousands separator
%	Formats the value as a percentage
E	Placeholder for the exponent (displayed in uppercase)
e	Placeholder for the exponent (displayed in lowercase)

The exponent symbols “E” and “e” may be followed by a “+” or “-” sign. A “-” sign means that the exponent sign must be displayed only if it is negative, whereas a “+” sign or no sign means that the sign of the exponent is always displayed.

When a “%” sign appears in the format specification, the numeric value to be formatted is multiplied by 100 before formatting and a “%” sign appears in the result.

“0” symbols are replaced by zeros when there is no corresponding digit in the number value.

Examples

Value	Format	Result
1234.567	#,##0.00	1 234.57
1234.567	#,##0.0#	1 234.57
1234.5	#,##0.00	1 234.50
1234.5	#,##0.0	1 234.5
1.5	0,000.00	0 001.50
1234	0.00 E+00	1.23 E+03
1234	0.00 E-00	1.23 E03
1234	0.00 E-##	1.23 E3
0.5432	#.# %	54.3 %

The characters used to represent the decimal point and the thousands separator in the formatted result depend on application settings that can be changed. Typically, they depend on the country where the application is used. These settings do not affect the symbols used as placeholders for the decimal point and for the thousands separator in format specifications. You should thus always use the placeholders listed above for these values as only the output depends on the application settings.

Note that the maximum precision is 15 digits for double values and 7 for float values.

Date Formats

Date formats refer to the formatting of days, dates and times. You can use the following symbols to specify a date format.

Symbol	Description
/	Placeholder for date separator
:	Placeholder for time separator
<	Following characters will be converted to lowercase
>	Following characters will be converted to uppercase
d	Placeholder for day of month (1-31)

Symbol	Description
dd	Placeholder for day of month (01-31)
ddd	Placeholder for day of week (Sun-Sat). Depends on the language setting for the application (see `IlvDisplay`)
dddd	Placeholder for day of week (Sunday-Saturday). Depends on the language setting for the application (see `IlvDisplay`)
ddddd	Placeholder for full date (ex: 8/3/96). Depends on the global settings for the application (see `IliFormat`)
dddddd	Placeholder for full date (ex: 03 August 1996). Depends on the language setting (see `IlvDisplay`) and the global settings (see `IliFormat`) for the application
w	Placeholder for day of week (1-7)
ww	Placeholder for week of year (1-53)
m	Placeholder for month (1-12)
mm	Placeholder for month (01-12)
mmm	Placeholder for month (Jan-Dec). Depends on the language setting for the application (see `IlvDisplay`)
mmmm	Placeholder for month (January-December). Depends on the language setting for the application (see `IlvDisplay`)
q	Placeholder for quarter (1-4)
y	Placeholder for year day (1-366)
yy	Placeholder for year (00-99)
yyyy	Placeholder for year (1970-2099)
h	Placeholder for hour (0-23)
hh	Placeholder for hour (00-23)
H	Placeholder for hour (0-11)
HH	Placeholder for hour (00-11)

Symbol	Description
p	Placeholder for AM or PM
n	Placeholder for minutes (0-59)
nn	Placeholder for minutes (00-59)
s	Placeholder for seconds (0-59)
ss	Placeholder for seconds (00-59)
ttttt	Placeholder for full time (ex: 05:32:12). Depends on the global settings for the application (see `IliFormat`)

Examples

Value	Format	Result
`12 jan 96`	`d/m/yy`	`12/1/96`
`12 jan 96`	`d mmmm yyyy`	`12 January 1996`
`12 jan 96`	`q`	`1`

The placeholders for the date and time separators are formatted according to application settings that may vary (typically, depending on the country).

Also, two of the format specifications depend on application settings that control wether the date should be displayed before or after the month. For example:

Value	Format	Application properties	Result
`12 jan 96`	`dddddd`	MDY, English language	`January 12 1996`
`12 jan 96`	`dddddd`	DMY, French language	`12 Janvier 1996`

Literal Characters

In any format specification, you can include literal characters. They will be output “as is” when formatting a value.

Symbols

A literal character is specified by one of the three methods below:

- **`\c`** Prefix the character with a back slash.
- **`“abc”`** Enclose a string of characters in double-quotes.
- Any character that is not a special symbol, or cannot be part of one, is considered as being a literal character.

Examples

Value	Format	Result
`1234.5`	`#,##0.0# Frs`	`1 234.5 Frs`
`1234.5`	`#,##0.0# “Frs”`	`1 234.5 Frs`
`1234.5`	`#,##0.0# \F\r\s`	`1 234.5 Frs`
`12 jul 96`	`“Quarter” q`	`Quarter 3`
`forms`	`ILOG >@<`	`ILOG Forms`

APPENDIX

C

SQL Schema Editor

This appendix describes the SQL Schema Editor. This editor is provided if you need a simple editor to create, delete, or edit tables in a database for the examples in this manual or for other uses.

You can use the procedure described in this appendix to create the database schema for the examples used in the tutorial.

You can find the following topics:

- ◆ *Using the SQL Schema Editor*
- ◆ *Creating the Tutorial Database Schema*
- ◆ *Other Features*

Using the SQL Schema Editor

A schema is the structure in the form of tables in which the data is stored. You use the Schema Editor to edit the table definitions and the data contained in the tables.

Data Access provides you with a simple schema editor to allow you to immediately get started with the example in this tutorial or for other uses. You can, of course, use your DBMS schema editing tools if you prefer.

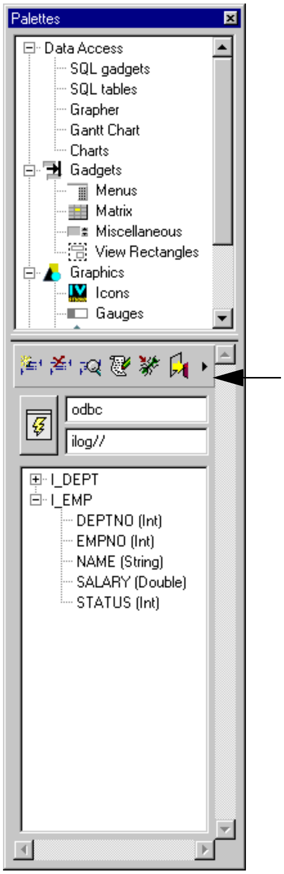
Warning: *Use the SQL Schema Editor only in the design phase of your database. If you use the SQL Schema Editor for tables that contain data, this data will be lost when you export the table definition. To change the schema for tables that already contain data, check your database system documentation for the appropriate tools.*

Creating the Tutorial Database Schema

To create the schema for the example, do the following:

1. Choose the SQL Tables from the Data Access palette in Studio.

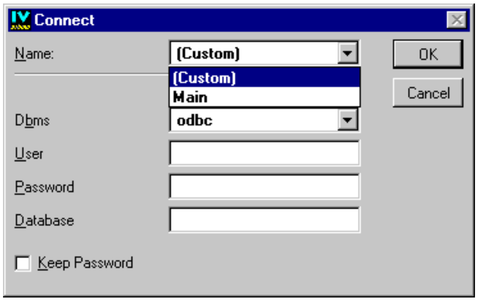
The SQL Schema Editor appears.



SQL Schema Editor

2. To connect to the database, click the Connect button from the SQL Tables palette.

   The Connect panel appears. The elements composing this panel depend on the database system being used (for example, you can have an Options and/or Database field, depending on your system). The Connect panel for an ODBC system is shown here.



Connect

Name: (Custom)

(Custom)
Main

Dbms: odbc

User

Password

Database

□ Keep Password

3. Type the data needed for accessing the database:

   - **User** and **Password**—(Click the Keep Password button if you want the password to appear the next time you connect.)
   - **Database** field—The name of the database. This field is database dependent.

4. To define the table, click the Create table button from the SQL Schema Editor toolbar.

   The Table Definition panel appears.

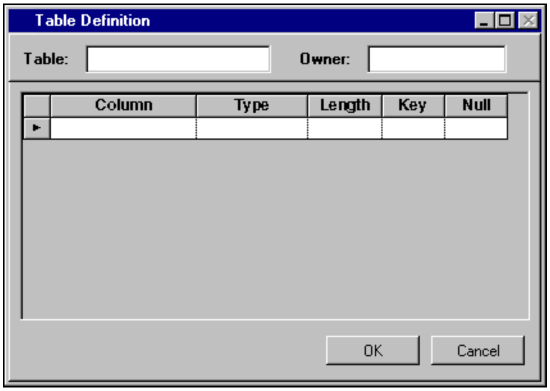


Table Definition

Table:

Owner:

	Column	Type	Length	Key	Null
►					

5. To give a name to the table, type `I_EMP` in the Table field.

6. Make sure the Owner field is correct. By default, the name displayed in the User field of the Connect panel is displayed here. If you want another owner, type that name.

7. Define the five columns for the I_EMP table by entering the information in the cells as shown in the following figure. (Note `Yes` in the Null column for SALARY.):

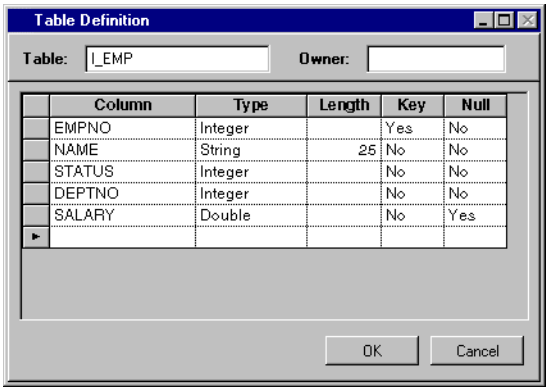


Table: I_EMP

Owner:

Column	Type	Length	Key	Null
EMPNO	Integer		Yes	No
NAME	String	25	No	No
STATUS	Integer		No	No
DEPTNO	Integer		No	No
SALARY	Double		No	Yes

For help on editing tables in general, see *Additional Information about Editing Tables* on page 67.

There are five parameters to be edited for each column in the table:

- **Column**—Required. The name of the column.
- **Type**—Required. The type of data to be entered in the column. Combo box provided.
- **Length**—Optional. The maximum number of characters a user can enter in the column. Applicable only when the type is `String`. If left empty, an unlimited number of characters can be entered.
- **Key**—Required: `Yes` or `No`. Selects whether or not this column is to be part of the key for the row. A key uniquely identifies a row. Combo box provided.
- **Null**—Required: `Yes` or `No`. Selects whether or not the column can have a null value. `Yes` means the column can be null. `No` means the user must enter a value in the column. Combo box provided.

8. Click OK.

The table appears in the SQL Tables palette.

9. Define the three columns for the I_DEPT table in the same way as you did for the I_EMP table (see steps 4-8 above). Use the data shown in the following figure. Then click OK.

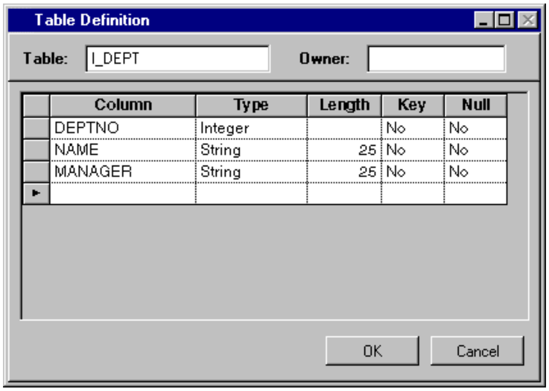


Table: I_DEPT

Owner:

Column	Type	Length	Key	Null
DEPTNO	Integer		No	No
NAME	String	25	No	No
MANAGER	String	25	No	No

10. In the SQL Schema Editor toolbar, click the Export selected table button to export the schema of the selected table of the table list. Choose Save As... from the File menu. A standard file selection dialog box appears.

> ***Note:*** *Data Access generates the SQL statements needed for creating the tables. These statements are then submitted to the Database Management System (DBMS), which creates the table.*

11. Give it the name `EmployDB.ilx`,. Then click Apply. (If necessary, create a directory outside of Data Access in which to save it.)

The schema representation, which is specific to Data Access, is saved. You can use it in the future to reproduce the schema without having to rebuild it. To do so, you would use the Import table button to import a schema representation.

> ***Warning:*** *Remember, only use the SQL Schema Editor to create new tables. If you open and edit tables that already contain data, you will lose the data when you valid the changes.*

Other Features

In addition to creating new tables in the database, you can also edit the data in a table or remove a table from the database.

Deleting a Database Table

To remove a table from the database:

1. Ensure that you are connected to the database.
2. Select the table that you want to remove from the database in the SQL Table list.
3. Click the Drop selected table button to drop the table.
4. A confirmation dialog appears for each table to be deleted.

Editing a Database Table

To edit the data in a table:

1. Ensure that you are connected to the database.
2. Select the table that you want to edit in the SQL Table List.
3. Click the Edit data of the selected table button to edit the table.
4. A window appears showing the table with all the data. You can edit the data in the table or add new data to the table in the same way as you do in a table gadget.
5. When you have finished editing the data for your table, click the Close button.

Index