A Demo Project Using PowerBuilder 2022 R3

TABLE OF CONTENTS

1. INTRODUCTION
   1. INTRODUCTION TO POWERBUILDER AND PROJECT
   2. FUNCTIONAL AND NON FUNCTION REQUIREMENTS
2. BUILDING A CLIENT SERVER APPLICATION
   1. BUILDING A LOGIN WINDOW
   2. CONNECTING TO THE DATABASE
   3. CREATING ANCESTOR WINDOW
   4. SETTING UP THE MENUS
   5. BUILDING DATA WINDOW OBJECTS AND ATTACHING
   6. RUNNING THE DEBUGGER
   7. PREPARING THE APPLICATION FOR DEPLOYMENT
3. AUTOMATING BUILD AND DEPLOY
4. **INTRODUCTION** 
   1. **INTRODUCTION TO POWERBUILDER AND PROJECT**

PowerBuilder is an enterprise development tool that allows you to build many types of applications and components. It is one of a group of Appeon products that together provide the tools to develop client/server, multitier, and Internet applications. A PowerBuilder client application can contain:

* A user interface

Menus, windows, and window controls that users interact with to direct an application.

* Application processing logic

Event and function scripts in which you code business rules, validation rules, and other application processing. PowerBuilder allows you to code application processing logic as part of the user interface or in separate modules called custom class user objects.

**PowerBuilder applications are event driven**

In a client application, users control what happens by the actions they take. For example, when a user clicks a button, chooses an item from a menu, or enters data into a text box, one or more events are triggered. You write scripts that specify the processing that should happen when events are triggered.

Windows, controls, and other application components you create with PowerBuilder each have a set of predefined events. For example, each button has a Clicked event associated with it and each text box has a Modified event. Most of the time, the predefined events are all you need. However, in some situations, you may want to define your own events.

**PowerScript language**

You write scripts using PowerScript, the PowerBuilder language. Scripts consist of PowerScript commands, functions, and statements that perform processing in response to an event.

For example, the script for a button's Clicked event might retrieve and display information from the database; the script for a text box's Modified event might evaluate the data and perform processing based on the data.

The execution of an event script can also cause other events to be triggered. For example, the script for a Clicked event in a button might open another window, triggering the Open event in that window.

**PowerScript functions**

PowerScript provides a rich assortment of built-in functions that can act on the various components of your application. For example, there is a function to open a window, a function to close a window, a function to enable a button, a function to update the database, and so on.

You can also build your own functions to define processing unique to your application.

**Object-oriented programming with PowerBuilder**

Each menu or window you create with PowerBuilder is a self-contained module called an object. The basic building blocks of a PowerBuilder application are the objects you create. Each object contains the particular characteristics and behaviours (properties, events, and functions) that are appropriate to it. By taking advantage of object-oriented programming techniques such as encapsulation, inheritance, and polymorphism, you can get the most out of each object you create, making your work more reusable, extensible, and powerful.

**Installable cloud apps**

PowerBuilder provides the PowerServer Toolbar to help you deploy your PowerBuilder applications as installable cloud applications.

Installable cloud applications are deployed to the server and installed and run over the Internet. It is made up of the client side (most of which are the same as in PowerBuilder traditional client/server applications) and the server side (which are industry-standard REST APIs for accessing the database and services).

**Multitier applications**

PowerBuilder lets you build applications that run in a distributed computing environment. A multitier application lets you:

* Centralize business logic on servers, such as JBoss, WebLogic, WebSphere, or COM+
* Partition application functions between the client and the server, thereby reducing the client workload
* Build scalable applications that are easy to maintain

**Database connectivity**

PowerBuilder provides easy access to corporate information stored in a wide variety of databases. Data can be accessed through the PowerBuilder ODBC interface, or through a native or direct connection to a database.

PROJECT:

This project aims to create a simple database application with customers and products.

* 1. FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

1. BUILDING A CLIENT SERVER APPLICATION

Create a new workspace

Create a target

[Specify an icon for the application](https://docs.appeon.com/pb2025/getting_started/XREF_76881_Specify_an_icon.html)

Change the size of the main window

2.1 BUILDING A LOGIN WINDOW

Create a new window

Add controls to the window

Change the tab order on the window

Code some Help events and preview the window

Write the script to open the window

2.2 CONNECTING TO A DATABASE

Look at the Demo Database

Run the Connection Object wizard

Declare a global variable

Modify the connection information

Complete the login and logout scripts

Run the application

2.3 CREATING ANCESTOR WINDOW

Add a library to the search path

Create a new ancestor sheet window

Create a new sheet window inheritance hierarchy

Add a Data Window control for the master Data Window

Add a Data Window control for the detail Data Window

View the scripts inherited from the user object

Add user events and event scripts

Add scripts to retrieve data for the Data Window controls

Adjust a runtime setting for sheet window size

2.4 SETTING UP THE MENUS

Modify the frame menu

Create a new sheet menu

Add menu scripts to trigger user events

Attach the new menu and run the application

2.5 BUILDING DATA WINDOW OBJECTS AND ATTACHING

Create and preview a new Data Window object

Save the Data Window object

Make cosmetic changes to the first Data Window object

Create a second Data Window object

Make cosmetic changes to the second Data Window object

Attach the Data Window object to the master Data Window control

Attach the Data Window object to the detail Data Window control

Run the application

Attach Data Window objects to the Product window

Run the application again

2.6 RUNNING THE DEBUGGER

Add breakpoints in application scripts

Run in debug mode

Set a watch and a conditional breakpoint

2.7 PREPARING THE APPLICATION FOR THE DEPLOYMENT

Create the Project object

Create the executable file

Create a shortcut

Test the executable file

1. AUTOMATING THE BUILD AND DEPLOY