

Саликов Валентин Александрович

*доцент кафедры автоматизированных систем обработки информации
Днепропетровский национальный университет им. О. Гончара*

Salikov V. A.

Sc.D., associate professor

Oles Honchar Dnipropetrovsk National University

Сторчак Светлана Александровна

студент 4-го курса кафедры АСОИ

Днепропетровского национального университета им. О. Гончара

Storchak S. A.

student

Oles Honchar Dnipropetrovsk National University

РАЗРАБОТКА ПРИЛОЖЕНИЯ КЛИЕНТОВ БАЗЫ ДАННЫХ ПО ТЕХНОЛОГИИ PowerBuilder

CLIENT APPLICATION DEVELOPMENT FOR DATABASE WITH PowerBuilder TECHNOLOGY

Аннотация. В статье описывается процесс разработки клиентского приложения базы данных в PowerBuilder с использованием технологии окон данных DataWindow.

Ключевые слова: информационная система, база данных, PowerBuilder, DataWindow, профиль базы данных, источник данных, пользовательский интерфейс, SQL, событие, элемент управления.

Summary. Article describes the process development of database client application using DataWindow technology in PowerBuilder.

Key words: information system, database, PowerBuilder, DataWindow, database profile, data source, user interface, SQL, event, element control.

Вопросам разработки приложений посвящены фундаментальные работы [1,2]. Однако написаны они для PowerBuilder версий 4.0–6.0. С тех пор интерфейс Builder для версий 10.0–12.0 существенно изменился, что затрудняет использование [1,2]. В данной статье авторами подробно рассмотрен процесс разработки типичного клиентского приложения в PowerBuilder 10.5 на примере создания GUI для базы данных (БД) сельской библиотеки [3].

Разработка приложения в Builder начинается с создания нового рабочего пространства (File → New → Workspace). Далее создается проект приложения (File → New → Target → Application) и, как минимум, одно окно для взаимодействия с пользователем (File → New

→ PObject → Window). Клиентское приложение будет работать с БД, размещенной на сервере MS SQL Server. В Builder БД выступает в качестве источника данных, подключение к которому осуществляется через интерфейс ODBC [4]. Для создания нового источника данных на панели инструментов нужно кликнуть пиктограмму Database Profiles, и в открывшемся окне выбрать из списка ODB ODBC → Utilities → ODBC Administrator. В «Администраторе источников данных ODBC» следует нажать «Добавить», а в списке драйверов выбрать SQL Native Client. После выбора имени источника и подключаемого сервера БД нажимают «Далее» и в окне «Создание источника данных для SQL-сервера» выбирают имя требуемой базы.

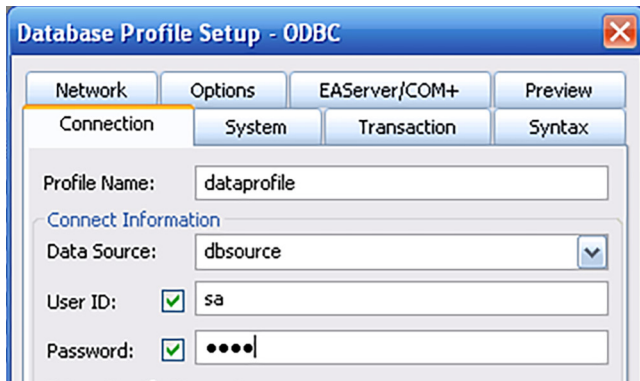


Рисунок 1. Создание профиля базы данных

Заметим, что данное имя базы необходимо использовать по умолчанию. После завершения настроек Builder предложит проверить подключение к источнику данных, и если программист сделал всё верно, то система сообщит об успешном завершении теста. На следующем шаге нужно вернуться в «Database Profiles» и с помощью кнопки «New» создать новый профиль. В диалоговом окне ввести новое имя профиля, User ID и Password, которые используются при подключении к БД в SQL Server (рис. 1). Созданный профиль отобразится среди остальных профилей ODB ODBC, и после выполнения подключения кнопкой «Connect» на его пиктограмме появится соответствующий значок (рис. 2).

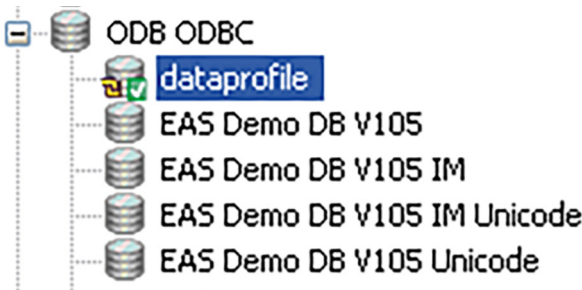


Рисунок 2. Подключенный профиль

После подключения к БД можно переходить к разработке экранных форм интерфейса пользователей. Входная экранная форма пользователей информационной системы должна предусматривать авторизацию клиентов и меню для выбора «режимов работы» (рис. 3). В окне авторизации клиент вводит логин, пароль и выбирает режим работы. Если введенные данные верны, то приложение открывает одно из диалоговых окон для дальнейшей работы. Иначе пользователю сообщается об ошибке и предлагается пройти авторизацию ещё раз. В пределах создаваемого окна нужно разместить различные элементы управления, такие как надписи (label), поля ввода (single line edit control), командные кнопки (command button), пере-

ключатели (radio button) и другие. Любые действия пользователя влекут за собой изменения состояния окна или элементов управления и называются событиями. К примеру, событием является нажатие кнопки (clicked), ввод и изменение текста (modified), открытие/закрытие (open, close) окна. События включаются тогда, когда по отношению к объекту выполняются какие-то действия. Сценарий, написанный для события, определяет, какая работа будет выполнена в ответ на событие [2].

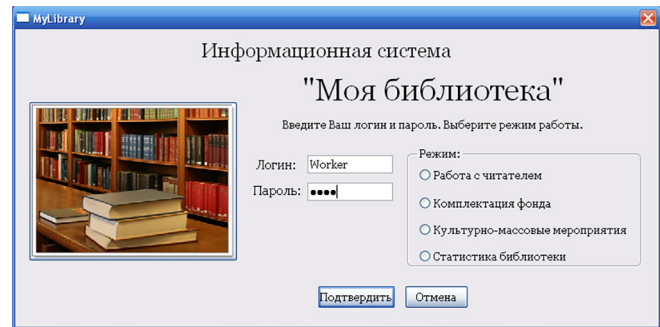


Рисунок 3. Окно авторизации

Для того, чтобы при запуске приложения происходило подключение к БД и открытие окна авторизации, нужно в обработчике события open прописать параметры стандартного объекта транзакций SQLCA (их можно получить из Dataprofiles → ODB ODBC dataprofile → Edit → Preview), и с помощью этого объекта и ключевых слов CONNECT USING организовать соединение и открыть окно с именем w_main (см. рис. 4). Для выхода из приложения нужно в обработчике события close применить аналогичную конструкцию DISCONNECT USING.

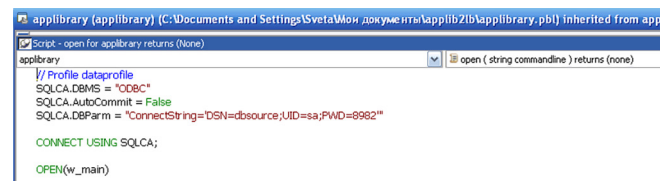


Рисунок 4. Скрипт события open

Проверка данных авторизации и выбор окна для дальнейшей работы выполняется скриптом, приведенным на рис. 5. Логин и пароль, введенные в соответствующие поля, сравниваются с использованием свойства text со значениями, прописанными в скрипте. Также, с помощью этого свойства пользователю сообщается об ошибке (текст «Введите логин...» изменяется на «Ошибка при вводе...»). Открытие окна происходит в операторе IF...THEN в зависимости от значения свойства checked (true – переключатель выбран/активен, false – не выбран/не активен).

```
cb_accept_enter
clicked ( ) returns long [pbm_bnclicked]
IF sle_log.text <> "Worker" OR sle_pswd.text <> "1111" THEN st_log_mode.text = "Ошибка при вводе данных! Попробуйте снова."
IF sle_log.text = "Worker" AND sle_pswd.text = "1111" THEN
  IF rb_reader.Checked = TRUE THEN
    open(w_reader)
  ELSEIF rb_books.Checked = TRUE THEN
    open(w_fond)
  ELSEIF rb_publics.Checked = TRUE THEN
    open(w_people)
  ELSEIF rb_stat.Checked = TRUE THEN
    open(w_statistic)
  END IF
END IF
```

Рисунок 5. Обработчик события clicked кнопки «Подтвердить»

Окно «Работа с читателем» предоставляет информацию о всех читателях библиотеки и по коду читателя позволяет найти и просмотреть его формуляр (рис. 6). Для создания интерфейса по технологии Data Window необходимо: 1) разместить окно данных (элемент управления); 2) создать объект datawindow; 3) выбрать для объекта представление и источник данных; 4) связать окно данных с объектом; 5) настроить возвращаемые аргументы; 6) с помощью скриптов организовать между элементами управления взаимодействие. Для создания окна данных нужно нажать на соответствующей пиктограмме «Create datawindow control» или воспользоваться вкладкой Insert в меню Builder и разместить элемент управления в окне, настроив его размеры и местоположение. Далее по команде File → New → PBOject → DataWindow выбрать представление Grid и источник данных SQL Select. В диалоговом окне Select Tables (см. рис. 7) следует выбрать одну или несколько таблиц, из которых будут отображаться данные. После этого необходимо выбрать столбцы, которые будут использоваться в запросе в качестве источника, и Builder определит код запроса в Select Painter (см. рис. 8). Для установления связи между объектом и элементом управления окна данных, следует выделить элемент управления и на боковой панели свойств, во вкладке General, выбрать имя сохраненного объекта dw_reader окна данных (см. рис. 9).

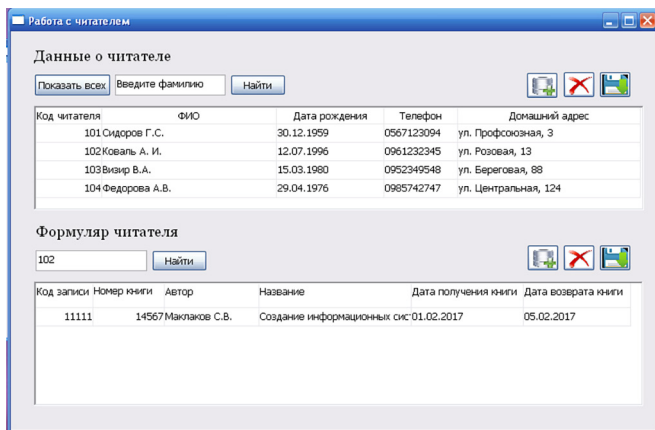


Рисунок 6. Диалоговое окно «Работа с читателем»

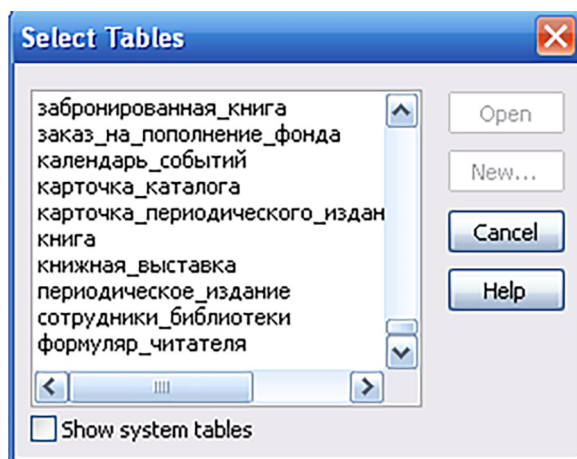


Рисунок 7. Выбор таблиц из БД

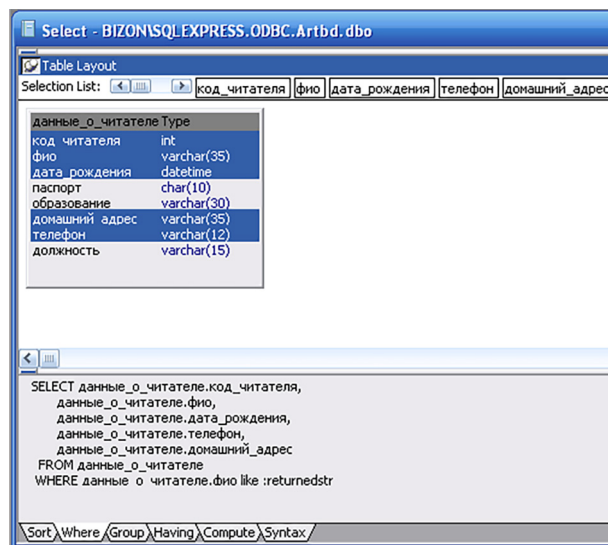


Рисунок 8. Создание SQL-запроса в PowerBuilder

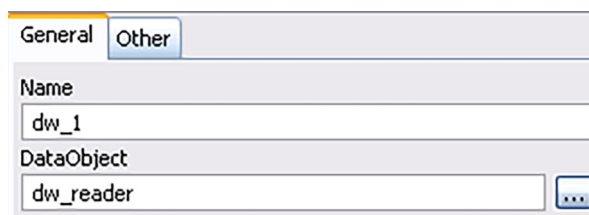


Рисунок 9. Связывание окна данных и объекта DataWindow

Для отображения и управления информацией из БД в окне данных также определим Retrieval Arguments (значения аргументов для получения методом Retrieve()). Вернемся в Select Painter и по команде Design → Retrieval Arguments выберем имя и тип данных для возвращаемого аргумента. На вкладке Where установим значения column, operator и value так, как показано на рис. 10. Результатом этого действия будет добавление секции WHERE в сгенерированный запрос. (см. рис. 11). На примере окна «Работа с читателем» (рис. 9) рассмотрим один из возможных способов организации интерфейса. С помощью окон данных, кнопок и методов DataWindow реализуем основные действия, доступные пользователю.

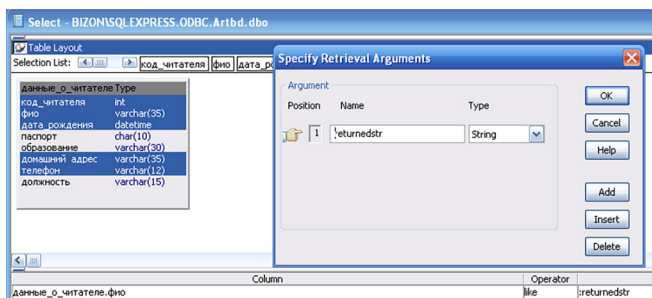


Рисунок 10. Настройка возвращаемых значений

```
SELECT данные_о_читателе.код_читателя,
данные_о_читателе.фio,
данные_о_читателе.дата_рождения,
данные_о_читателе.телефон,
данные_о_читателе.домашний_адрес
FROM данные_о_читателе
WHERE данные_о_читателе.фio like :returnedstr
```

Рисунок 11. Сгенерированный SQL запрос

В обработчике события open для элементов управления окнами данных dw_1 и dw_2, используя метод SetTransObject() (рис. 12), назначим объект SQLCA для организации взаимодействия БД с окнами данных [5].

Для просмотра данных создадим командную кнопку «Показать всех» и в clicked вызовем метод Retrieve(«%»), который будет возвращать все строки из БД. Аналогичный скрипт (см. рис. 13), но уже с учетом значений из поля ввода, используется для поиска (кнопка «Найти»).

```
cb_search
dw_1.Retrieve("%%");
dw_1.Retrieve(sle_1.text);
```

Рисунок 12. Использование метода SetTransObject()

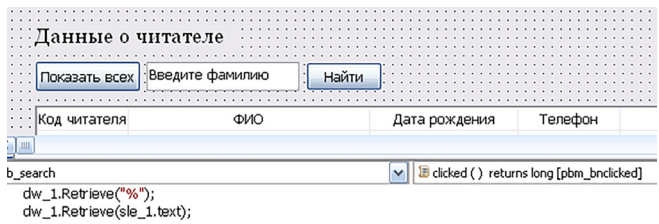


Рисунок 13. Скрипт для кнопки «Найти»

Так как окно dw_1 (и объект dw_reader) представляет данные из одной таблицы, то оно является редактируемым по умолчанию и изменение/добавление строк осуществляется непосредственно в окне данных. При этом дополнительные формы ввода или редактирования не создаются (dw_2 включает столбцы из таблиц «Книга» и «Формуляр читателя», поэтому для него необходимо разработать вспомогательные окна данных). Для сохранения внесенных изменений в БД используется метод Update() (рис. 14). Если этот метод сработал успешно (вернул «1»), то БД обновляется, иначе выполняется откат. В обработчиках clicked кнопок-пиктограмм для вставки и удаления строк вызываются методы InsertRow() и DeleteRow() [5]. Диалоговые окна «Комплектация фонда», «Культурно-массовые мероприятия» и «Статистика библиотеки» разрабатываются аналогичным образом.

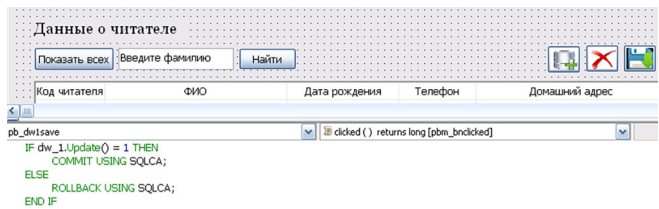


Рисунок 14. Скрипт кнопки-пиктограммы «Сохранить»

Созданное приложение нужно передать заказчику для опытной эксплуатации. По её результатам принимается решение о необходимости доработки программы.

Литература

1. Касчиато Ч. PowerBuilder. Руководство разработчика: Пер. с англ. — М.: Малип, 1997. — 416 с., ил.
2. PowerBuilder 6.0. Энциклопедия пользователя: Пер. с англ./Симон Галлагер и Симон Херберт — К.: Издательство «ДиаСофт», 1998. — 816 с., ил.
3. Саликов В.А., Сторчак С.А. Создание базы данных сельской библиотеки по технологии Allfusion Process+Data Modeler — International scientific journal «INTERNAUKA», № 2(24), 2017, том 1, pp. 160–164.
4. Sybase DataWindow Programmer's Guide PowerBuilder 10.5 [Электронный ресурс]. — Режим доступа: <https://www.scribd.com/document/57138604/dwprgug>
5. Sybase DataWindow Reference PowerBuilder 10.5 [Электронный ресурс]. — Режим доступа: http://infocenter-archive.sybase.com/help/topic/com.sybase.help.pb_10.5.dwref/pdf/dwref.pdf.