



OpenEdge Architect

Учебное пособие на русском языке

©Башкатов В.Г. 2008 г.



PROGRESS
SOFTWARE

Содержание

ВВЕДЕНИЕ В OPENEDGE ARCHITECT	3
РАБОЧИЕ ОБЛАСТИ, ПРОЕКТЫ И РЕСУРСЫ	3
СТАРТ И СОЗДАНИЕ ВАШЕЙ ПЕРВОЙ РАБОЧЕЙ ОБЛАСТИ	3
СОЗДАНИЕ ПРОЕКТА	5
СОЗДАНИЕ ВАШЕГО ПЕРВОГО РЕСУРСА	9
ВЫПОЛНЕНИЕ КОДА В РЕЖИМЕ GUI	12
ВЫПОЛНЕНИЕ КОДА В РЕЖИМЕ CHUI	13
ИМПОРТ ИСХОДНОГО КОДА В ПРОЕКТ	15
OPENEDGE ABL И OPENEDGE EDITOR.....	19
ПОДКЛЮЧЕНИЕ БАЗЫ ДАННЫХ К ПРОЕКТУ	19
СОЕДИНЕНИЕ ABL APPLICATION И БАЗЫ ДАННЫХ	19
ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ КОДА OPENEDGE EDITOR	22
<i>Исследование особенностей редактора OpenEdge Editor.....</i>	<i>22</i>
OPENEDGE DATABASE И DATABASE NAVIGATOR.....	39
СОЗДАНИЕ НОВОЙ БАЗЫ ДАННЫХ.....	39
ИСПОЛЬЗОВАНИЕ DATABASE NAVIGATOR ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ	39
НАСТРОЙКА И ЗАПУСК СЕРВЕРА БАЗЫ ДАННЫХ	42
КОНФИГУРИРОВАНИЕ СЕРВЕРА БАЗЫ ДАННЫХ	42
ЗАПУСК СЕРВЕРА БАЗЫ ДАННЫХ	48
СОЕДИНЕНИЕ DATABASE NAVIGATOR С БАЗОЙ ДАННЫХ.....	50
СОЕДИНЕНИЕ DATABASE NAVIGATOR С СЕРВЕРОМ DATABASE SERVER.....	50
ИССЛЕДОВАНИЕ И МОДИФИКАЦИЯ СХЕМЫ БАЗЫ ДАННЫХ.....	54
ИССЛЕДОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ.....	54
МОДИФИКАЦИЯ СХЕМЫ БАЗЫ ДАННЫХ	56
ПРОГРАММА ОТЛАДКИ: OPENEDGE DEBUGGER.....	61
НАСТРОЙКА СРЕДЫ ОТЛАДКИ	61
РАЗБЛОКИРОВАНИЕ ФУНКЦИИ ОТЛАДКИ.....	61
СОХРАНЕНИЕ И ГЕНЕРАЦИЯ R-КОДА.....	62
ОТЛАДКА ABL ПРИЛОЖЕНИЯ	65
ИСПОЛЬЗОВАНИЕ ОТЛАДЧИКА DEBUGGER	65
ИНСТРУМЕНТЫ ДЛЯ БИЗНЕС-ЛОГИКИ	74
СОЗДАНИЕ КОМПОНЕНТНОЙ ДИАГРАММЫ.....	74
СОЗДАНИЕ ФАЙЛА ДИАГРАММЫ И PRODATASET КОМПОНЕНТА	74
СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ С ИНФОРМАЦИЕЙ О СТРУКТУРЕ СХЕМЫ БАЗЫ ДАННЫХ	77
ДОБАВЛЕНИЕ СВЯЗЕЙ ОТНОШЕНИЙ	79
ЗАБЛАГОВРЕМЕННЫЙ ПРИКЛАДНОЙ КОД	87
ИНТЕГРАЦИЯ СОЗДАННОГО КОДА В ПРИЛОЖЕНИЕ.....	90
ИСПОЛЬЗОВАНИЕ СОЗДАННОГО КОДА В ПРИЛОЖЕНИИ	90

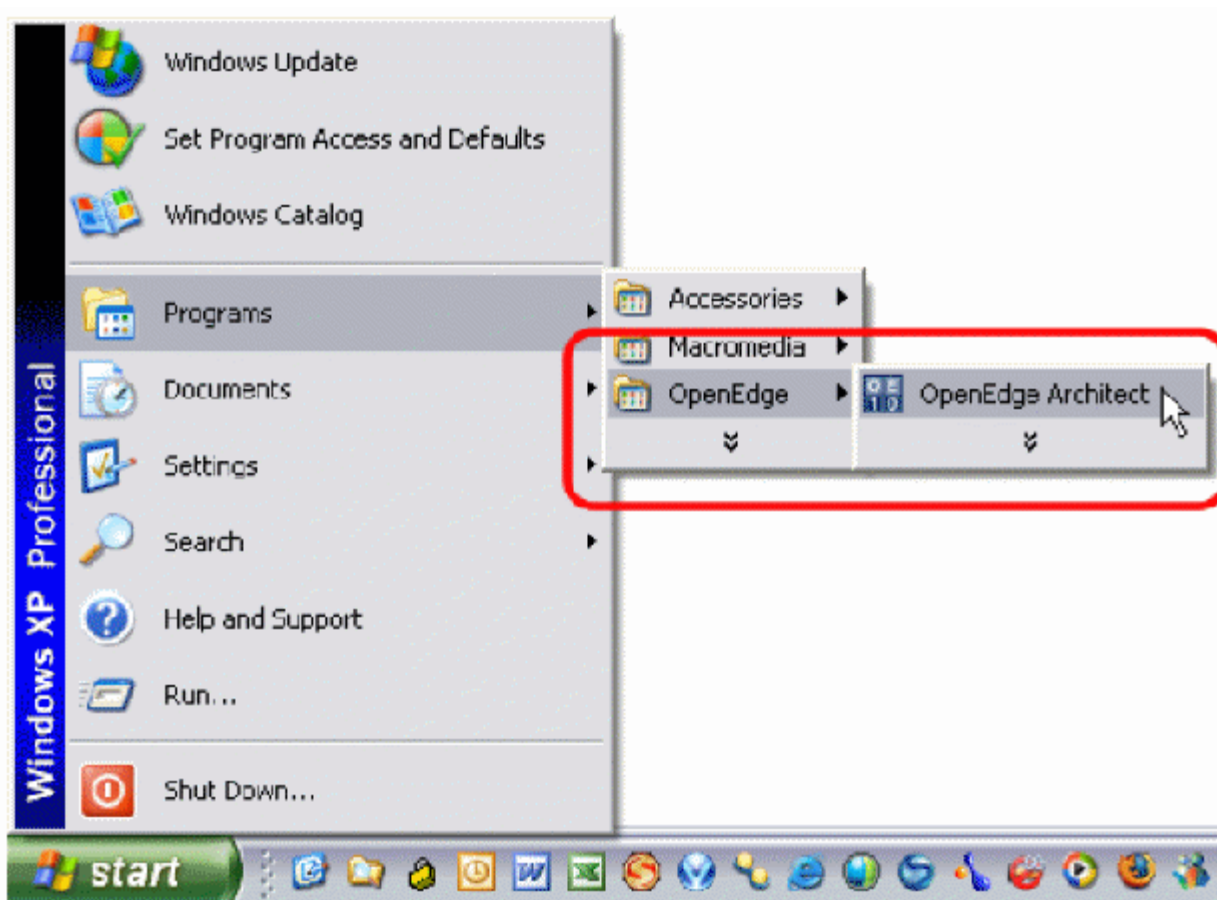
Введение в OpenEdge Architect

Рабочие области, проекты и ресурсы

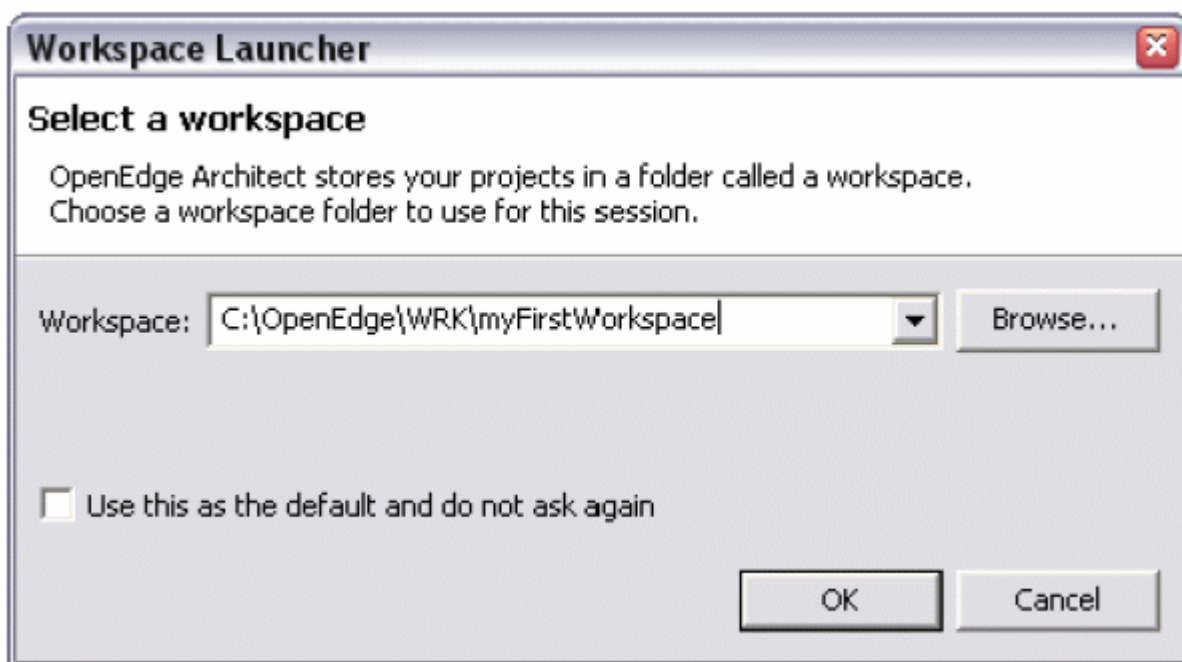
Старт и создание Вашей первой рабочей области

При первом запуске OpenEdge Architect Вам будет предложено указать рабочую область. После чего будет открыто окно приветствия, содержащее ссылки на ресурсы и дополнительную информацию относительно среды разработки OpenEdge. В этом упражнении мы запустим Architect и создадим первую рабочую область.

1. Выберите иконку OpenEdge Architect из меню “Start”

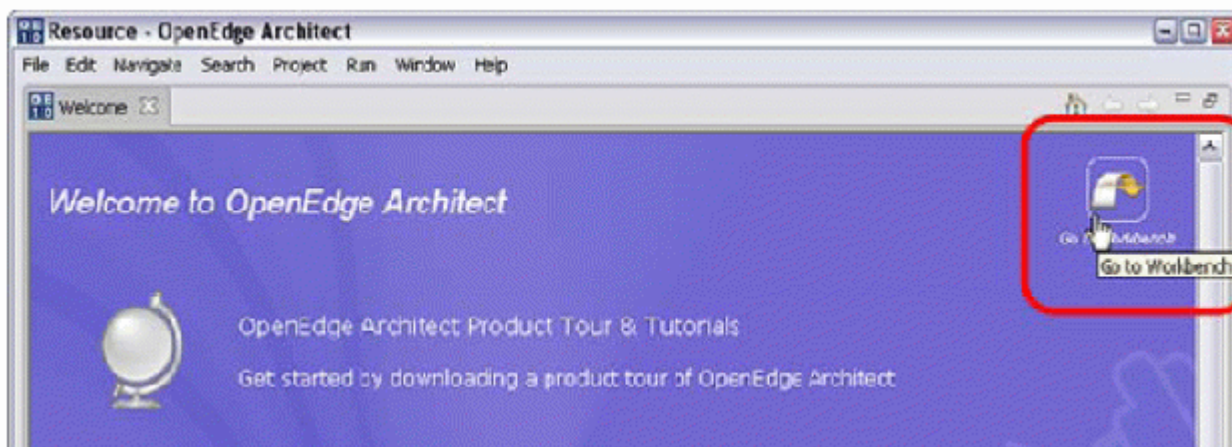


2. После запуска, Вам будет предложено создать рабочую область. Введите “C:\OpenEdge\WRK\myFirstWorkspace” вместо предложенной рабочей области по умолчанию. Мы будем использовать эту директорию в качестве рабочей области наших проектов.

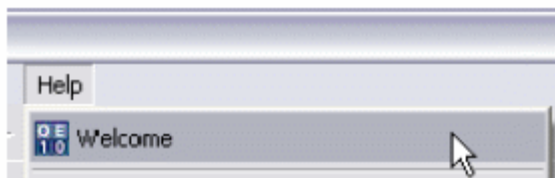


Примечание: Когда Вы создаёте рабочую область, помните, что Architect чувствителен к регистру. Т.е. если будет введено нечто вроде “C:\OpenEdge\wrk\myFirstWorkspace”, где каталог “wrk” должен быть на самом деле в верхнем регистре, Architect даст Вам это сделать и продолжит работу, но у Вас возникнут сложности в будущем с рабочей областью и проектом связанным с ней, поскольку Architect не сможет правильно найти этот каталог.

3. Далее будет открыто окно приветствия, содержащее ссылки на дополнительные материалы, имеющие отношение к OpenEdge Architect и другим OpenEdge продуктам. Просто щелкните на иконке “Go to Workbench” в правом верхнем углу экрана.



Вы можете в любое время вернуться в окно приветствия, выбрав в меню Help -> Welcome.



Либо выбрать иконку “Return to Welcome” (если окно приветствия не было закрыто), расположенную в левом нижнем углу экрана.

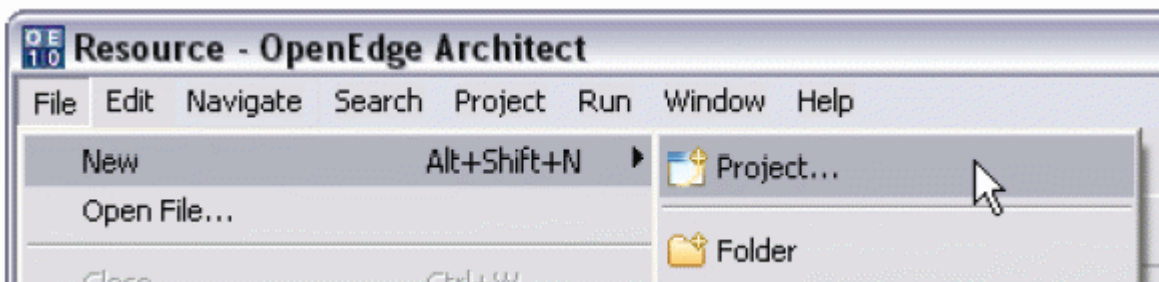


4. Теперь Вы перешли к основному экрану Architect – Main Screen. Поздравляем, Вы успешно запустили Architect и создали свою первую рабочую область.

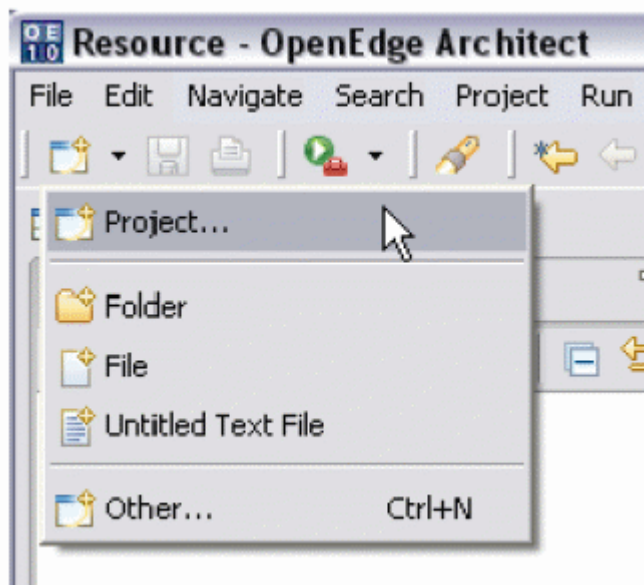
Создание проекта

Создания проекта – простой процесс. Процесс создания происходит с помощью специального окна, которое поможет настроить Ваш проект. Architect автоматически настроит структуру проекта и создаст директории для его хранения.

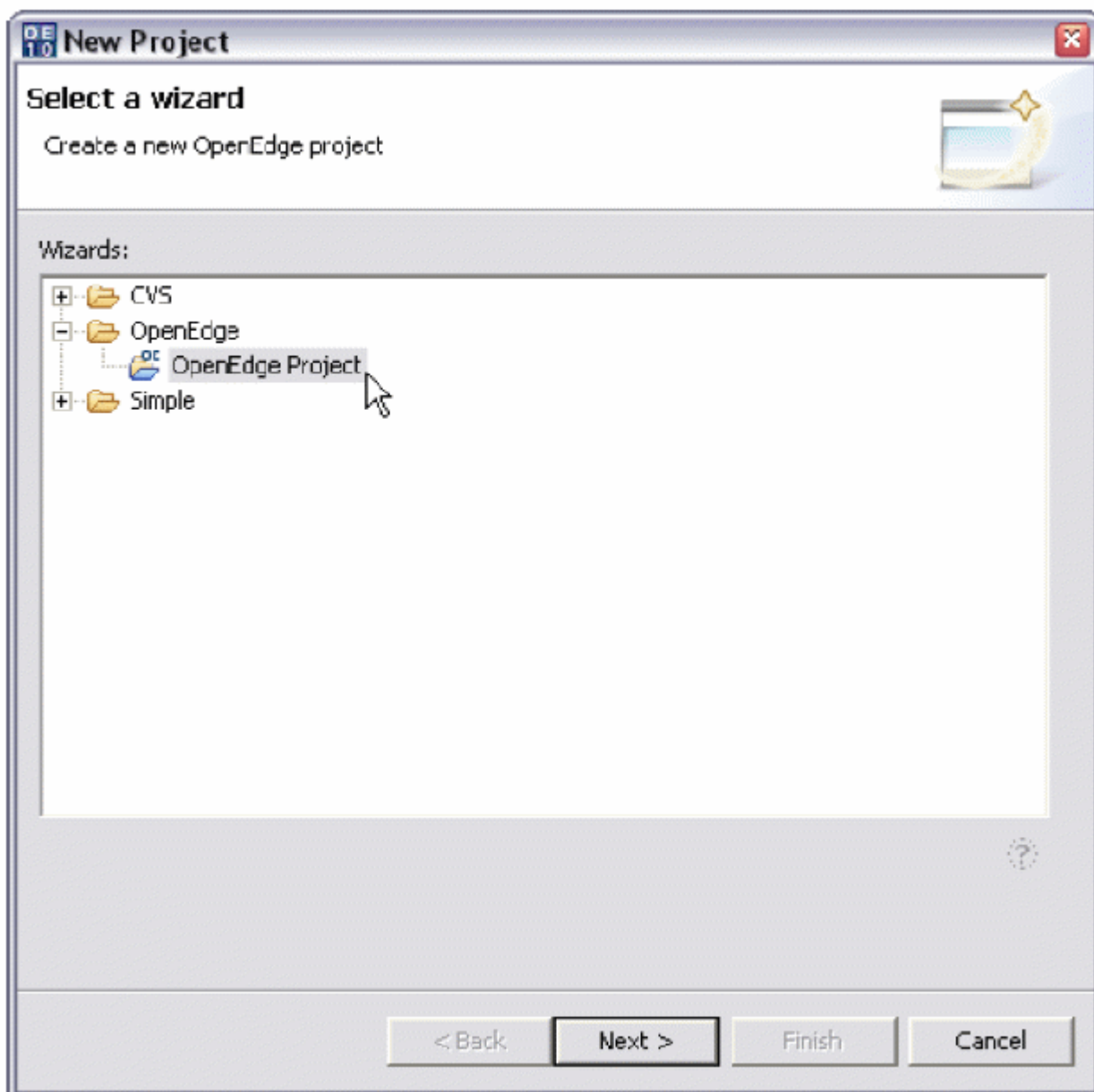
1. Для создания нового проекта выберите в меню File -> New -> Project.



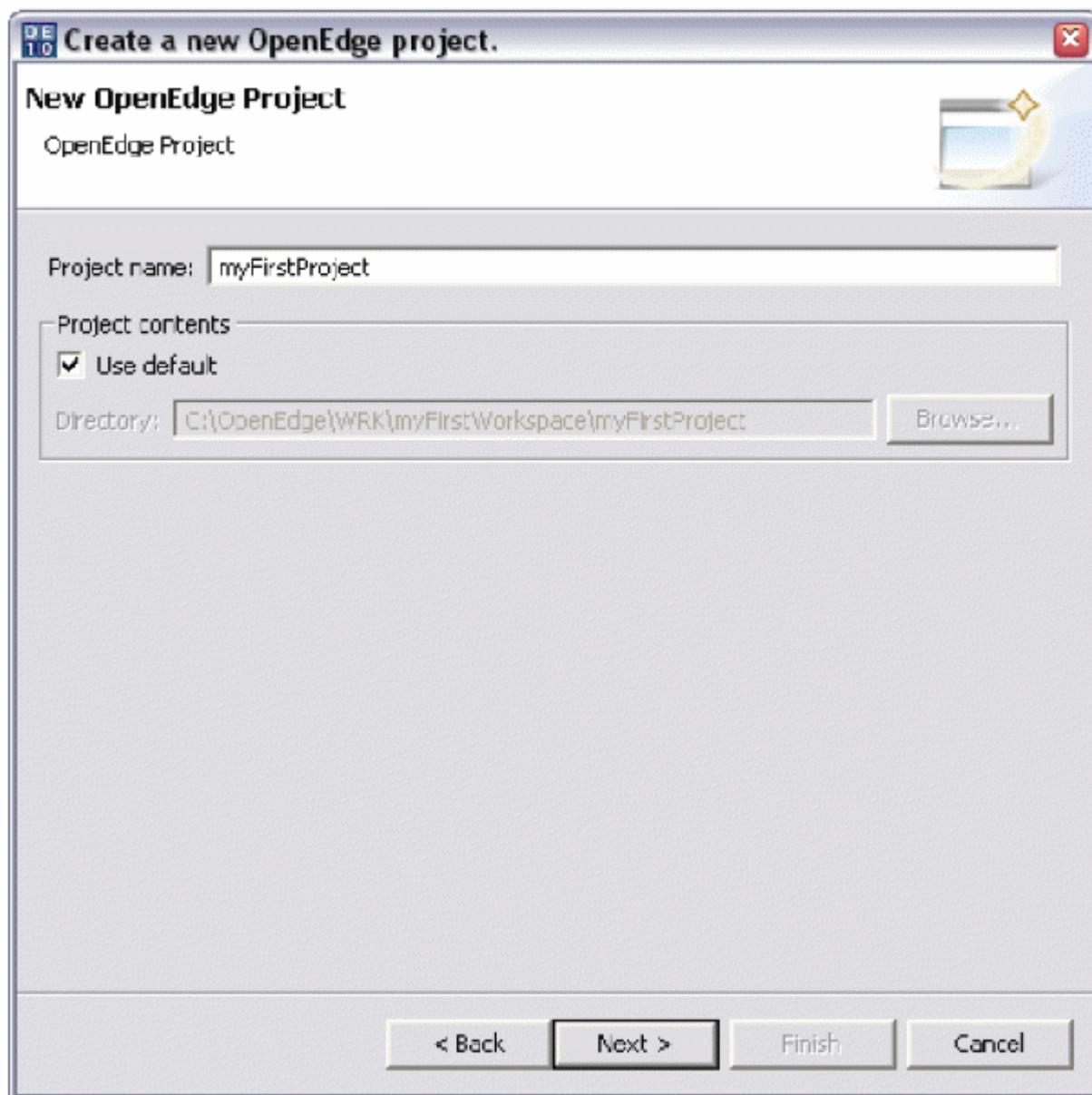
Или щелкните стрелку около кнопки “New” в меню и выберите опцию “Project”.



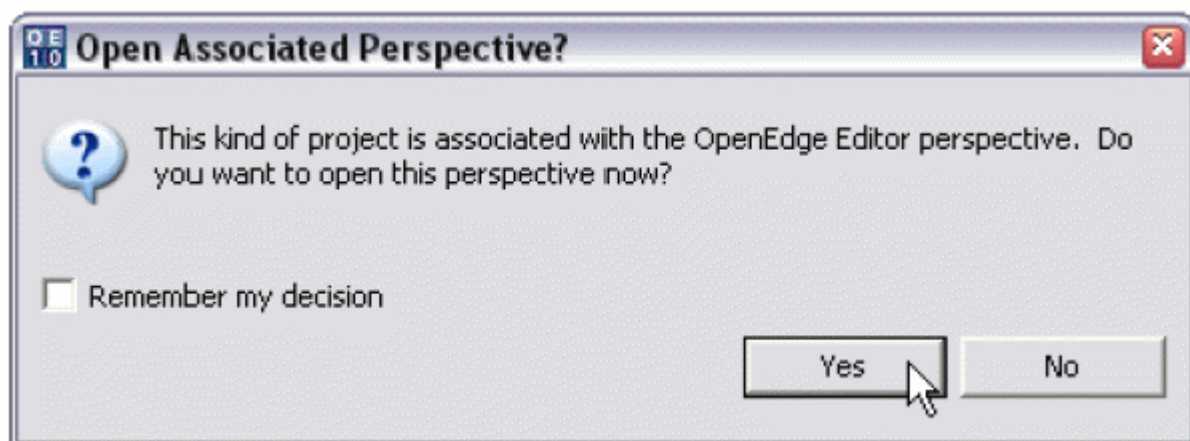
2. После чего будет отображен экран, в котором можно выбрать тип создаваемого проекта. Выбор типа проекта определяет, какой тип Wizard`а будет запущен для завершения процесса. Для создания проекта раскройте папку OpenEdge и выделите "OpenEdge Project". Щелкните на кнопке "Next".



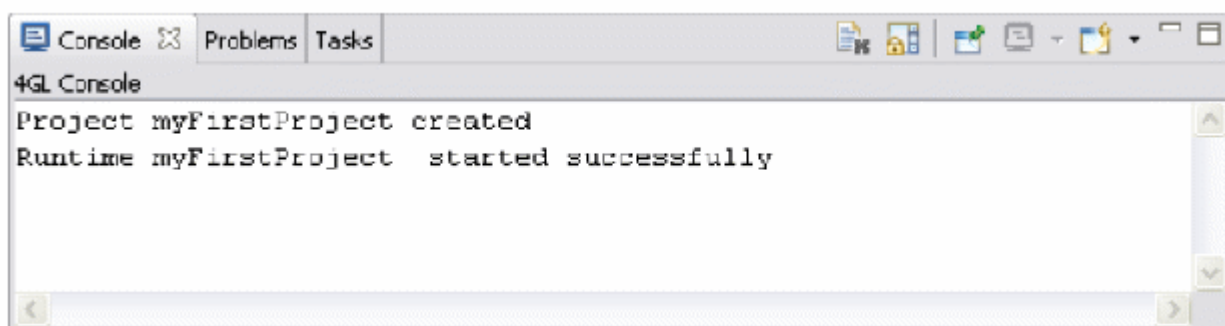
3. Следующий экран позволяет ввести имя проекта, которое необходимо ввести в поле “Project Name”. По умолчанию, Architect создает директорию проекта в текущей рабочей области, используя имя проекта введенное Вами. Если Вы хотите хранить файлы проекта в директории, отличной от предложенной по умолчанию, необходимо отменить опцию “Use Default” и ввести свой вариант. Для этого упражнения используйте директорию по умолчанию. Щелкните кнопку “Next”.



4. Architect начнет процесс создания нового проекта. Будет выведен экран с вопросом о необходимости связать проект с OpenEdge Editor Perspective. Выберите кнопку "Yes".



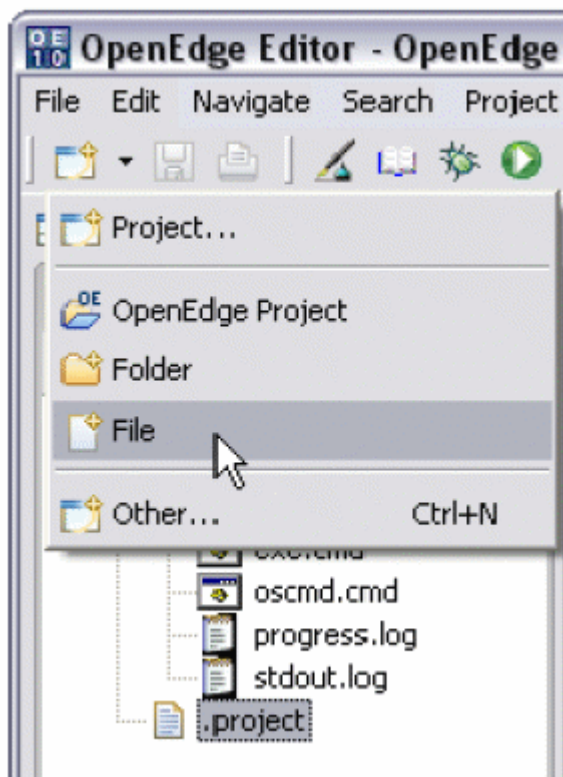
5. Теперь процесс создания нового проекта завершен. Сообщения о создании проекта и его статус будет отображен в Console View. Позже мы подробнее остановимся на деталях проекта, ну а пока необходимо рассмотреть управление Architect`ом и научиться понимать его сообщения.



Создание вашего первого ресурса

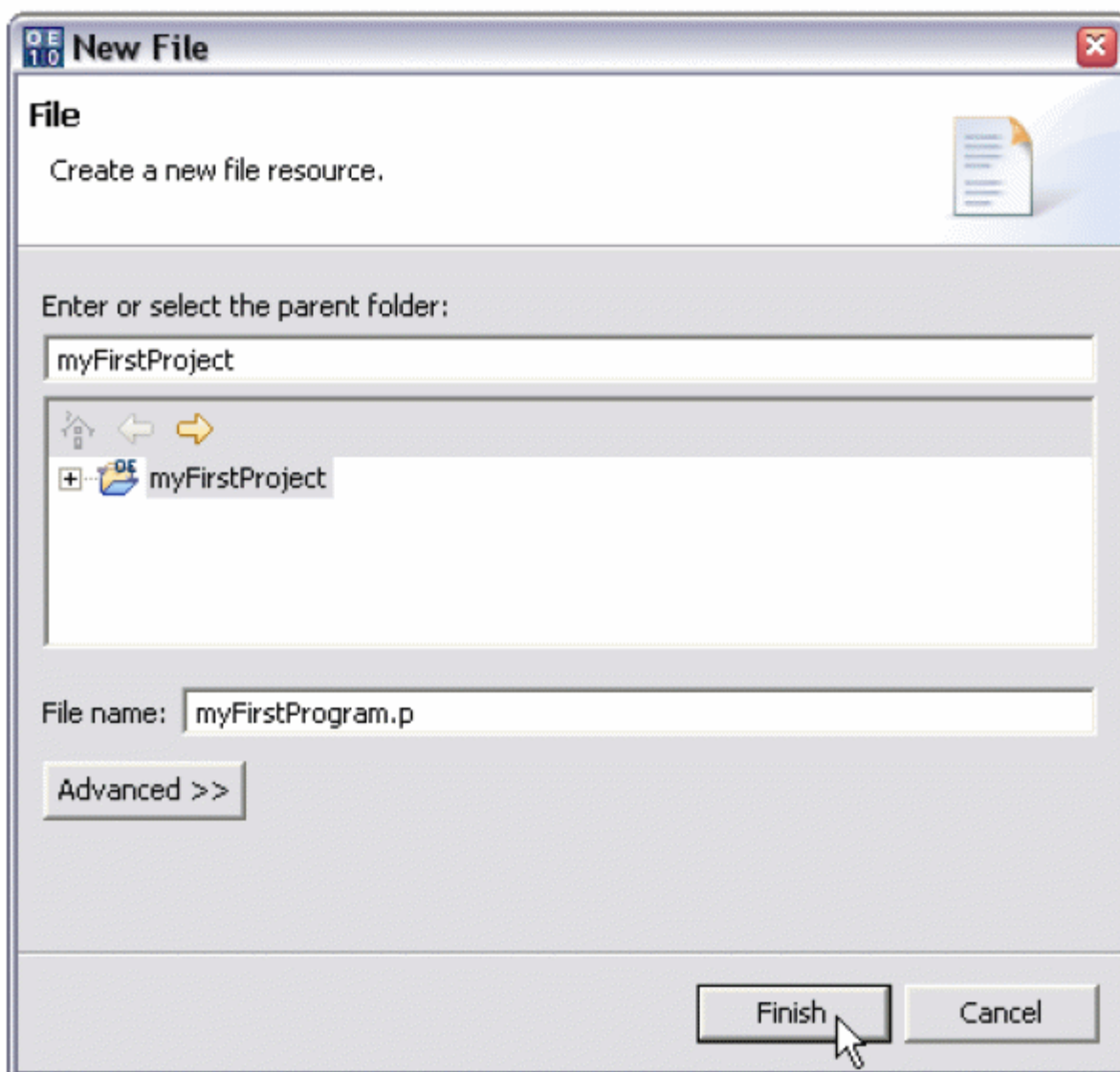
Исходный код ABL, содержащий синтаксис интерфейса пользователя и бизнес ABL, наиболее распространенный файловый ресурс, используемый в проекте. Этот тип автоматически связывается с OpenEdge Editor`ом. В этом упражнении Вы создадите Ваш первый файловый ресурс исходного кода ABL.

1. Создайте новый файл, щелкнув стрелку около кнопки “New” и выбрав опцию “File”.



Создать новый файл можно также из меню опций File -> New -> File.

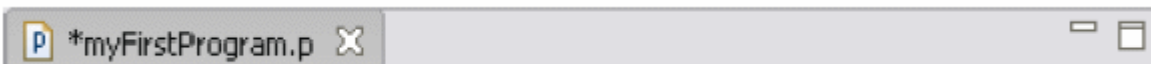
2. В диалоговом окне "New File" можно изменить директорию файла, заданную по умолчанию, а также дать имя новому файлу. Для данного примера, оставьте директорию по умолчанию, myFirstProject, и присвойте имя файлу – "myFirstProgramm.p". Щелкните на "Finish" для продолжения.



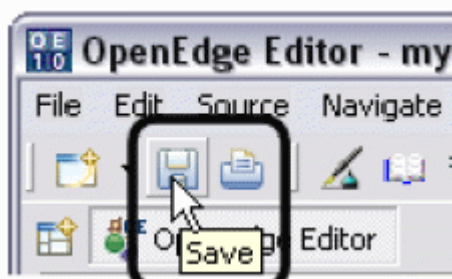
3. Теперь Architect создаст файл и откроет его с помощью OpenEdge Editor.
4. Введите следующий синтаксический код, используя OpenEdge Editor View

```
Project: myFirstProject Program: myFirstProgram.p  
DISPLAY "HELLO WORLD!".  
PAUSE.  
QUIT.
```

5. Когда Вы измените текст, перед названием файл в View Tab появится звездочка (*). Это означает, что после последнего сохранения файла, в нем произошли изменения.



Чтобы сохранить файл, выберите кнопку “Save”, либо используйте меню File -> Save

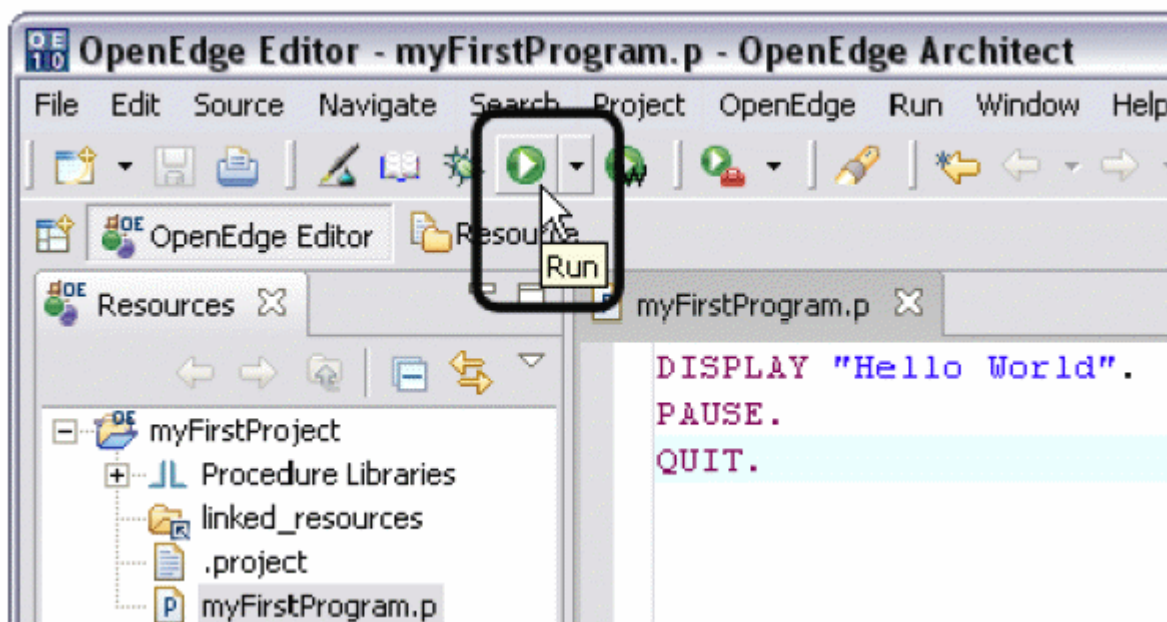


Это завершающий процесс для создания файла исходного кода.

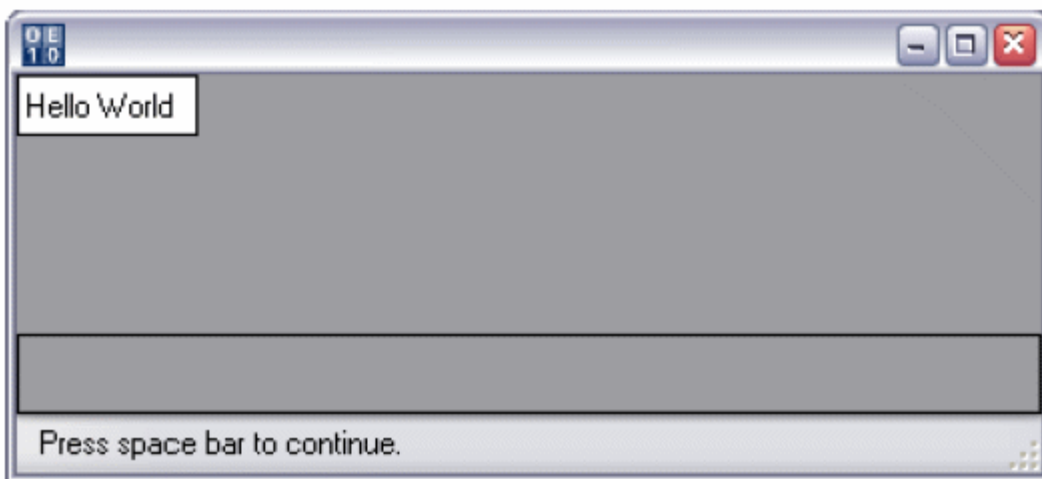
Выполнение кода в режиме GUI

Architect может создавать и тестировать код ABL в следующих средах – GUI, Character и Web . GUI режим - режим по умолчанию, используемый Architect. Следующим нашим шагом будет тестирование нашего кода в GUI режиме.

1. Выберите кнопку “Run” в Main-меню.



2. Вы увидите результат выполнения программы в GUI режиме в виде слова “Hello World” в верхнем углу. Нажмите пробел для завершения работы программы.



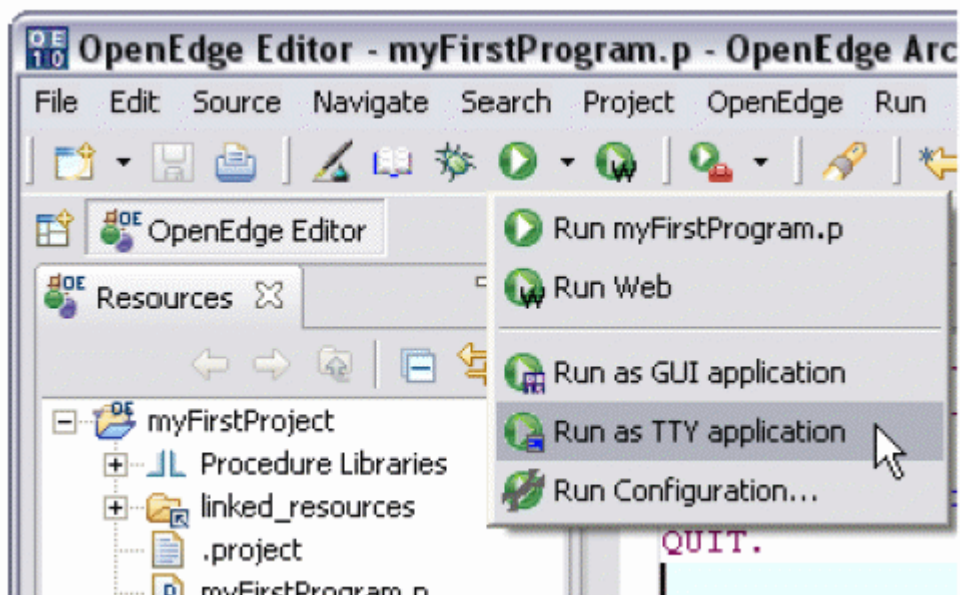
Выполнение кода в режиме CUI

Вы также можете запустить Вашу программу в Character – режиме.

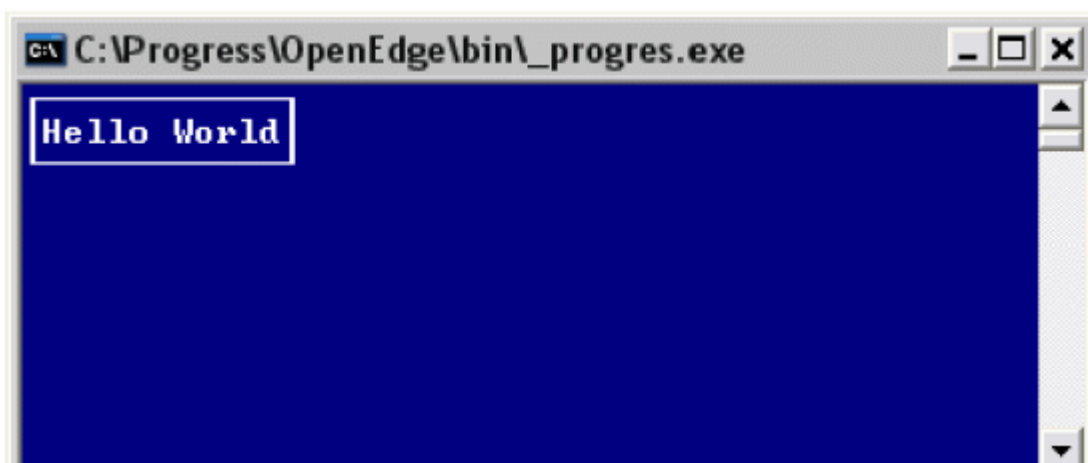
1. Выберите стрелку около кнопки “Run”



2. В раскрывшемся меню выберите “Run as TTY application”

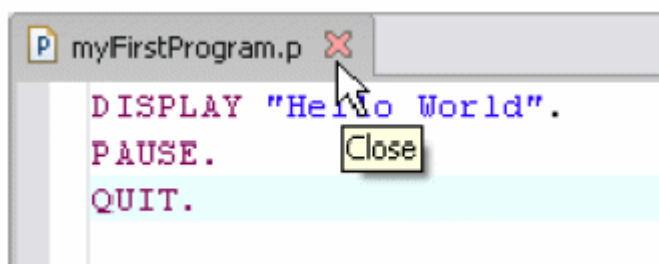


3. Символьный экран отобразит следующий результат.



Нажмите пробел для закрытия окна

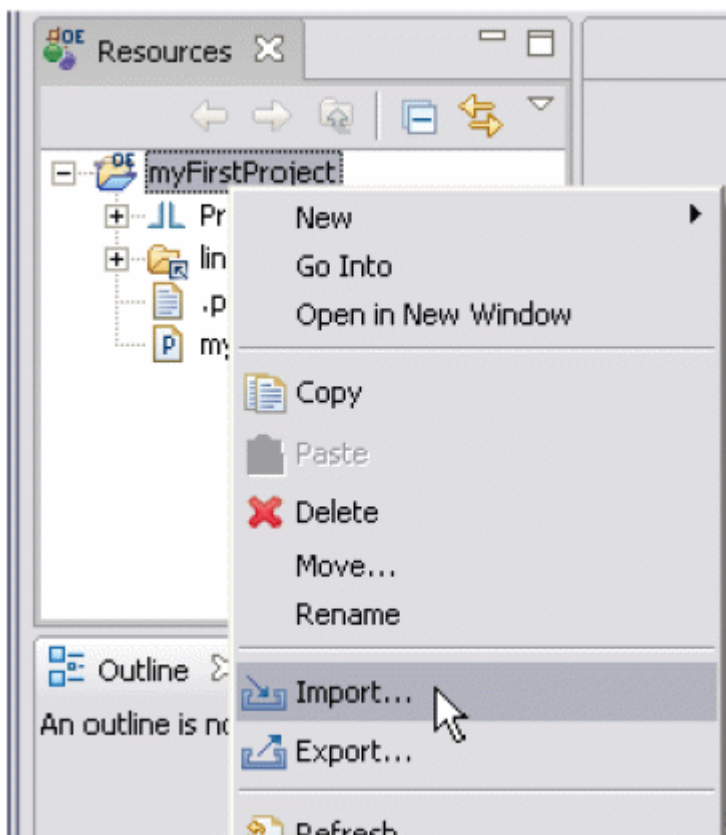
4. Закройте myFirstProgram.p файл.



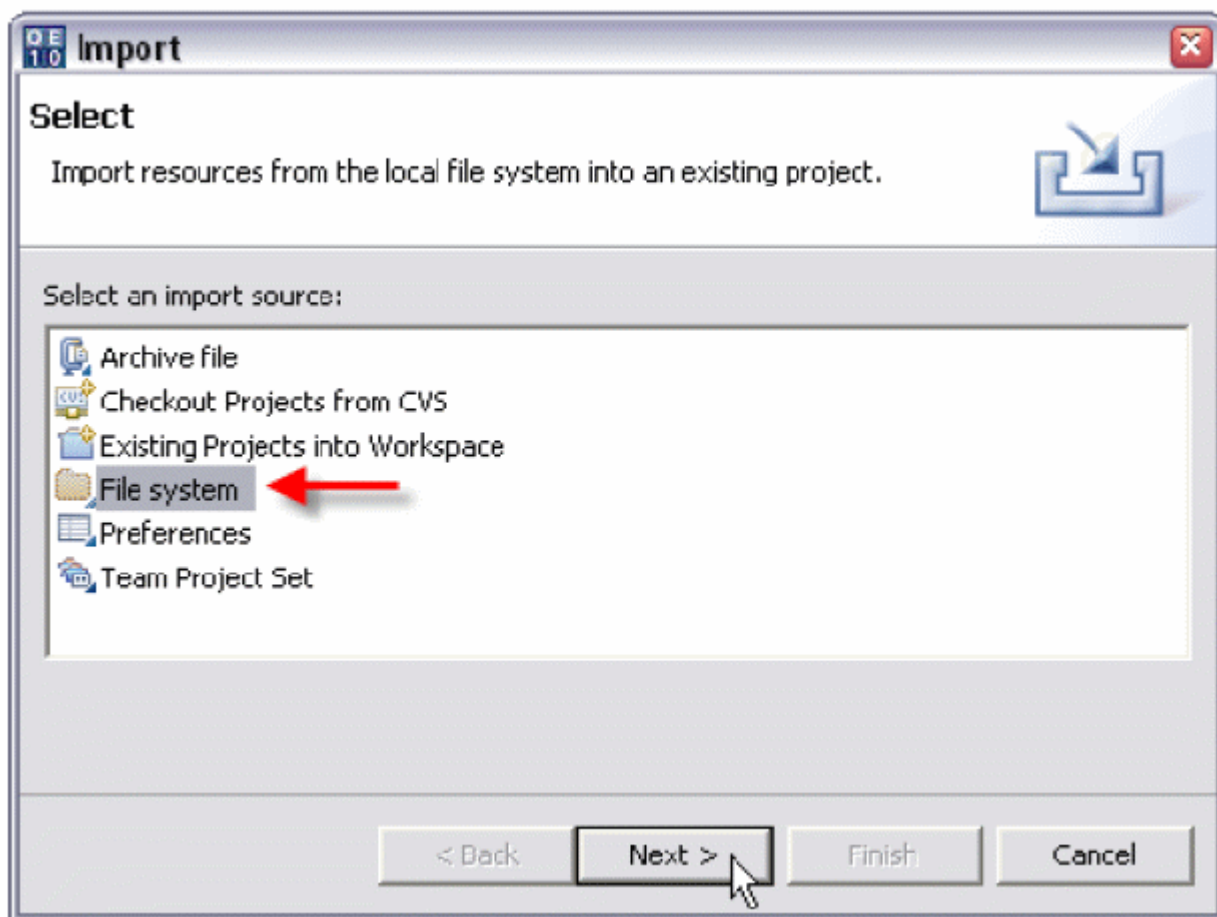
Импорт исходного кода в проект

Architect умеет импортировать код, созданный как другими проектами, так и код созданный в среде, отличной от среды Architect. Для того чтобы найти исходный код для нашего проекта, сделаем следующее:

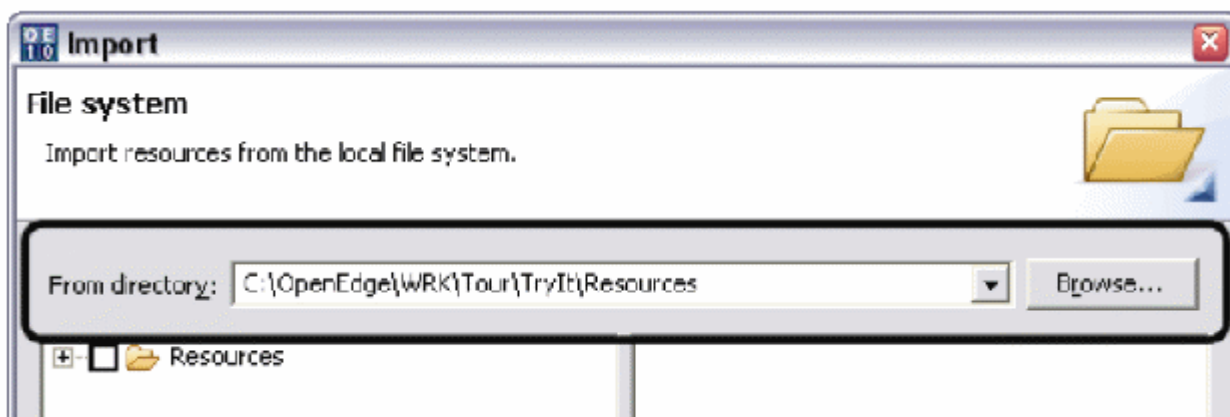
1. Щелкните правой кнопкой мыши на myFirstProject в Resource View и выберите опцию Import. Так же можно использовать опцию File -> Import в меню Main



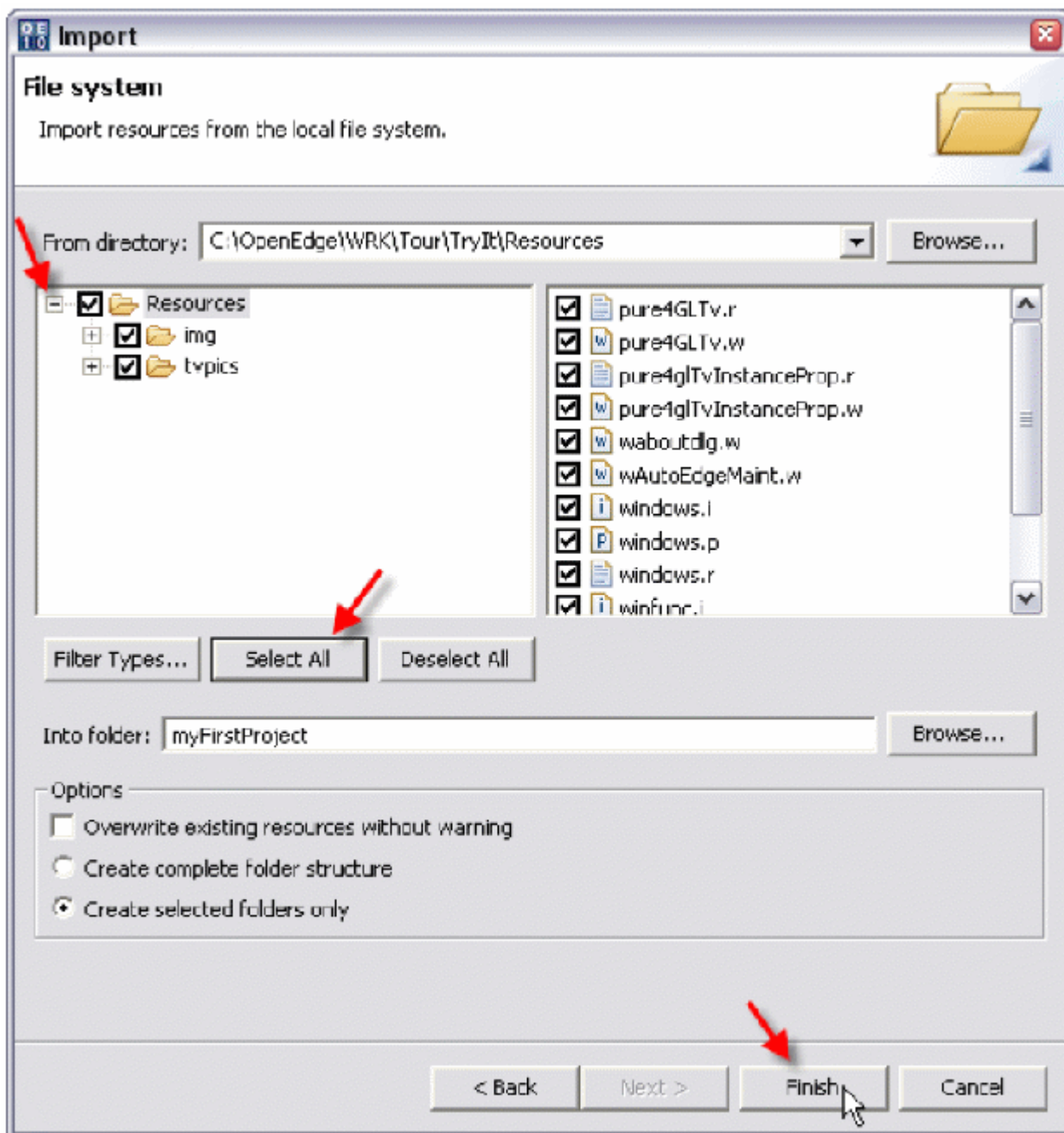
2. Запустится Wizard импорта. В первом окне для импорта выберите тип ресурса "File System". Щелкните кнопку "Next" для продолжения.



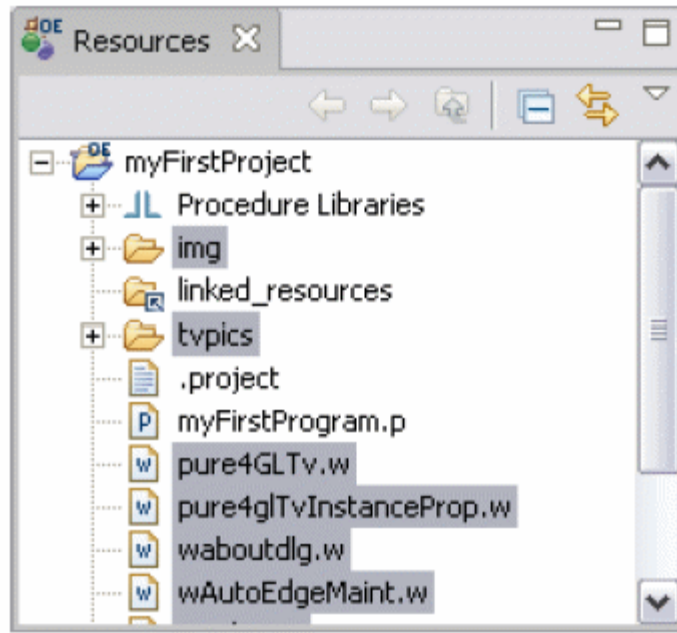
3. В поле “From Directory” введите директорию с исходным кодом. Вы также можете использовать кнопку “Browse” для поиска.



4. Разверните папку “Resources” и щелкните кнопку “Select All”. Все директории и файлы, находящиеся в папке “Resources” будут помечены для импорта в проект. Сама же директория “Resources” импортирована не будет. Щелкните кнопку “Finish”.



5. В окне Resources View теперь Вы можете увидеть импортированные файлы и директории. Раскройте одну из импортированных директорий, и вы увидите файлы, импортированные вместе с ней.



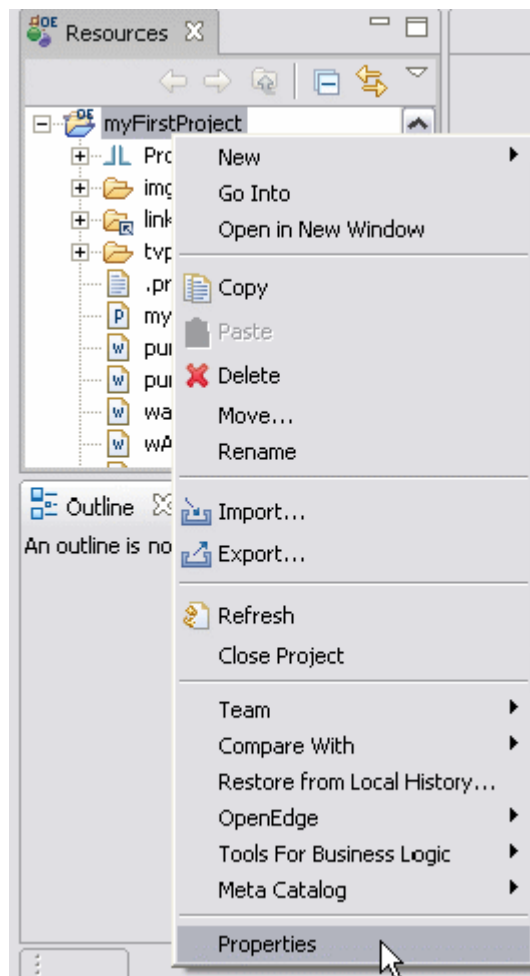
OpenEdge ABL и OpenEdge Editor

Подключение Базы Данных к проекту

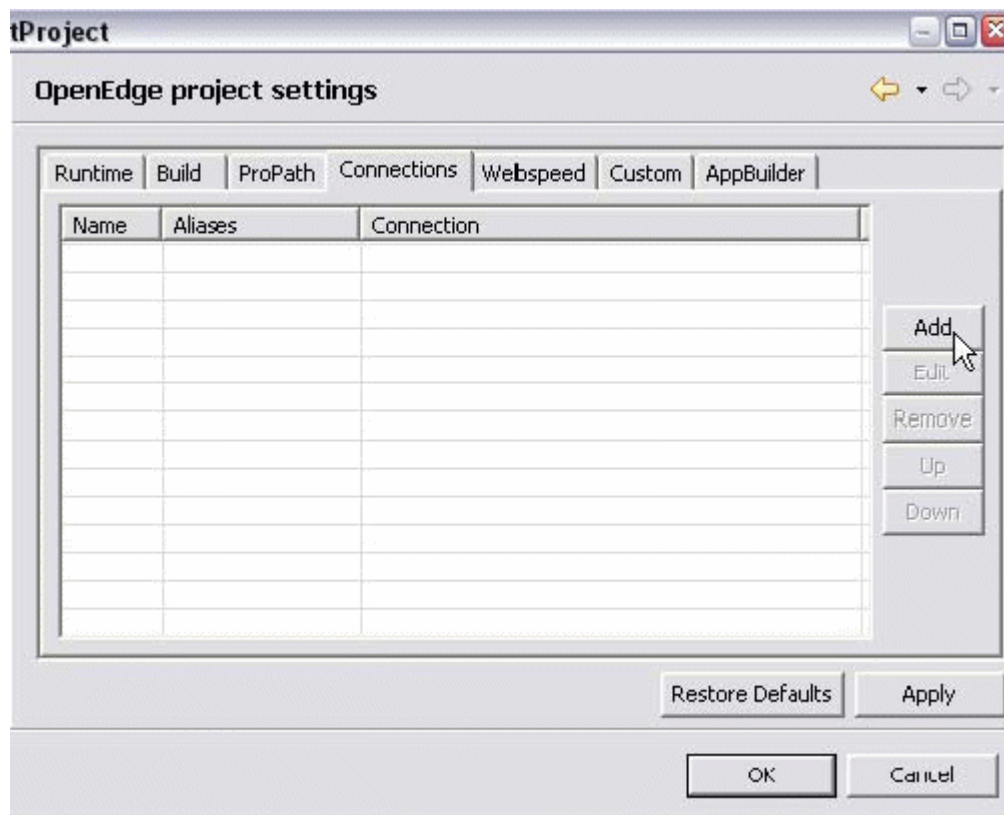
Соединение ABL Application и Базы Данных

Прежде, чем начать программировать ABL-код для доступа к базе данных, необходимо сначала определить набор свойств, позволяющих проекту подсоединиться к ней. Для создания соединения проекта myFirstProject с базой autoEdge выполним следующее.

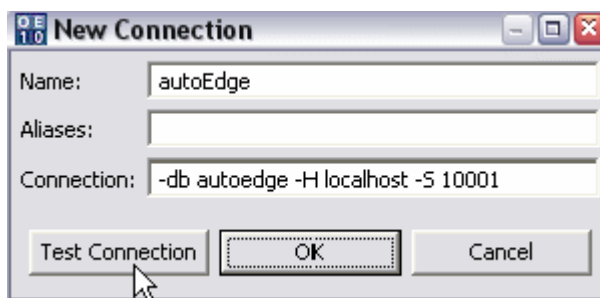
1. Убедитесь, что открыт редактор OpenEdge Editor Perspective.
2. Правой кнопкой мыши кликните на пункте myFirstProject в обозревателе Resources и выберите опцию Properties.



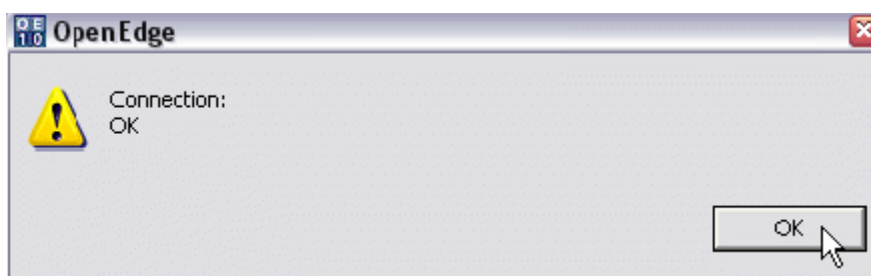
3. С левой стороны окна Properties выберите пункт OpenEdge, вкладку Connection. Нажмите кнопку Add.



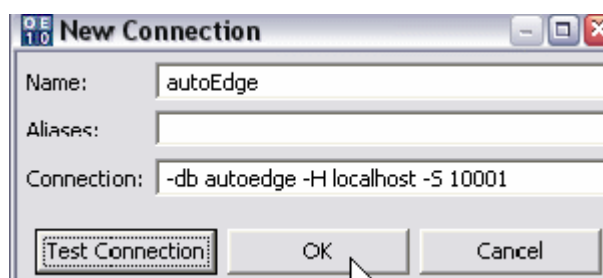
4. В окне New Connection в поле Name наберите “autoEdge”, в поле Connection – “-db autoedge -H localhost -S 10001”, нажмите кнопку Test Connection.



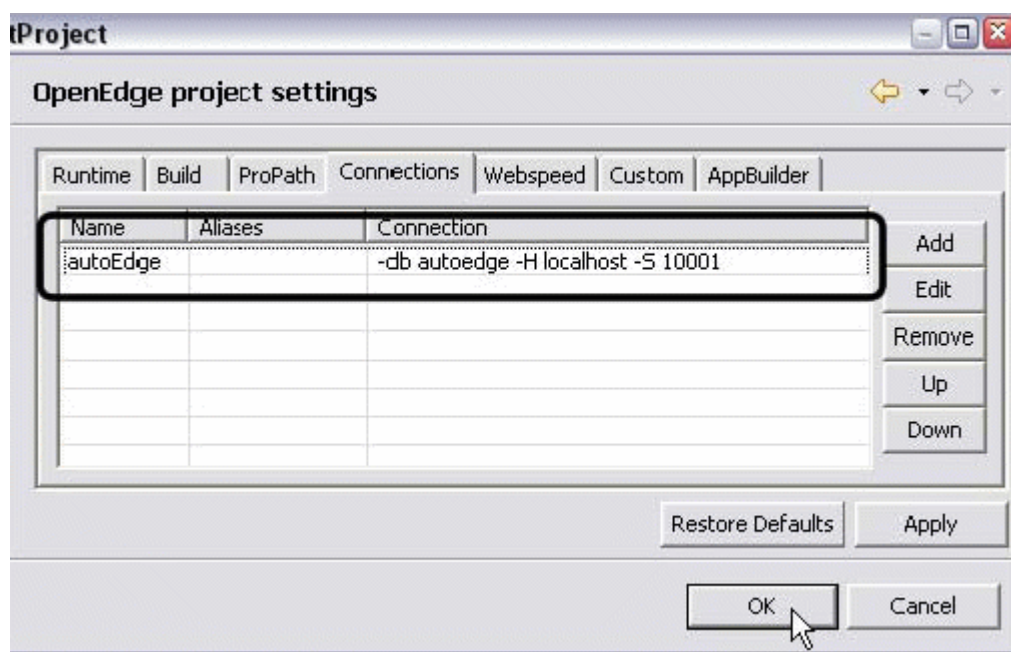
5. В случае успешного соединения должно появиться диалоговое окно Connection: ОК. Нажмите кнопку ОК.



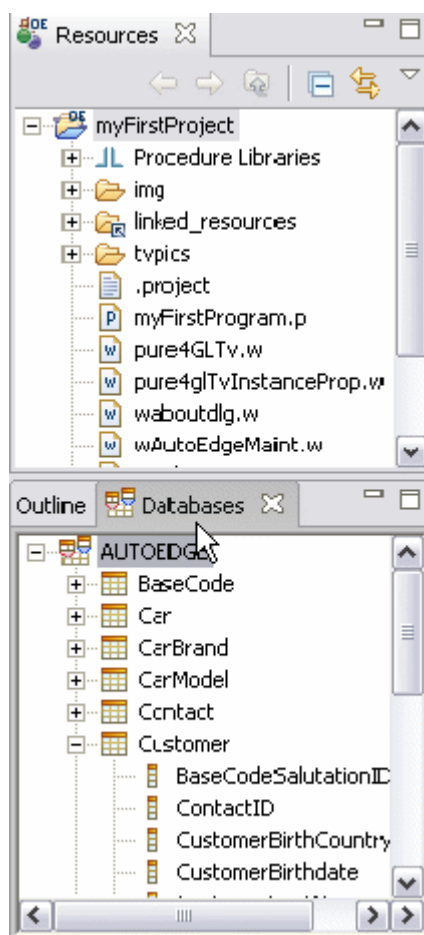
6. Теперь в окне New Connection для завершения создания соединения кликните OK.



7. В окне Properties появилось созданное новое соединение, отображившееся в браузере. Нажмите OK.



8. Выберите вкладку Databases следующую за вкладкой Outline. Данный обозреватель в режиме чтения отображает схему базы данных. Разверните пункты AUTOEDGE -> Customer. Как вы и убедились, эту вкладку можно использовать для быстрого просмотра таблиц и их полей, определенных в базе данных.



Использование вспомогательных функций кода OpenEdge Editor

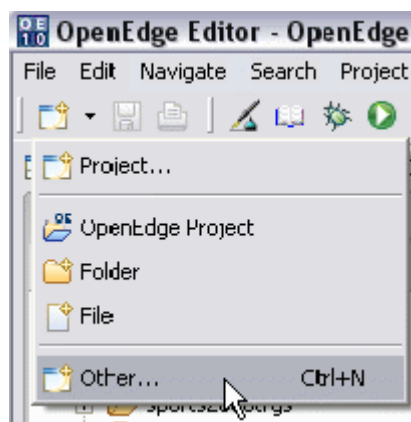
Исследование особенностей редактора OpenEdge Editor

В данном разделе вы научитесь составлять ABL код для доступа и вывода на экран информации, содержащейся в базе данных, на примере базы autoEdge, используя для редактирования вспомогательные возможности OpenEdge Editor Architect. Код будет находить и выводить на экран список имен покупателей и их дней рождений из базы autoEdge.

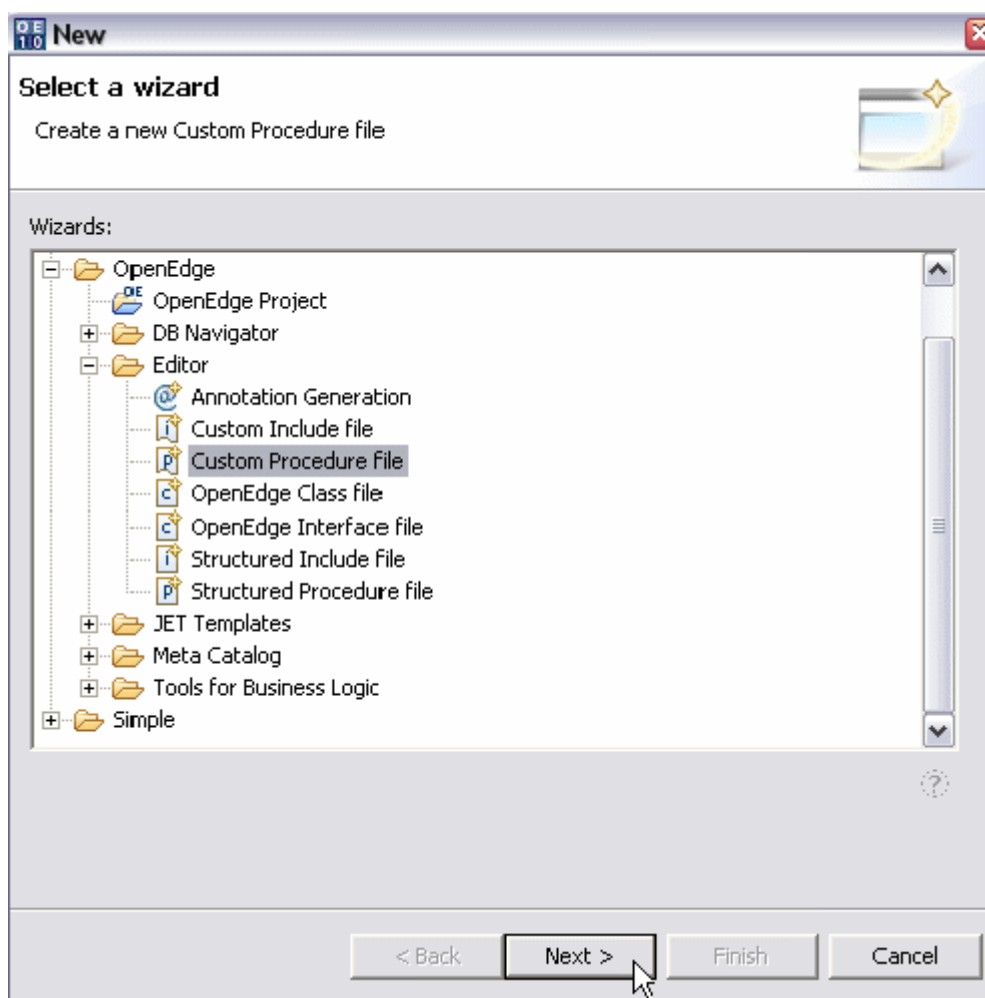
Создание нового файла с использованием File Template

File Template обеспечивает быстрый и легкий путь разработки кода на основе шаблона. Используем File Template для написания процедуры, сделав следующее.

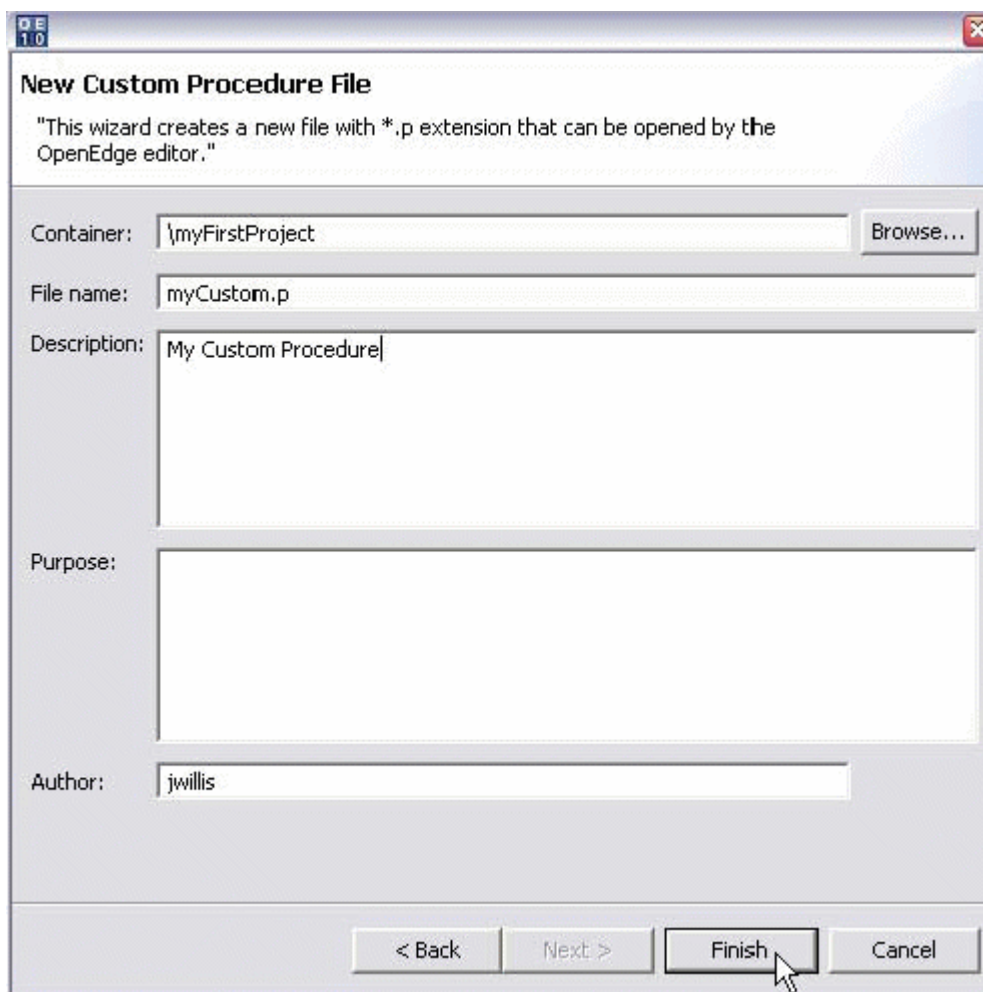
1. В Resources View кликните по myFirstProject.
2. Выберите через стрелку возле кнопки New опцию Other.



3. В окне Wizards разверните пункты OpenEdge -> Editor и выберите Custom Procedure File. Нажмите кнопку Next.



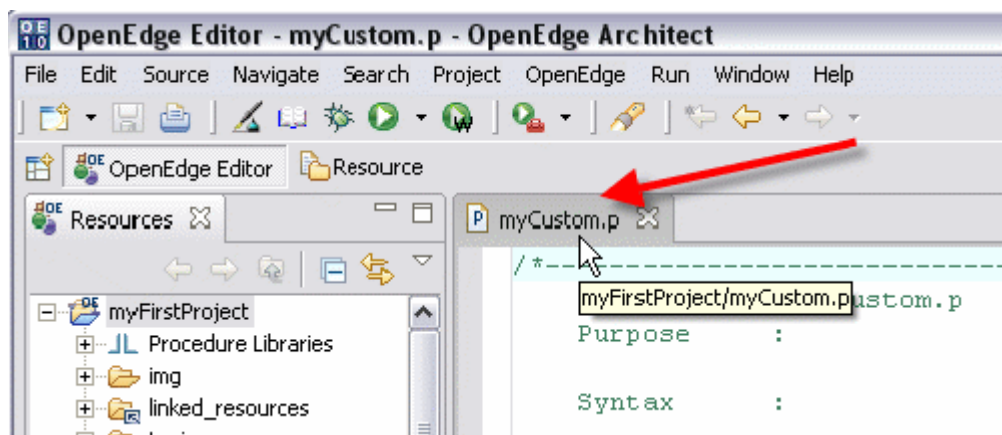
4. В открывшемся окне измените значение поля File Name на "myCustom.p" и дайте описание в поле Description, как показано ниже. Нажмите кнопку Finish.



5. В OpenEdge Editor открылась программа myCustom.p. Обратите внимание - как в новом программном файле была размещена информация, которой заполнялось окно мастера Wizard .



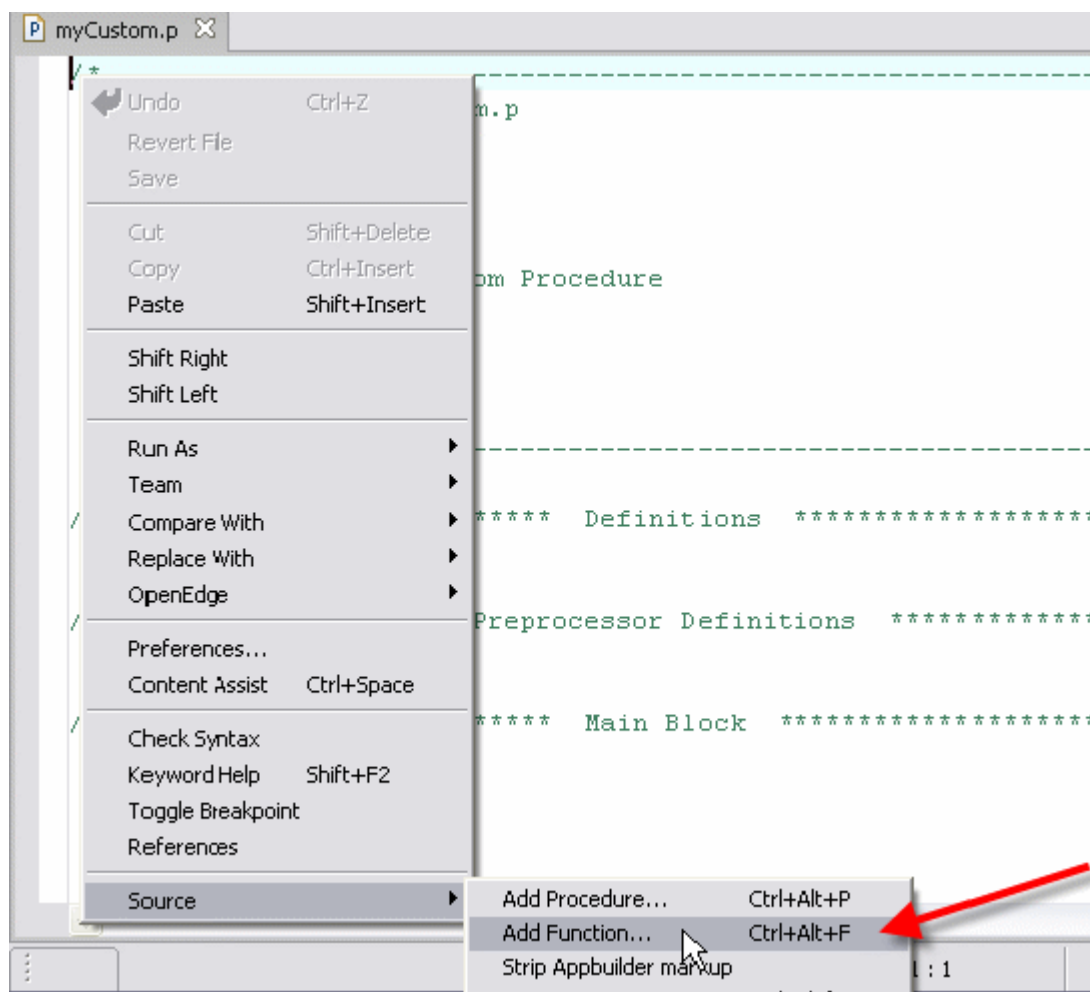
6. Кликните дважды по названию программы myCustom.p в редакторе. Редактор развернется на весь экран и займет всю площадь видимой области. Чем больше виден текст кода, тем легче становится процесс его написания.



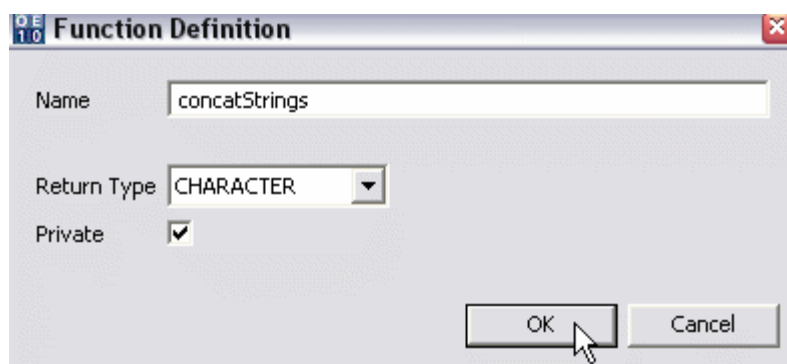
Использование Section Creators для создания фрагментов кода

Section Creators позволяет быстро создавать стандартные фрагменты кода, такие как функциональный или процедурный блок. В файле myCustom.p организуем функцию, которая бы объединяла две строки в одну с пробелом между ними. Данная функция будет использоваться при получении ФИО работников, составленного из полей First Name и Last Name.

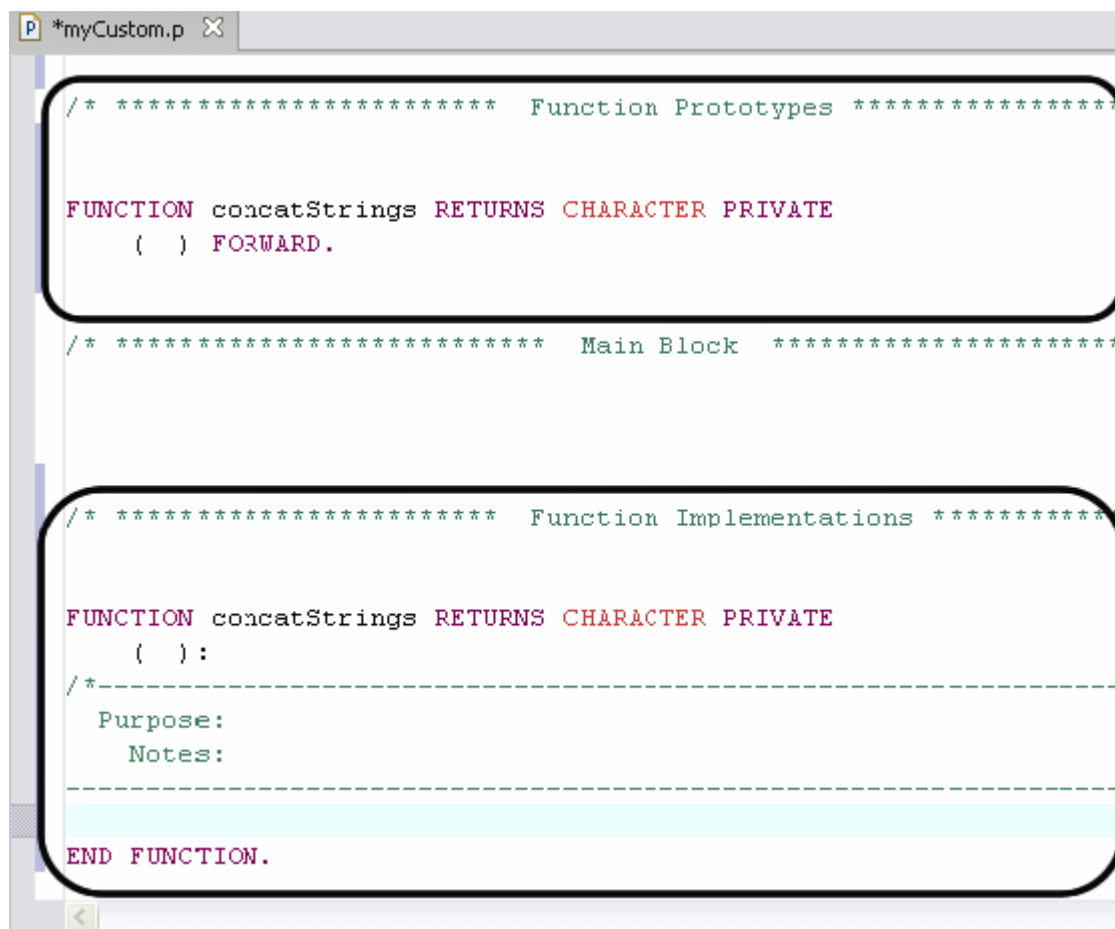
1. Правой кнопкой мыши щелкните в поле окна редактора и в появившемся меню выберите Source -> Add Function option.



2. В окне мастера Function Definition наберите “concatStrings” в поле Name. Оставив остальные значения полей по умолчанию, нажмите OK.



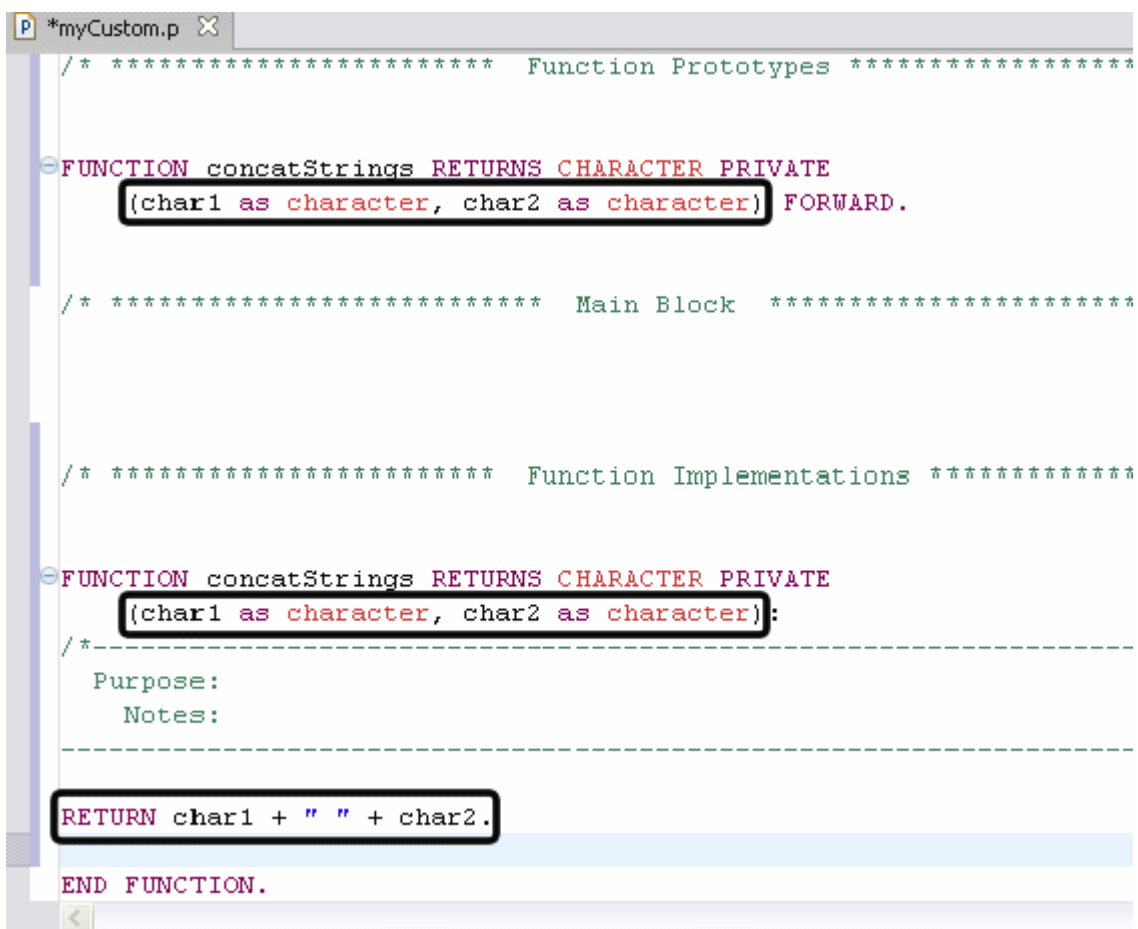
3. Созданный новый фрагмент кода функции приведен ниже.



```
/* ***** Function Prototypes *****  
  
FUNCTION concatStrings RETURNS CHARACTER PRIVATE  
    ( ) FORWARD.  
  
/* ***** Main Block *****  
  
/* ***** Function Implementations *****  
  
FUNCTION concatStrings RETURNS CHARACTER PRIVATE  
    ( ):  
/*-----  
    Purpose:  
    Notes:  
-----  
  
END FUNCTION.
```

Просмотрите, как Section Creator использовал информацию из мастера Function Definition при автоматической генерации кода. Обратите внимание, созданы поименованные раздел объявления функции Function Prototype и, содержащий текущий код, Function Implementation, которые располагаются вокруг Main Block.

4. Добавьте следующий код для функции как показано ниже.

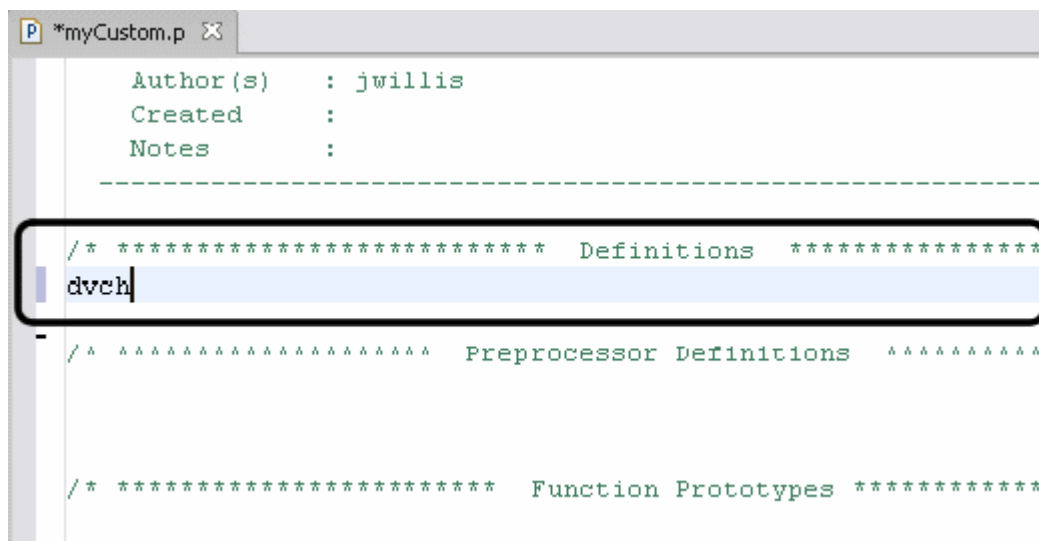


```
/* ***** Function Prototypes *****  
  
FUNCTION concatStrings RETURNS CHARACTER PRIVATE  
  (char1 as character, char2 as character) FORWARD.  
  
/* ***** Main Block *****  
  
/* ***** Function Implementations *****  
  
FUNCTION concatStrings RETURNS CHARACTER PRIVATE  
  (char1 as character, char2 as character):  
/*-----  
  Purpose:  
  Notes:  
-----  
  
  RETURN char1 + " " + char2.  
  
END FUNCTION.
```

Использование Code Templates для написания кода

Code Templates предусматривают макросы для быстрого добавления в программный код общих синтаксических выражений. Используем их для определения переменных в программе myCustom.p, выполнив следующее.

1. В разделе Definitions кода программы myCustom.p наберем “dvch”.



```
Author(s)   : jwillis  
Created    :  
Notes      :  
-----  
/* ***** Definitions *****  
dvch  
-----  
/* ***** Preprocessor Definitions *****  
  
/* ***** Function Prototypes *****
```

2. Нажмите клавишу пробела. Для вас будет сформирован синтаксис, определяющий переменную типа character. Введите имя переменной myChar и добавьте после Label и Format приведенный ниже код.

```

-----
      : myCustom.p
se   :
:    :
      :
ption : My Custom Procedure
:
(s)  : jwillis
ed   :
      :
-----*/

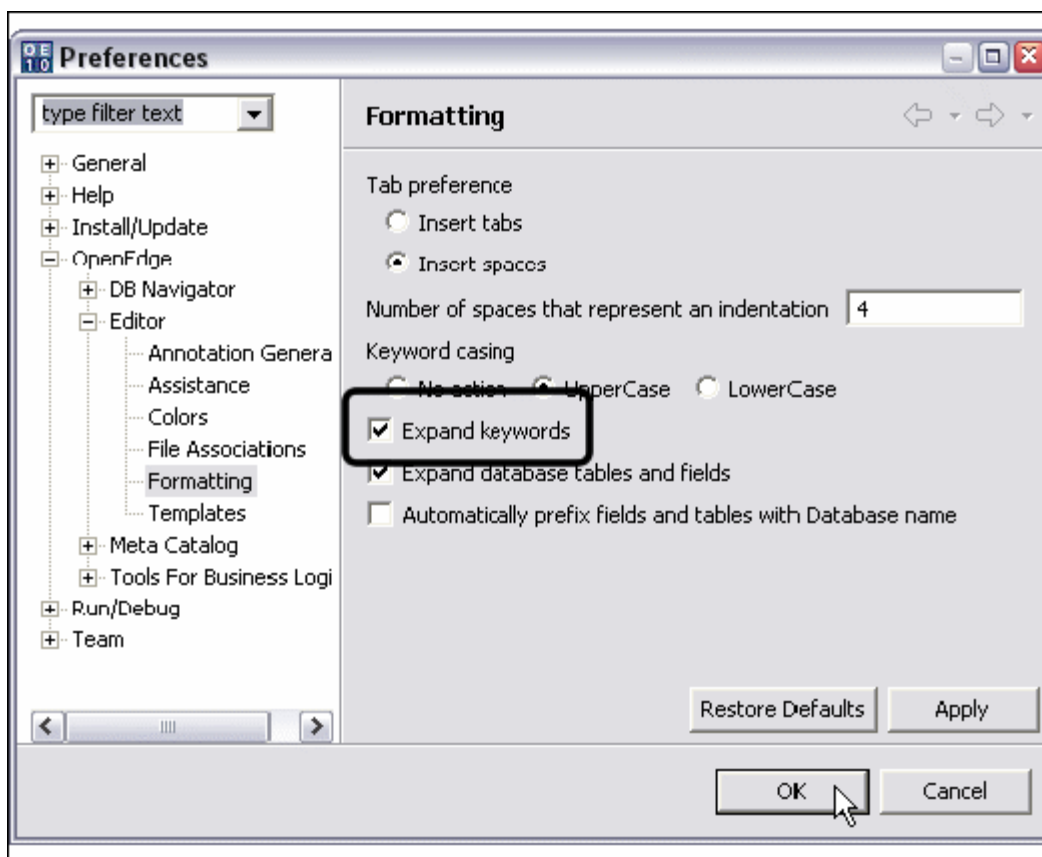
***** Definitions *****
TABLE myChar AS CHARACTER LABEL "Date of Birth" FORMAT 'x(30)' NO-UN
***** Preprocessor Definitions ***** */

```

Установка Auto-Complete Preferences

Авто-заполнение Auto-Complete экономит время разработки программы при завершении синтаксических выражений, обращениях к базе и взятии в скобки. Опция авто-заполнения Auto-Completion может быть установлена через Auto-Complete Preferences в Architect. Привилегия авто-заполнения доступна по умолчанию для расширения таблиц баз данных и имен полей. Однако не существует привилегии для расширения зарезервированных слов. Для обращения к Preference выполните следующее.

1. Откройте Window -> Preferences Main Menu option, выберите пункты OpenEdge -> Editor -> Formatting. Проверьте наличие выбора опции Expand keywords, нажмите кнопку ОК.



Использование Auto-Complete

В разделе Main Blok программного файла myCustom.p необходимо добавить код, выводящий на экран все имена покупателей и дни рождения из полей Customer Name и Dates of Birth. Ускорим этот процесс, используя некоторые свойства Auto-Complete и Auto-Format.

1. В Main Block наберем “for each cus”. Обратите внимание, каким шрифтом было выведено зарезервированное слово и, как изменился цвет “cus” после набора последнего символа “s”. Architect распознал этот частичный код, как соответствующий имени базы данных.

```

*myCustom.p
FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character) FORWARD.

/* ***** Main Block *****

FOR EACH cus

/* ***** Function Implementations *****

FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character):
/*-----
Purpose:
-----

```

- Нажмите клавишу пробела для автоматической вставки имени базы “customer”. Завершите код выражением “no-lock” и нажмите Enter или клавишу возврата.

```

*myCustom.p
FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character) FORWARD.

/* ***** Main Block *****

FOR EACH Customer NO-LOCK:

/* ***** Function Implementations *****

FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character):
/*-----

```

- Напечатайте “myChar = concatStrings?”. Обратите внимание, Architect самостоятельно закрыл (в данном случае простые) скобки.

```

P *myCustom.p X
FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character) FORWARD.

/* ***** Main Block *****
FOR EACH customer NO-LOCK.
    myChar = concatStrings(
/* ***** Function Implementations *****
FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character):
/*-----

```

4. Напишите “cus.customerf<space>, cus.customerl<space>”. Убедитесь, что после “f” и “l” поставлены пробелы и запятая между двумя выводимыми полями. Architect расширит текст для имени полей как показано ниже. Поставьте точку в конце выражения и нажмите клавишу Enter.

```

concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character) FORWARD.

/* ***** Main Block ***** */
NO-LOCK
concatStrings( Customer.CustomerFirstName , Customer.CustomerLastName ).
/* ***** Function Implementations ***** */

concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character):
/*-----

```

5. Наберите “disp” и нажмите пробел. Architect завершит написание зарезервированного слова, в соответствии с принятым синтаксисом.


```

FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character) FORWARD.

/* ***** Main Block *****

FOR EACH Customer NO-LOCK:
    myChar concatStrings(Customer.CustomerFirstName, Customer.
    DISPLAY |

/* ***** Function Implementations *****

FUNCTION concatStrings RETURNS CHARACTER PRIVATE
    (char1 as character, char2 as character):
    /*-----
    Purpose:
  
```

Использование всплывающего списка Pop-up Pick List и поддержки Syntax Help

Pick Lists – это всплывающий список синтаксических выражений, являющихся возможными продолжениями частично набранного кода. Pick List реагирует как на зарезервированные слова, так и на имена полей базы данных, причем вместе с ним также выводится на экран Pop-up Syntax Help, описывающая предлагаемые варианты выбора.

Использование Auto-Complete

Для завершения вывода на экран имен покупателей и дней рождения выполним следующее.

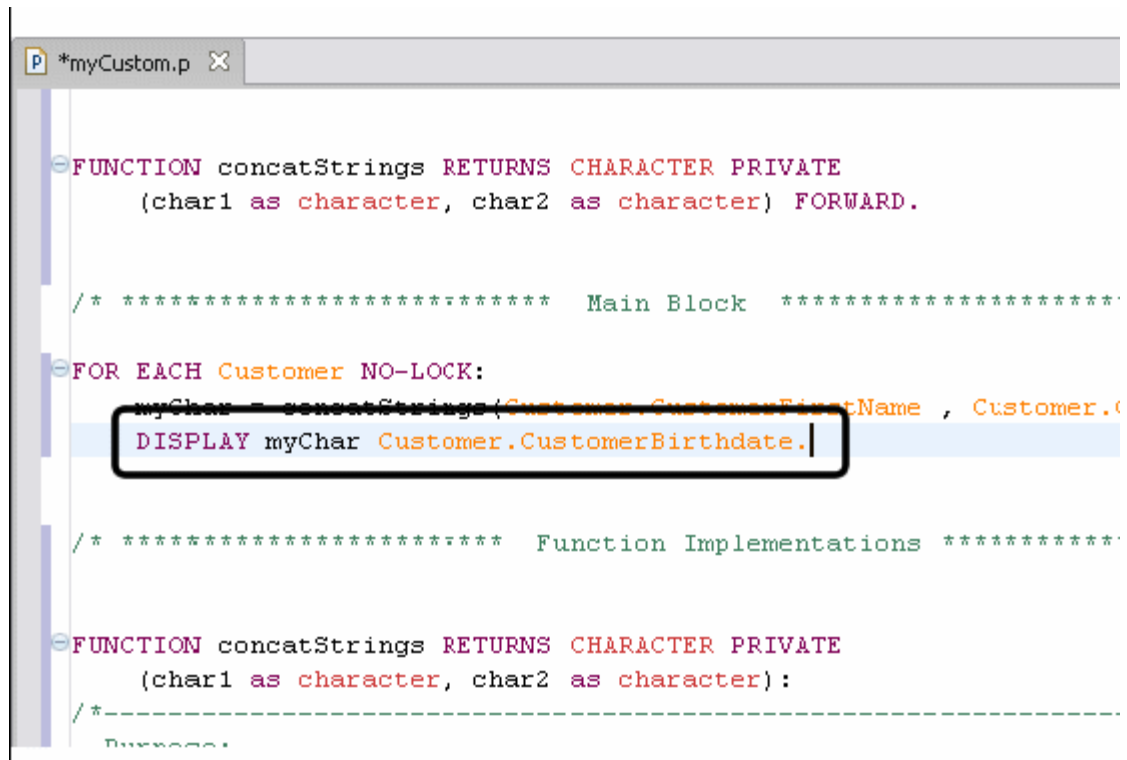
1. В редакторе установим курсор после оператора Display и через пробел добавим “myChar” и “cus.” (Architect самостоятельно допишет название имени таблицы “Customer”). Подождите немного, пока появится список выбора по базе данных Database Schema Pick List (его вызов возможен также нажатием клавиш CTRL + SPACE). Найдите в нем поле Customer.CustomerBirthdate, нажмите клавишу Enter.

The screenshot shows the OpenEdge Architect interface. The main window displays a code editor with the following code:

```
*myCustom.p X  
  
FUNCTION concatStrings RETURNS CHARACTER PRIVATE  
  (char1 as character, char2 as character) FORWARD.  
  
/* ***** Main Block *****  
  
FOR EACH Customer NO-LOCK:  
  myChar = concatStrings(Customer.CustomerFirstName , Cust  
  DISPLAY myChar Customer.  
  
RETURN char1 + " " + char2.
```

A data browser window is overlaid on the code editor, showing a list of fields for the 'autoEdge.Customer' table. The selected field is 'autoEdge.Customer.CustomerBirthdate'. The data browser also shows the value 'date 99/99/9999' and the description 'The date of birth of the applicant.' A red arrow points from the selected field in the data browser to the 'DISPLAY myChar Customer.' line in the code editor.

2. Поставьте точку по завершении строки и нажмите Enter.



```
*myCustom.p X
FUNCTION concatStrings RETURNS CHARACTER PRIVATE
  (char1 as character, char2 as character) FORWARD.

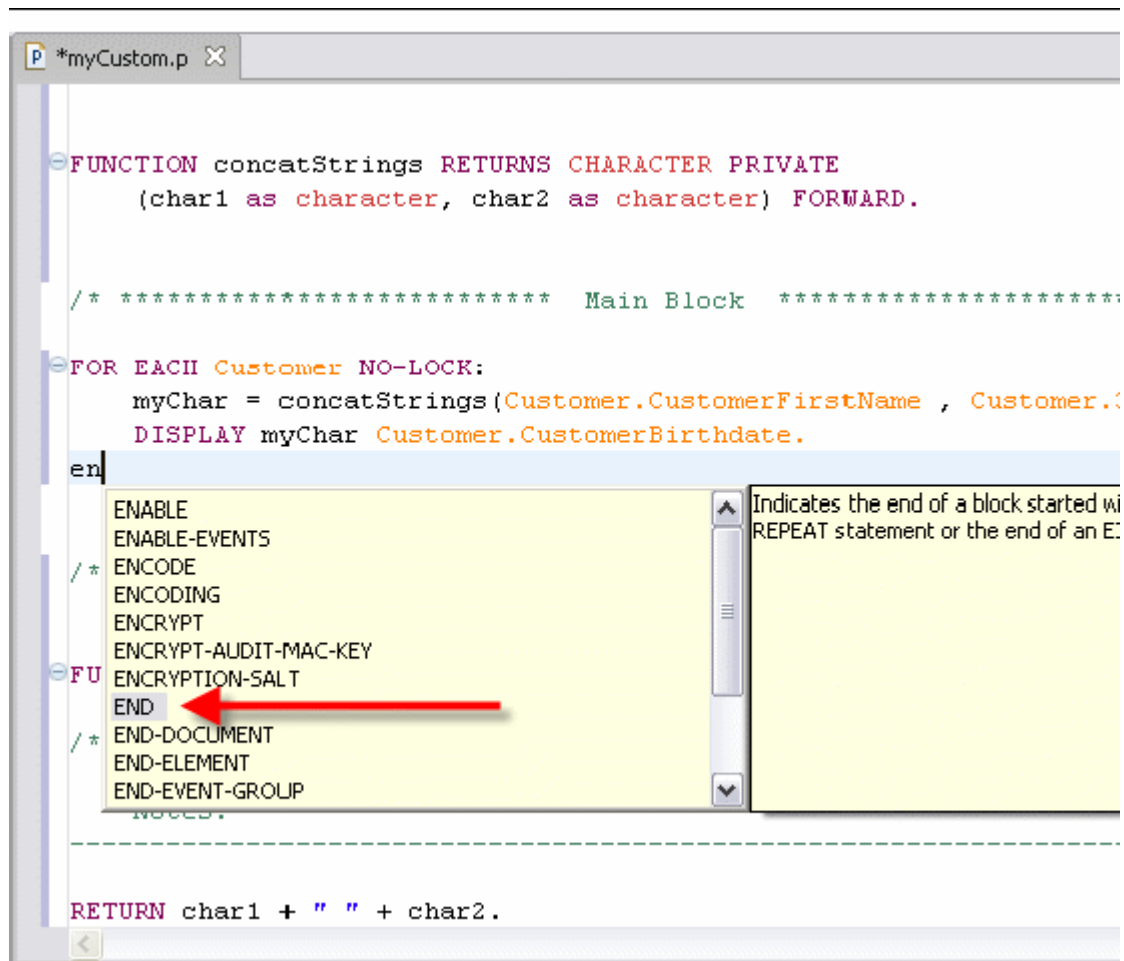
/* ***** Main Block *****

FOR EACH Customer NO-LOCK:
  myChar = concatStrings (Customer.CustomerFirstName , Customer.C
  DISPLAY myChar Customer.CustomerBirthdate.

/* ***** Function Implementations *****

FUNCTION concatStrings RETURNS CHARACTER PRIVATE
  (char1 as character, char2 as character):
/*-----
  RETURN;
```

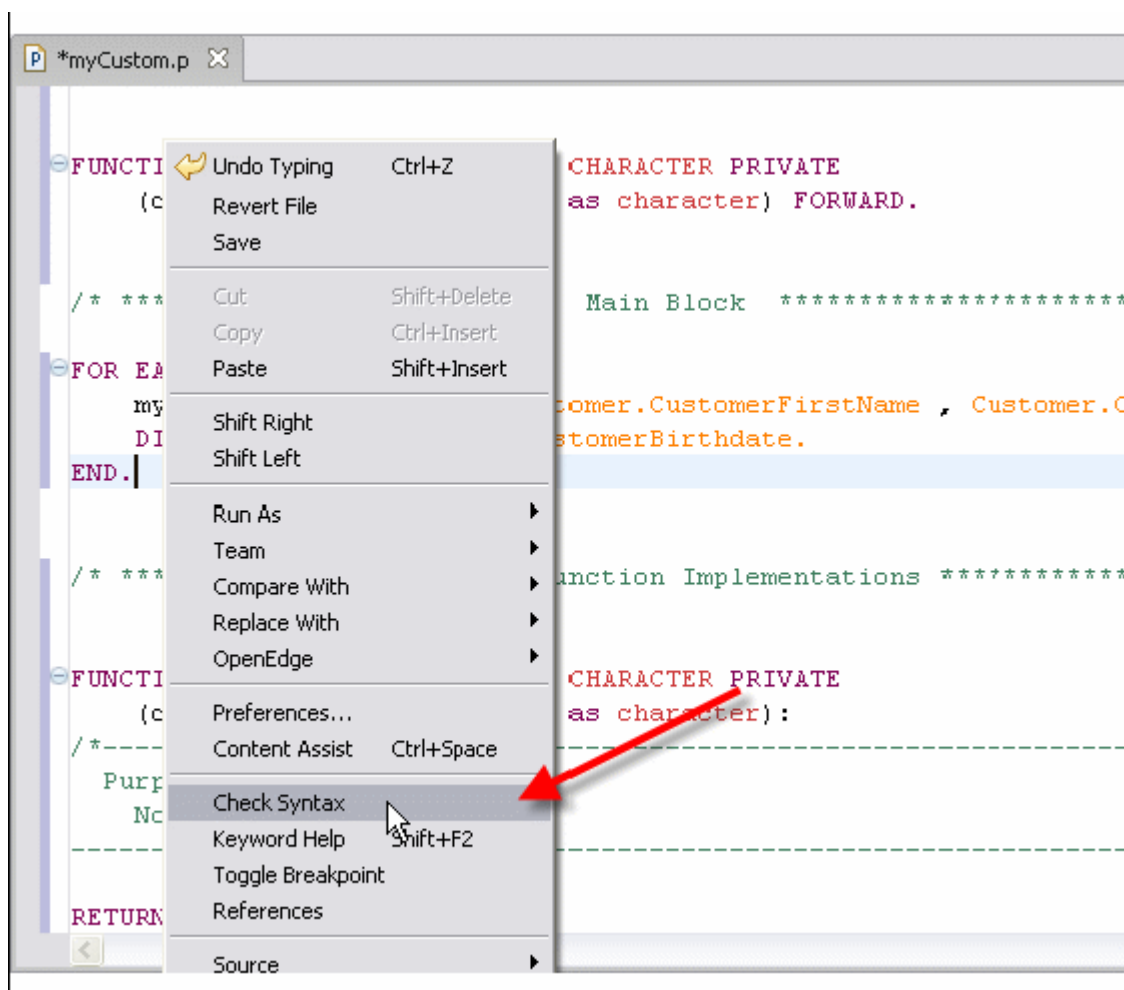
3. Установите курсор в начало следующей строки. Наберите “en” и нажмите Ctrl + Space чтобы появился список выбора зарезервированных выражений Keyword Pick List. Найдите в нем END и нажмите Enter, чтобы применить выражение. Завершите строку точкой.



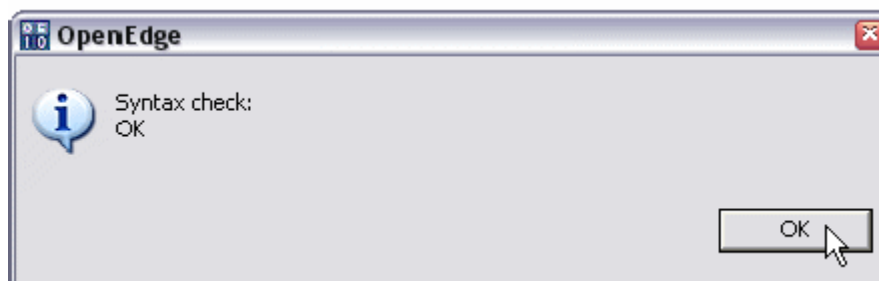
Синтаксическая проверка, сохранение и запуск программы

Для завершения и проверки программы выполним следующее.

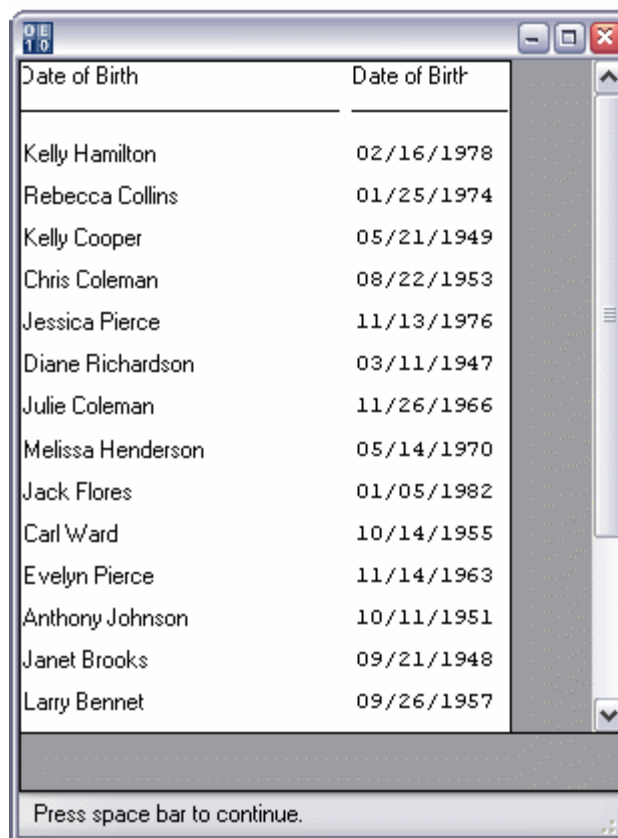
1. Щелкните правой кнопкой мыши в поле редактора и, в появившемся меню, выберите опцию проверки ошибок Check Syntax, чтобы убедиться в их отсутствии.



2. В диалоговом окне проверки синтаксиса Syntax Check нажмите OK.



3. Сохраните и запустите файл. Полученный результат выглядит примерно следующим образом.



The screenshot shows a terminal window titled "OIE 1.0". It displays a table with two columns: "Date of Birth" and "Date of Birth". The table contains 15 rows of data. At the bottom of the window, there is a prompt: "Press space bar to continue."

Date of Birth	Date of Birth
Kelly Hamilton	02/16/1978
Rebecca Collins	01/25/1974
Kelly Cooper	05/21/1949
Chris Coleman	08/22/1953
Jessica Pierce	11/13/1976
Diane Richardson	03/11/1947
Julie Coleman	11/26/1966
Melissa Henderson	05/14/1970
Jack Flores	01/05/1982
Carl Ward	10/14/1955
Evelyn Pierce	11/14/1963
Anthony Johnson	10/11/1951
Janet Brooks	09/21/1948
Larry Bennet	09/26/1957

4. Кликнув дважды в редакторе по метке с названием выполняемой программы, вернем ему первоначальный стандартный размер.

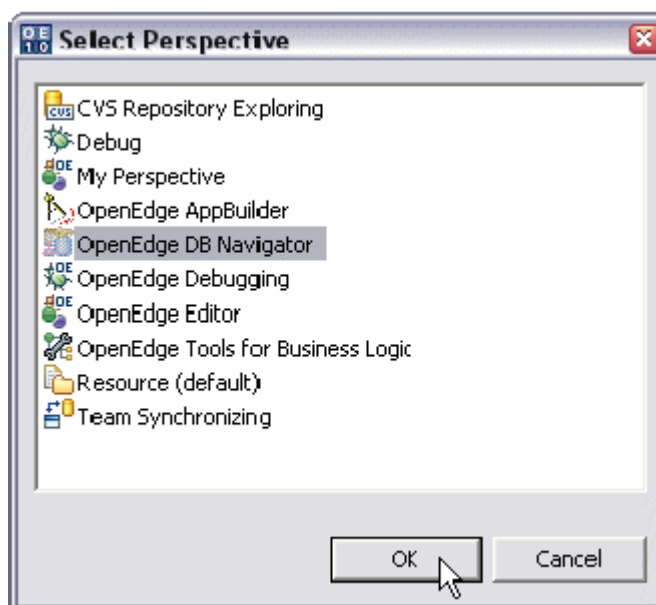
OpenEdge Database и Database Navigator

Создание новой базы данных

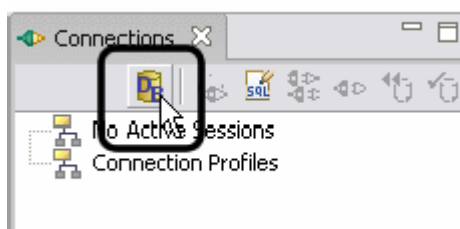
Использование Database Navigator для создания базы данных

OpenEdge упрощает создание и поддержку баз данных и их схем. Следующие шаги продемонстрируют, как создать новую базу с помощью Database Navigator.

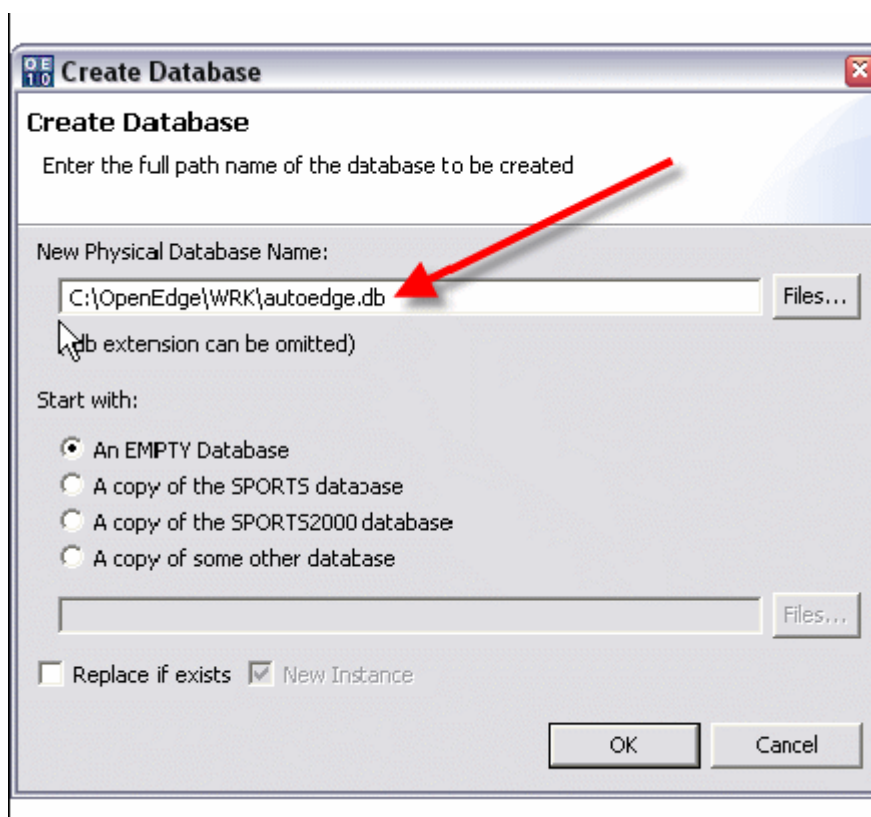
1. Откройте Database Navigator Perspective через Window -> Open Perspective -> Other option. Выберите из списка OpenEdge DB Navigator и нажмите OK.



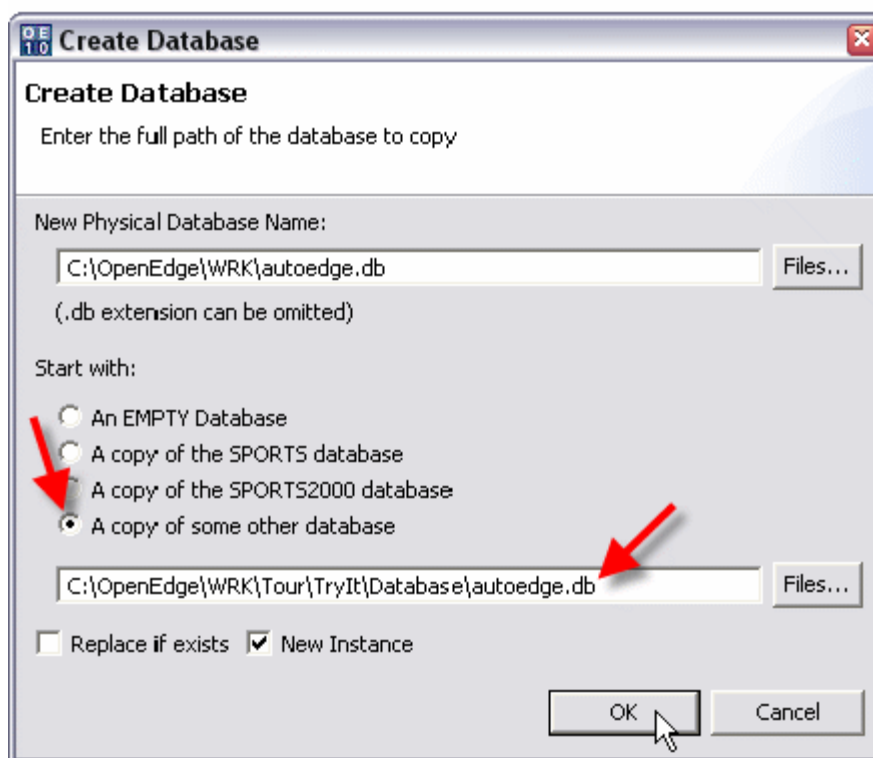
2. Для создания базы данных, выберите кнопку Create Database на панели инструментов обозревателя Connections.



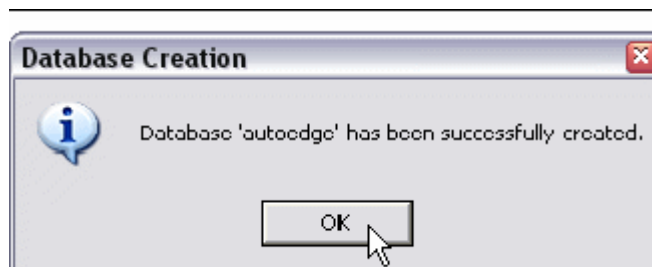
3. В поле New Physical Database Name наберите “C:\OpenEdge\WRK\autoedge.db”



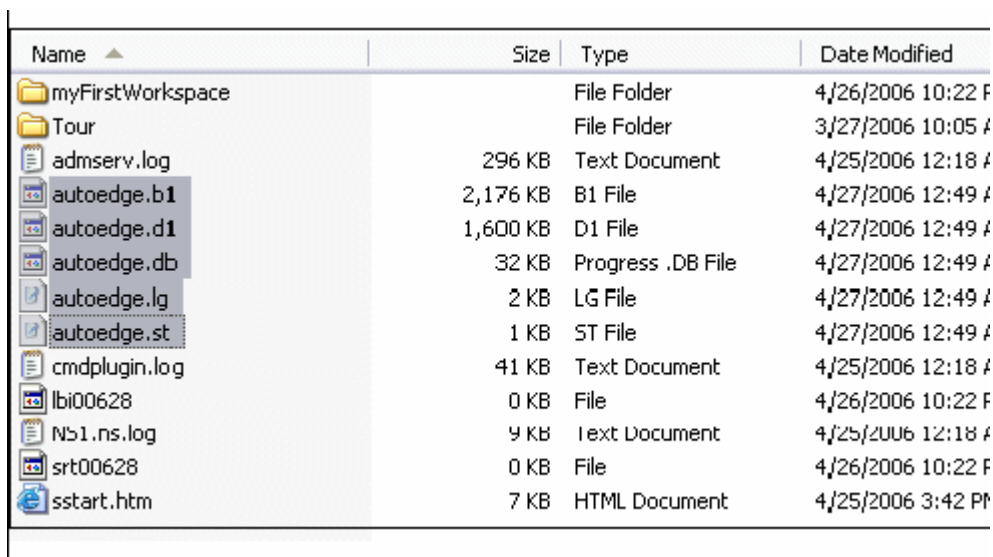
4. Выберите опцию копирования какой-либо базы “A copy of some other database” в области “Start With”. В поле, предоставленном для создания копии базы, наберите “C:\OpenEdge\WRK\Tour\TryIt\Database\autoedge.db”, либо найдите базу AutoEdge, используя для этого кнопку Files. Нажмите ОК.



5. Когда процесс будет завершен, вы получите сообщение об успешности создания базы данных. Нажмите ОК.



6. Содержимое обозревателя Database Navigator не изменилось после выполненных действий. Это связано с тем, что сначала необходимо запустить сервер для новой созданной базы, чтобы далее соединиться с ней и использовать другие возможности Navigator (это будет сделано в следующем примере). Однако, вы можете убедиться, что база действительно была создана, открыв рабочую директорию C:\OpenEdge\WRK. В ней вы увидите файлы базы данных AutoEdge, как приведено ниже.



Name	Size	Type	Date Modified
myFirstWorkspace		File Folder	4/26/2006 10:22 F
Tour		File Folder	3/27/2006 10:05 F
admserv.log	296 KB	Text Document	4/25/2006 12:18 F
autoedge.b1	2,176 KB	B1 File	4/27/2006 12:49 F
autoedge.d1	1,600 KB	D1 File	4/27/2006 12:49 F
autoedge.db	32 KB	Progress .DB File	4/27/2006 12:49 F
autoedge.lg	2 KB	LG File	4/27/2006 12:49 F
autoedge.st	1 KB	ST File	4/27/2006 12:49 F
cmdplugin.log	41 KB	Text Document	4/25/2006 12:18 F
lbi00628	0 KB	File	4/26/2006 10:22 F
N51.ns.log	9 KB	Text Document	4/25/2006 12:18 F
srt00628	0 KB	File	4/26/2006 10:22 F
sstart.htm	7 KB	HTML Document	4/25/2006 3:42 PM

Настройка и запуск сервера базы данных

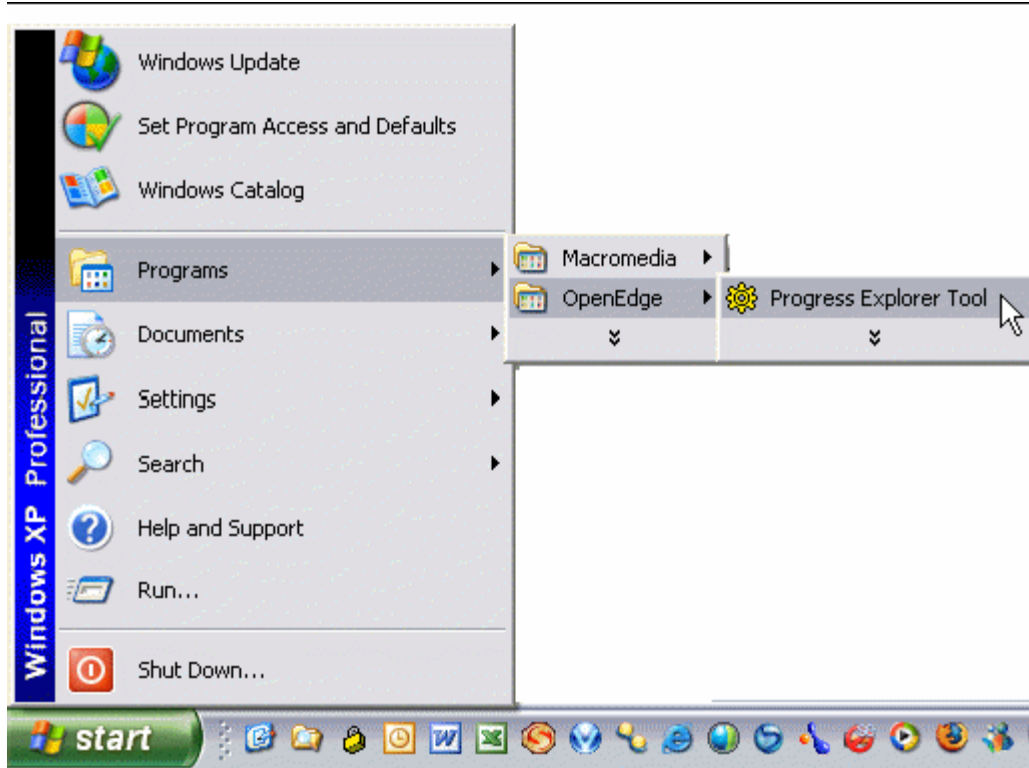
Конфигурирование сервера базы данных

Изначально, соединение базы данных с Database Navigator – многоступенчатый процесс. Сначала, необходимо сконфигурировать сервер для базы, а затем уже запускать на нем процессы, использующие определенную информацию конфигурации. Далее понадобится создать профайл соединения Connection Profile в навигаторе Database Navigator, чтобы последнему стал доступен сервер базы данных Database Server. В конечном итоге именно Connection Profile будет использоваться для соединения с сервером.

Повторять шаги по конфигурированию Database Server и созданию Connection Profile при последующих соединениях с базой данных нет необходимости, т.к. настройки сохраняются для их дальнейшего использования. Но при этом понадобится запускать Database Server и Connection Profile каждый раз при очередных соединениях. Настройки, необходимые для Database Server и Connection Profile, запускаются автоматически и будут рассмотрены детально при их пошаговой ручной установке.

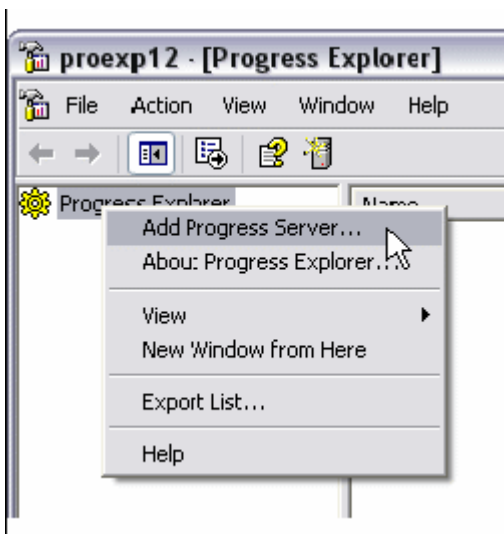
Выполните следующие шаги, чтобы сконфигурировать Database Server для базы данных AutoEdge, созданной ранее.

1. Через меню Start откройте Progress Explorer tool.



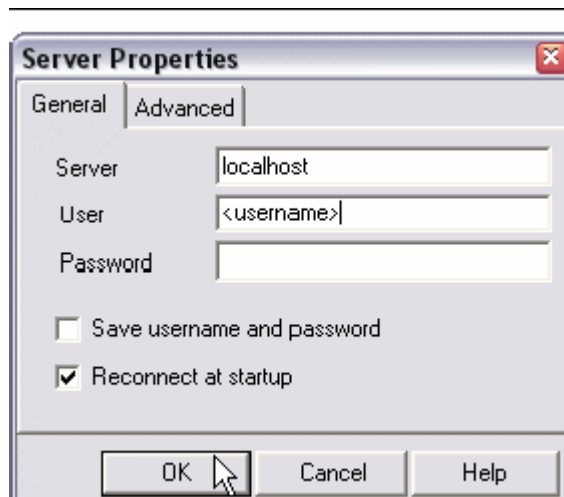
- Для создания Database Server, сначала потребуется соединиться с административным сервером Administration Server, который уже запущен как приложение. Если существуют какие-либо трудности с соединением, проверьте его наличие в Service Administration tool для Windows.

Кликните правой кнопкой мыши по пункту Progress Explorer и выберите опцию Add Progress Server.

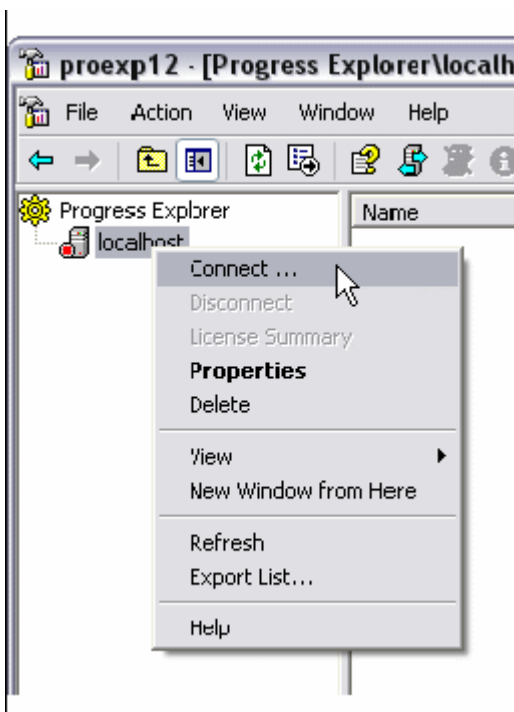


- Появилось окно свойств Server Properties, в котором в поле Server указано значение "localhost", в поле User – логин Вашего компьютера. Выберите опцию переподключения при запуске Reconnect at Startup, чтобы каждый раз при старте

Progress Explorer Tool прежде устанавливалось соединение с Administration Server.
Нажмите ОК.



4. Сформирован Administrative Server названный "localhost". Правой кнопкой мыши кликните по пункту "localhost" и выберите опцию Connect.

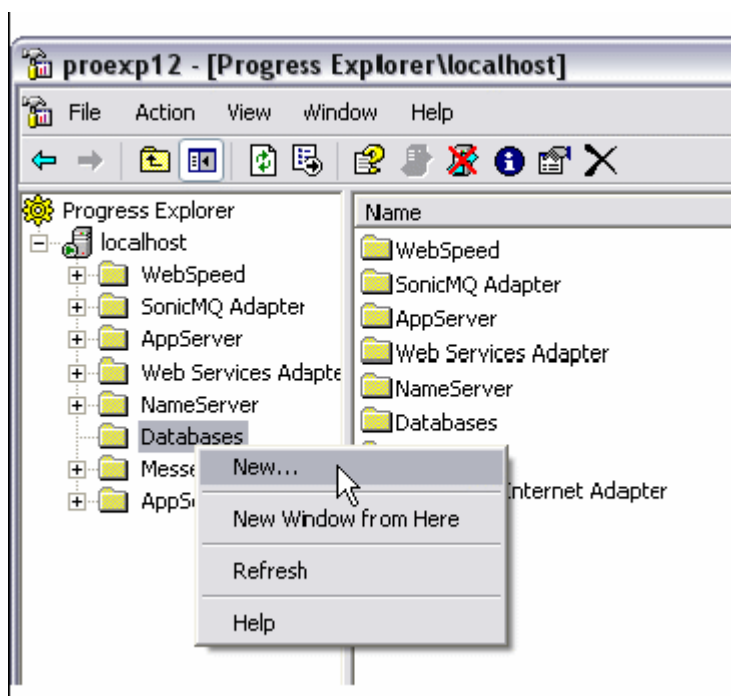


5. Открылось окно Login, где в поле User указан логин компьютера. Не изменяя данного значения и не заполняя поле Password, нажмите кнопку ОК.

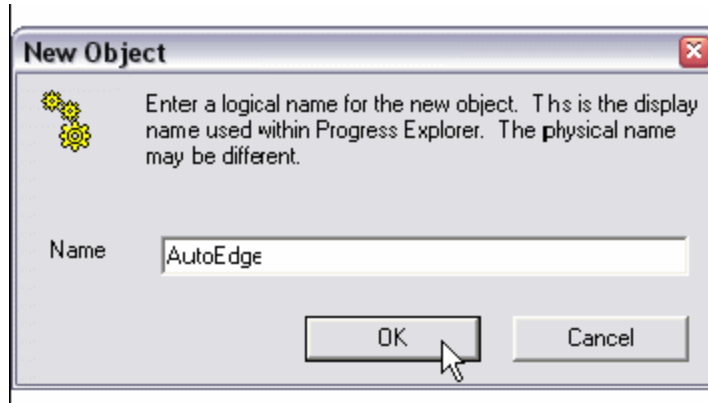


Подключившись таким образом, вы соединитесь с Administration Server, не задействовав логин и пароль. Если вы используете конфигурацию по-умолчанию, заполнение полей User и Password не обязательно. Если в течение процесса инсталляции продукта данная опция безопасности была задействована, в дальнейшем для соединения с Administration Server необходимо будет использовать примененные вами значения (обычно совпадающие с логином и паролем для доступа к компьютеру). Вы можете также установить данные значения, вернувшись в окно Server Properties (пункт 3), и сохранить их для будущего использования.

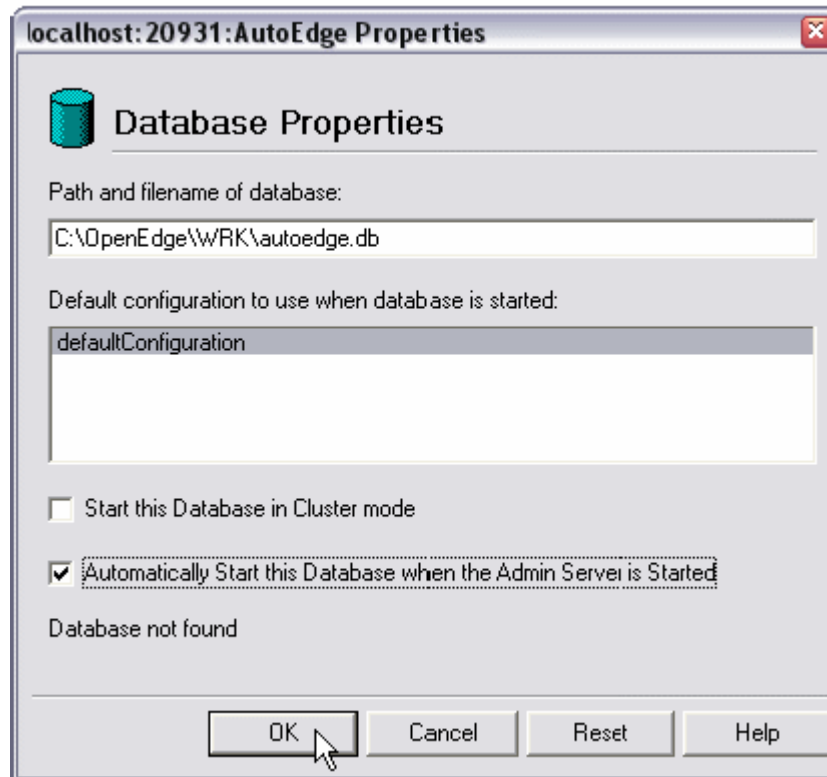
6. Если соединение с Administration Server произошло успешно, localhost заполнится новым содержимым. Правой кнопкой мыши щелкните по пункту Databases и выберите опцию New для создания нового сервера Database Server.



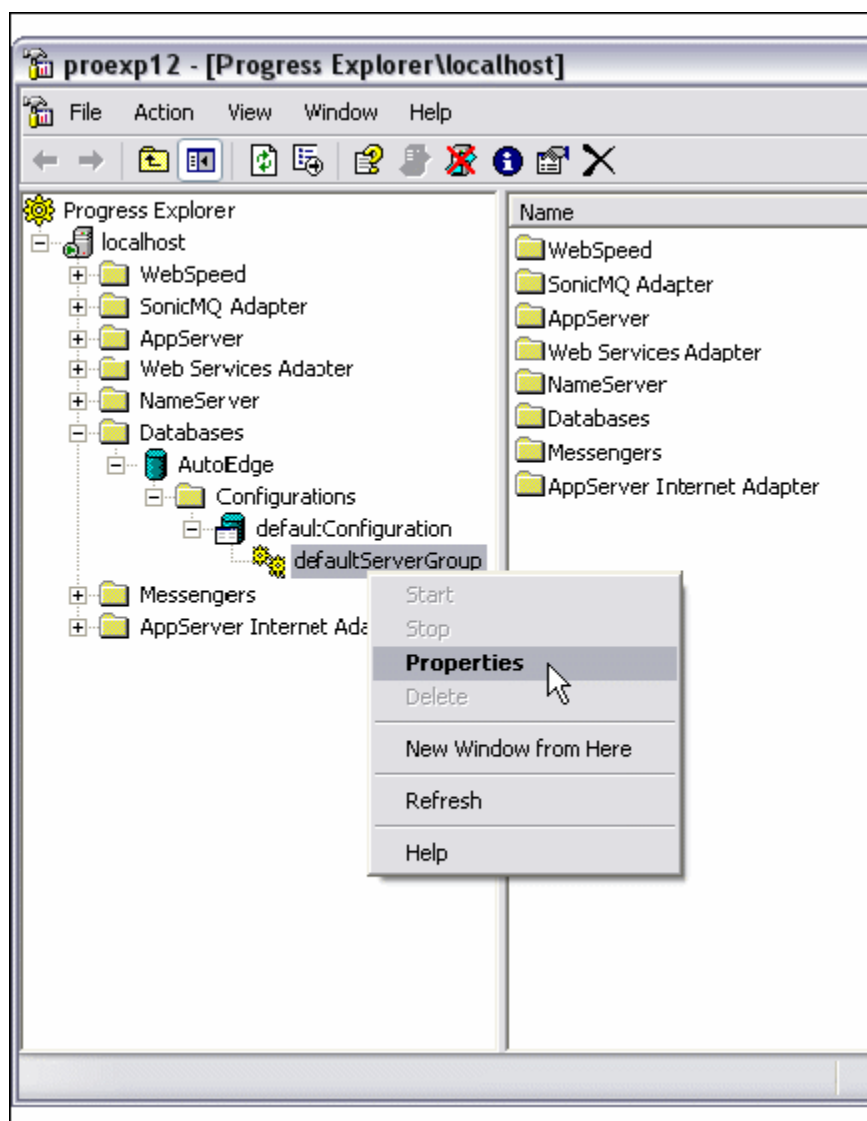
7. Наберите в поле Name название сервера базы данных AutoEdge, нажмите OK.



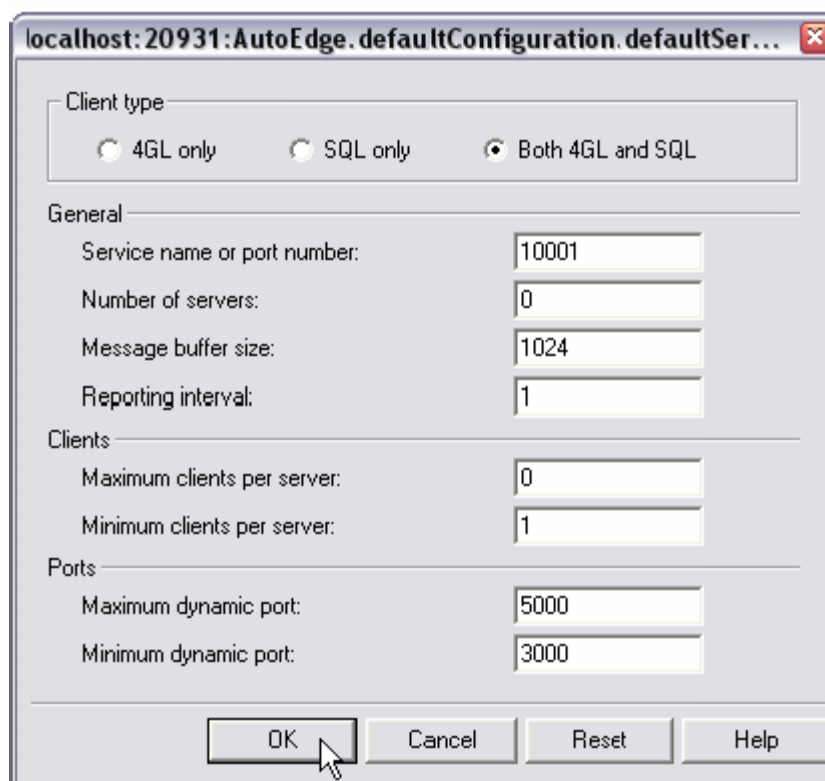
8. В окне свойств Database Properties, наберите “C:\OpenEdge\WRK\autoedge.db” в поле для выбора пути и названия базы данных path and filename of the database. Выберите опцию “Automatically Start this Database when the Admin Server is Started”. Эта опция запускает сервер базы данных Database Server при очередном старте Administration Server. Нажмите ОК.



9. Созданный сервер базы данных отразился ниже пункта Database. Далее необходимо установить свойства группы сервера Server Group Properties. Разверните следующие пункты Database -> AutoEdge -> Configuration -> defaultConfiguration. Щелкните правой кнопкой мыши по defaultServerGroup и выберите опцию Properties.



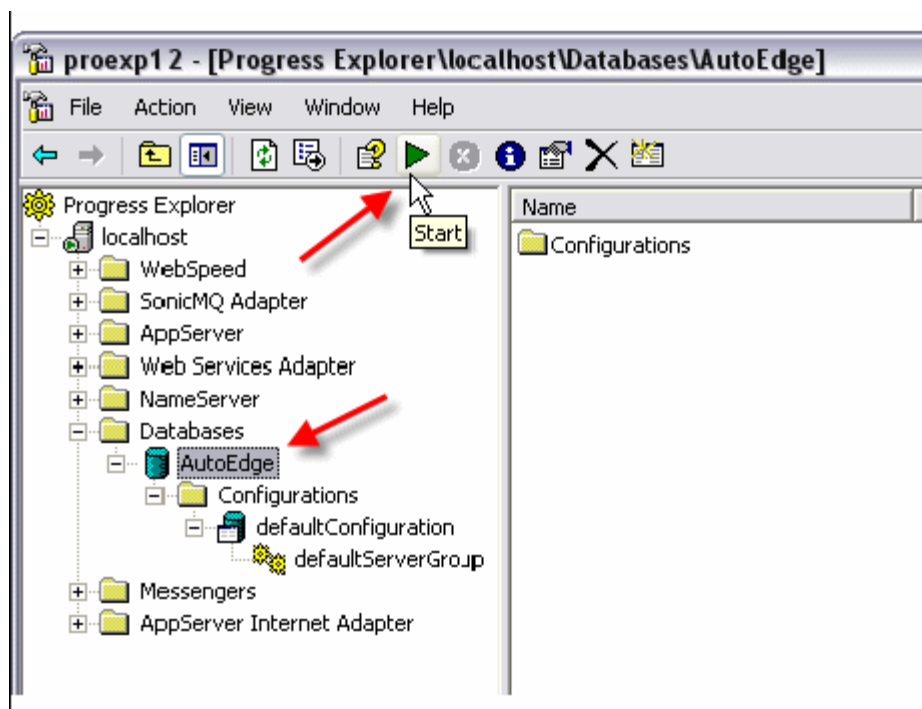
10. Убедитесь, что в разделе тип клиента Client Type появившегося окна Server Group Properties выбрана опция "Both 4GL and SQL". Она позволит базе данных присоединяться как с помощью ABL, так и SQL инструмента. Установите в поле "Service Name or Port Number" значение 10001 или любое другое, являющееся неиспользуемым номером порта. Список зарезервированных номеров портов компьютера можно найти в служебном файле, который обычно располагается на "C:\WINDOWS\system32\drivers\etc". После завершения нажмите OK.



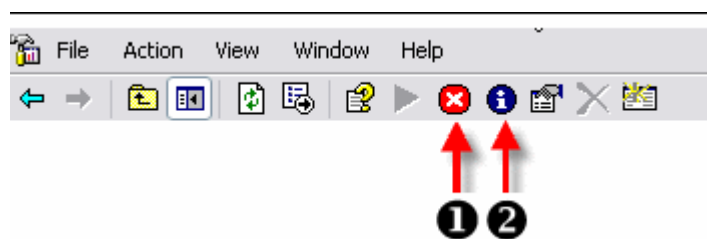
Запуск сервера базы данных

Теперь, после конфигурации Database Server, возможен его ручной запуск. Если же Database Server был установлен с учетом автоматического подключения (см. пункт 6), необходимость запускать его вручную после первого раза исчезает.

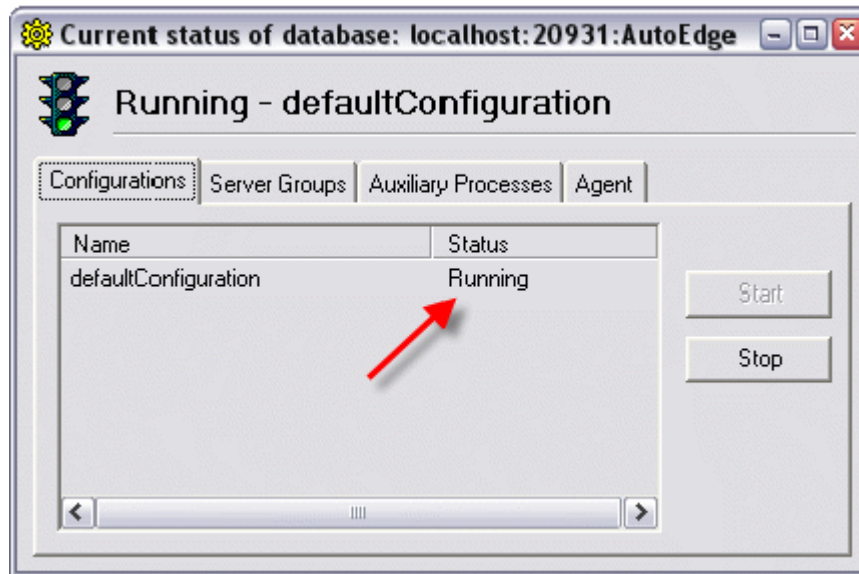
1. Выберите пункты Databases -> AutoEdge и нажмите зеленую кнопку запуска Start на панели инструментов.



2. После запуска Database Server кнопка Start сменится кнопкой Stop (красный круг с надписью "X"). Вы можете также удостовериться, что сервер запущен, нажав на кнопку состояния Status (синий круг с буквой "i").



Если соединение успешно, в окне Status будет запись Running.



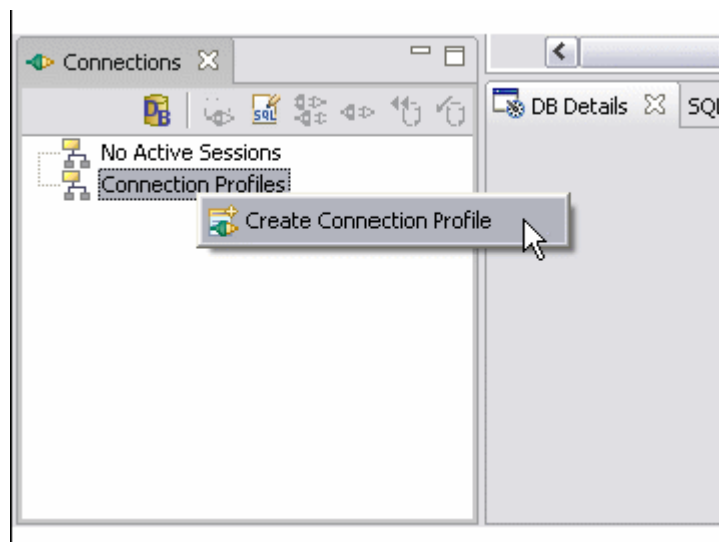
3. Выполнив все действия, закройте окно Progress Explorer Tool.

Соединение Database Navigator с базой данных

Соединение Database Navigator с сервером Database Server

После создания и запуска Database Server необходимо установить соединение, т.е. сконфигурировать Database Navigator для организации доступа к базе данных. Для этого потребуется сформировать Connection Profile базы. Выполните следующие шаги для соединения Database Navigator с AutoEdge Database Server, используя Connection Profile.

1. Правой кнопкой мыши кликните по пункту Connection Profiles в окне соединений Connections. Выберите опцию создания профайла Create Connection Profile.



2. Заполните поля Name, Host, Port, Database как указано ниже. Также выберите опцию “Open on Eclipse Startup”. Она позволяет автоматически устанавливать соединение с базой данных при каждом запуске Architect. Завершив действия, нажмите кнопку ОК.

Add new Connection Profile

New Connection Profile

Select to start the connection when Eclipse starts. A valid user and password must be entered.

Name:

Driver:

URL:

Host Details:

Host:

Service Port:

Database:

Additional Parameters:

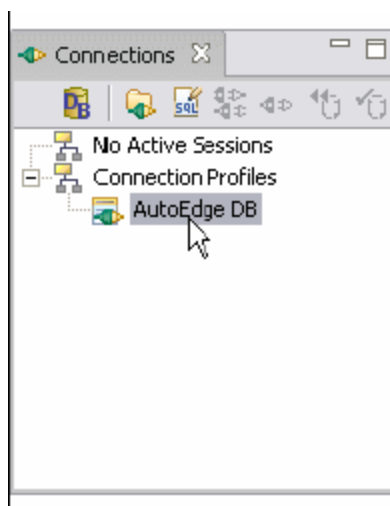
User Name:

Password:

Open on Eclipse Startup

Примечание: Вы можете ввести данные об имени пользователя и пароле для соединения с базой данных в поля User Name и Password, Architect сохранит их. Однако данная информация не зашифровывается, поэтому вы можете отказаться от хранения таких данных здесь.

3. По-окончании последнего шага, ниже Connection Profiles появится пункт AutoEdge DB. Кликните дважды по нему, чтобы установить соединение с базой autoEdge. Вам необходимо будет повторять этот и следующие шаги каждый раз при соединении с базой данных, кроме случая, если в профайле Profile (см. пункт 2) было указано автоматическое подключение.

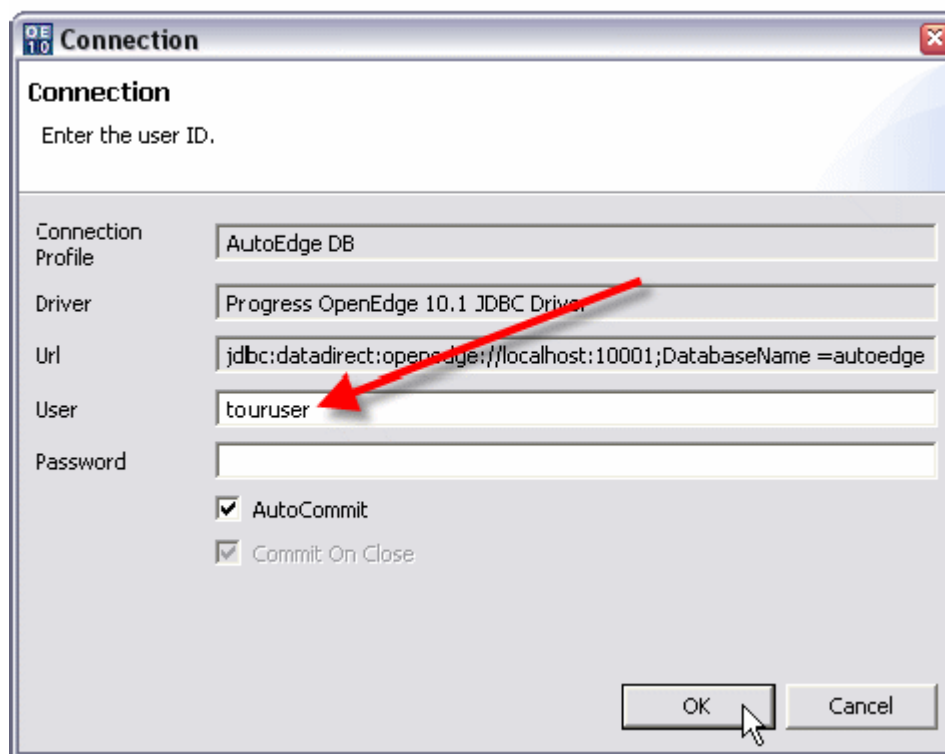


4. Перед завершением процесса, вам будет предложено соединиться с базой, подтвердив комбинацию имени пользователя и пароля, в окне, напоминавшем ранее о возможности внесения этих данных без их дальнейшего сохранения в Architect.

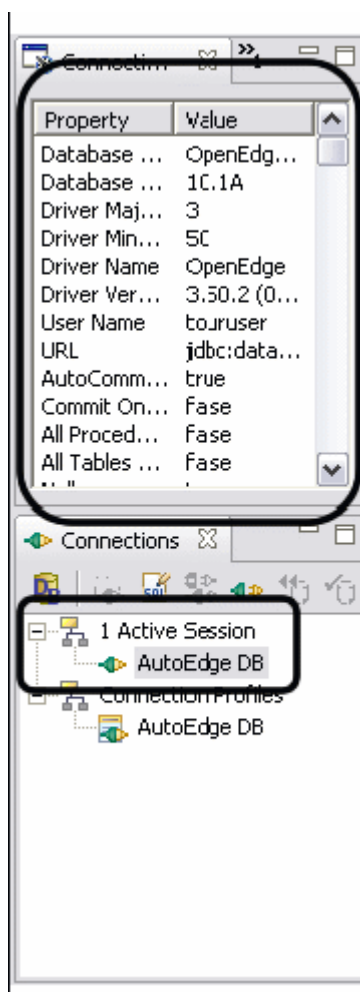
Т.к. вы только что создали базу данных, подтверждение логина отключено по-умолчанию. Для SQL-соединения все же требуется имя пользователя, даже если у него не существует соответствующей учетной записи в базе. Вы можете набрать любое имя пользователя, оставив пароль пустым, чтобы соединиться с базой данных для просмотра ее схемы.

Однако для изменения схемы базы потребуется соблюдение правил безопасности DBA. Как правило, при создании базы учетная запись SQL DBA создается автоматически, по-умолчанию используя название компьютера. Но у данной базы есть свое User Name.

Используйте User Name "touruser" и пустой пароль для подключения к базе данных autoEdge. По-завершении нажмите ОК.



5. Если соединение произошло успешно, ниже Active Session вы увидите пункт “AutoEdge DB”. Информативное окно Info Connection будет также заполнено данными про базу и ее подключение.

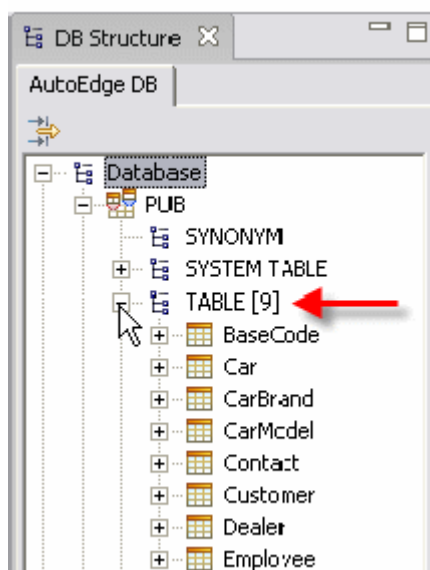


Исследование и модификация схемы базы данных

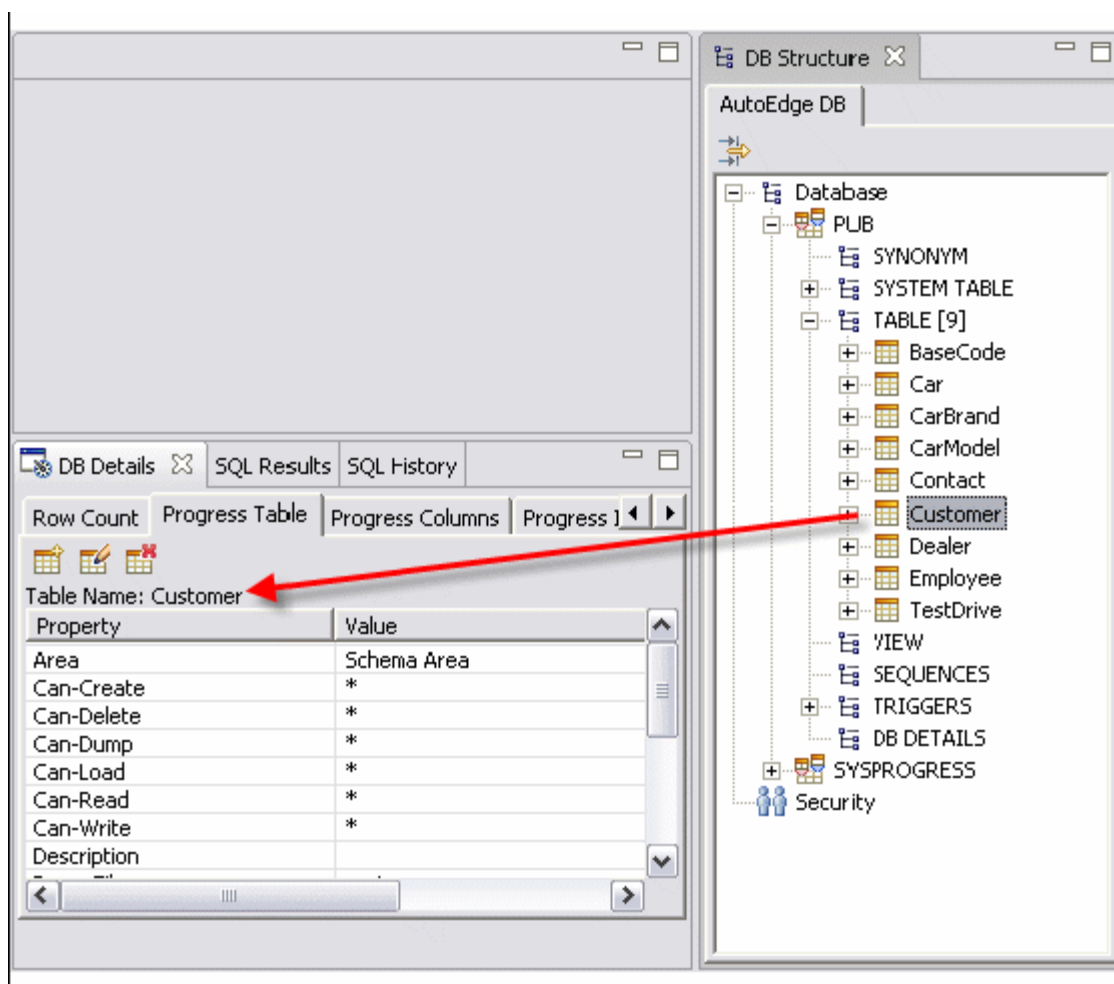
Исследование схемы базы данных

Database Navigator облегчает просмотр и внесение изменений в схему базы данных. Чтобы использовать Navigator для изучения табличной схемы базы AutoEdge, выполните следующее.

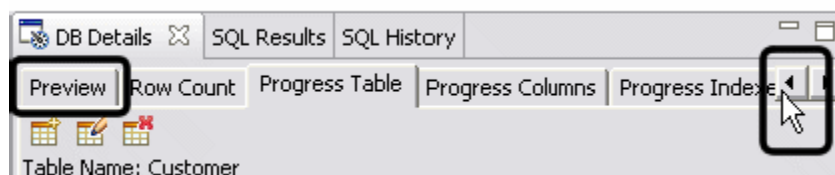
1. В окне DB Structure для AutoEdge DB раскройте пункты Database -> Pub -> Table. Проверьте, что в окне отразились 9 содержащихся в базе таблиц.



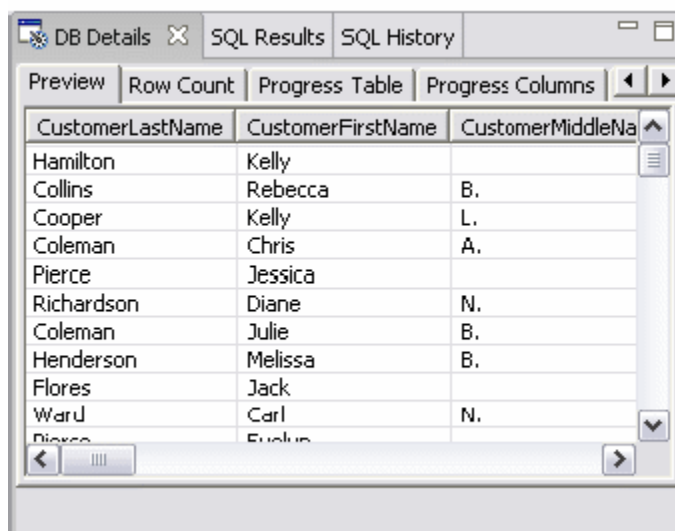
2. Выберите пункт Customer. При этом в окне детализации DB Details отобразится информация о таблице Customer.



3. В окне DB Details с помощью левой стрелки прокрутки найдите вкладку Preview.



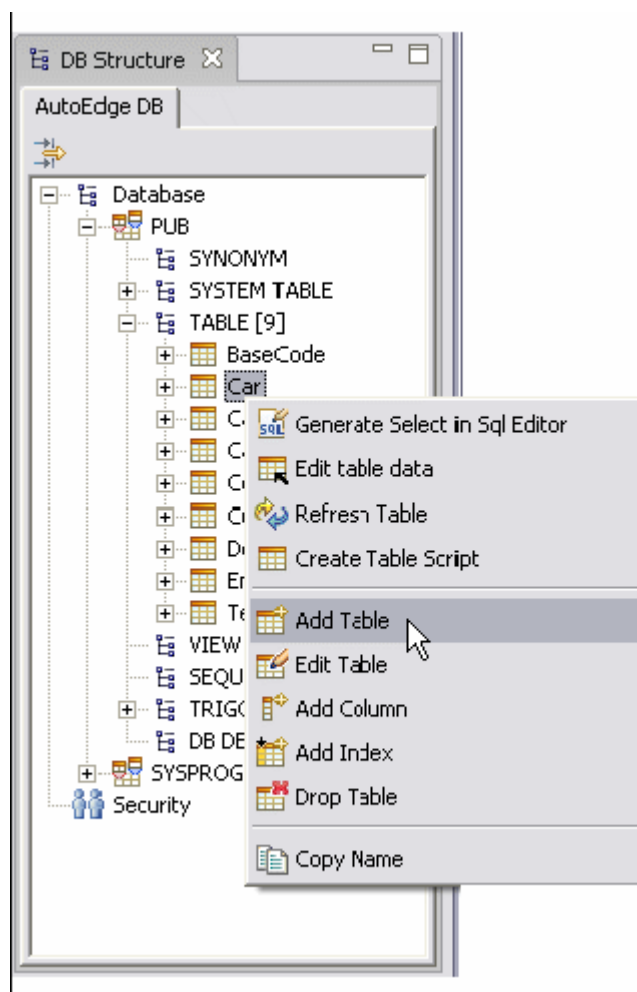
4. Выберите вкладку Preview. В ней отражаются текущие данные из таблицы базы.



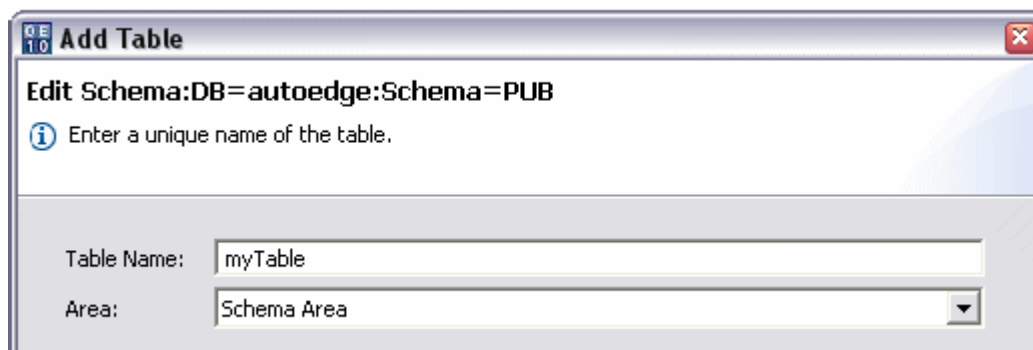
Модификация схемы базы данных

Чтобы использовать Navigator для изменения схемы базы AutoEdge и добавления в нее новой таблицы с полями и индексами, выполните следующее.

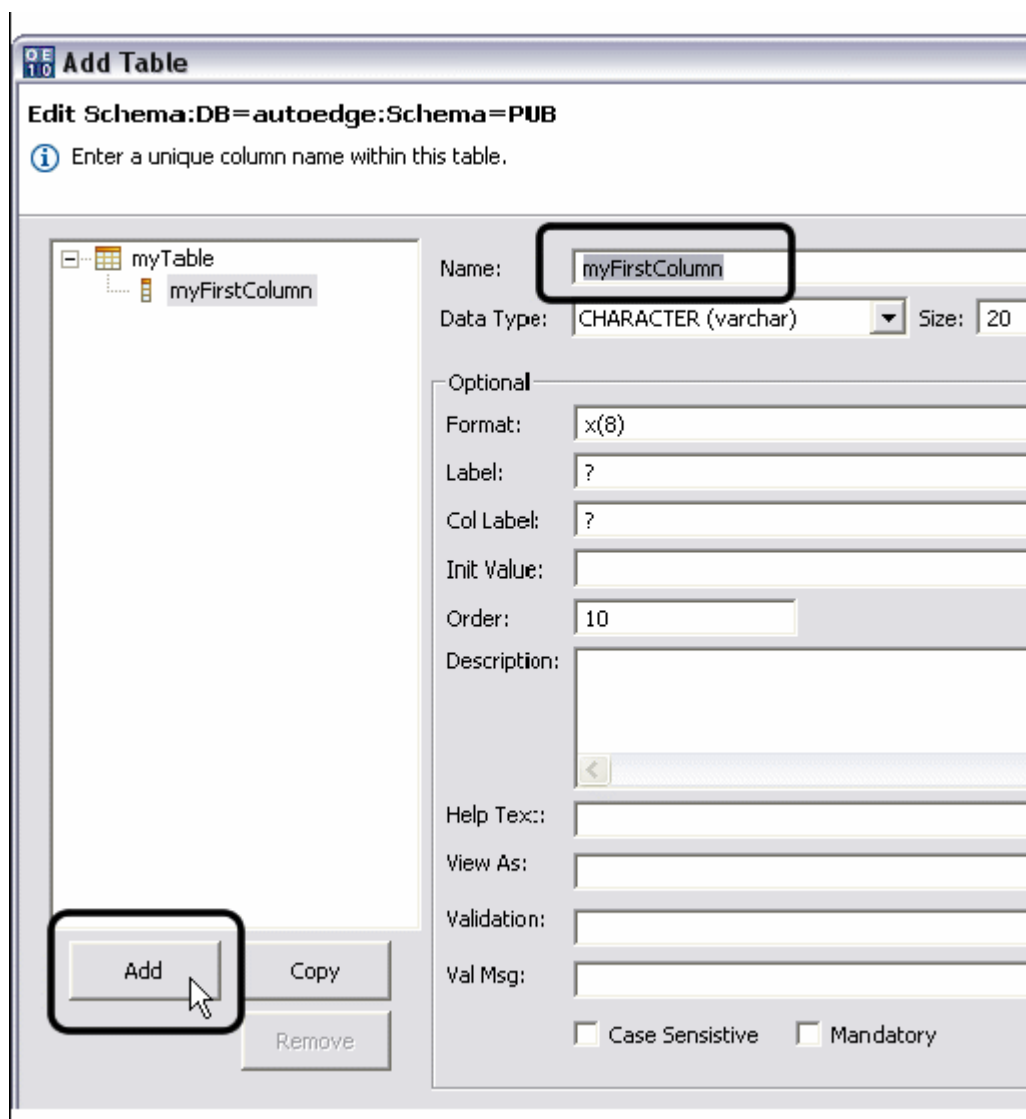
1. В окне DB Structure правой кнопкой мыши кликните по любой из таблиц, находящихся под пунктом Table и в появившемся меню выберите опцию добавления Add Table.



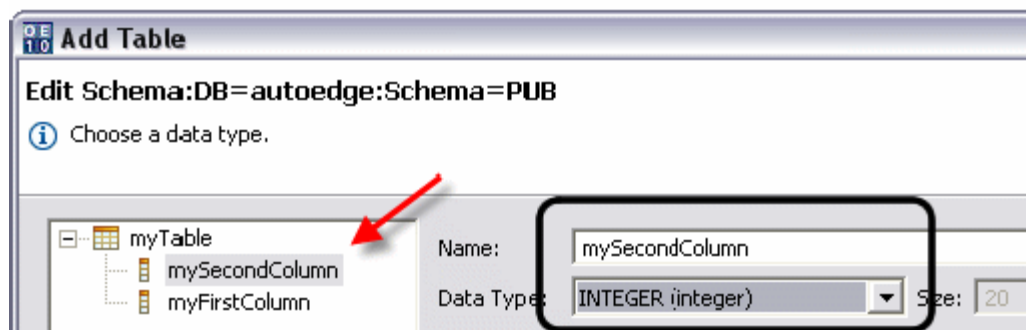
2. Для добавления таблицы будет вызван соответствующий мастер. В открывшемся окне Add Table наберите “myTable” в поле имени таблицы Table Name. Нажмите кнопку Next для продолжения.



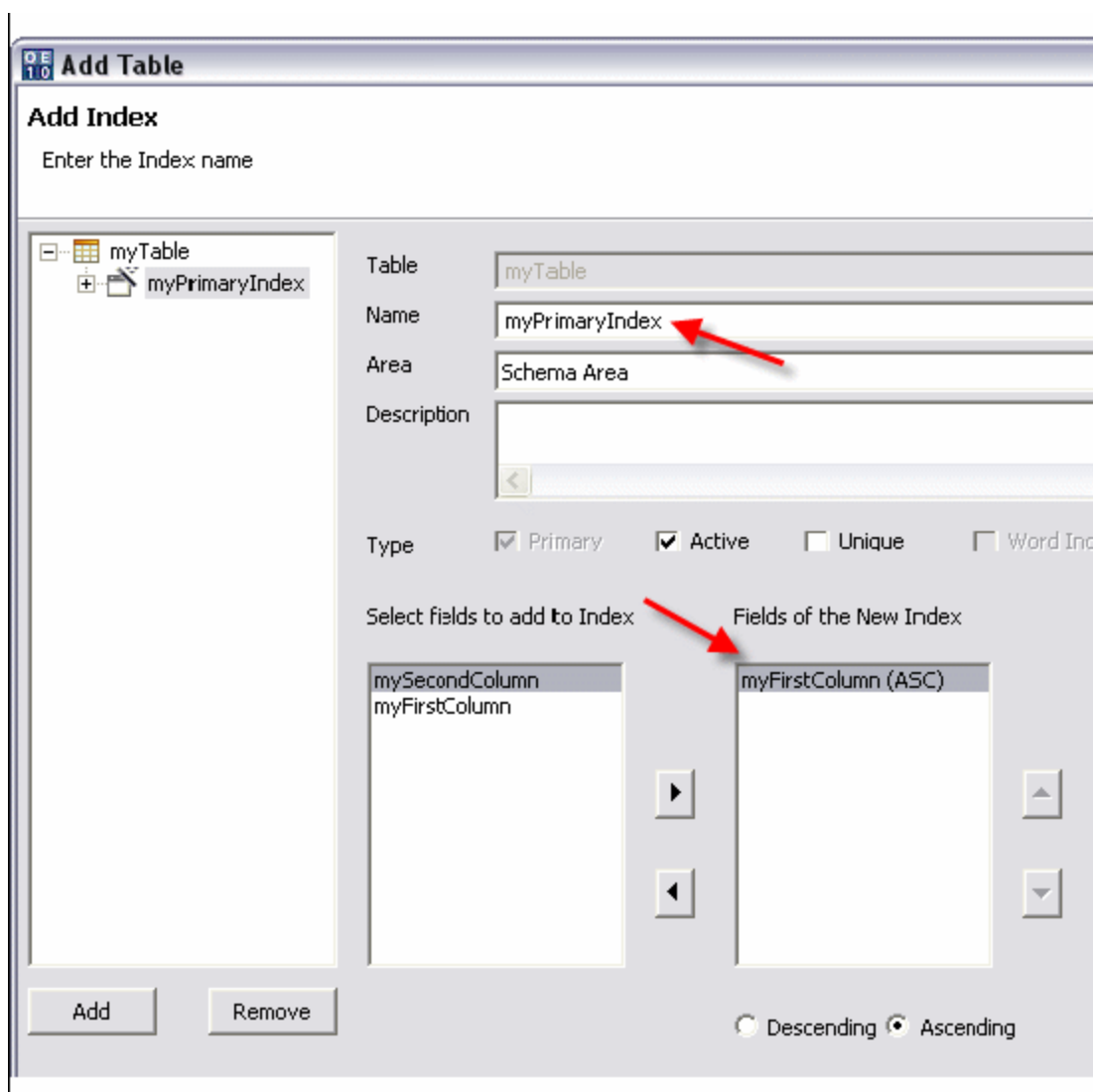
3. В следующем окне поля Name наберите myFirstColumn. Оставьте остальные поля заполненными по-умолчанию и нажмите кнопку Add для добавления поля.



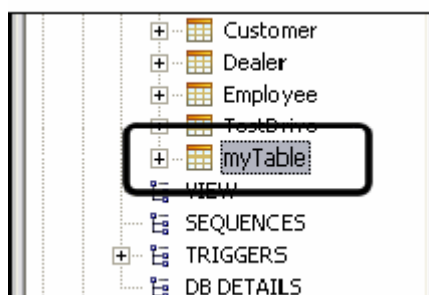
4. В поле Name измените New_Coloumn на значение mySecondColumn и тип с Data Type – на Integer. Нажмите Next.



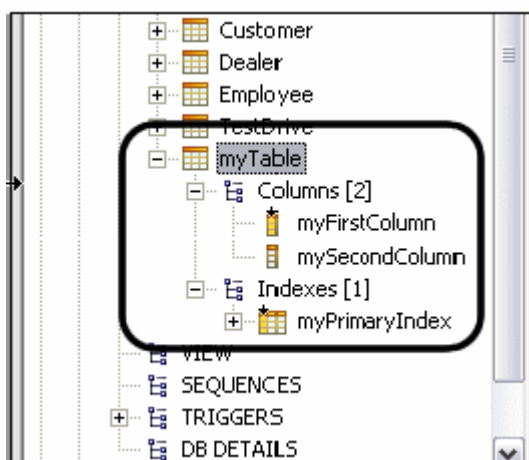
5. В открывшемся окне в поле Name наберите “myPrimaryIndex”. Выберите поле “myFirstColumn” и добавьте его из списка “Select fields to add to Index” в список “Fields of the New Index”, нажав на правую стрелку. Завершите действия кнопкой Finish.



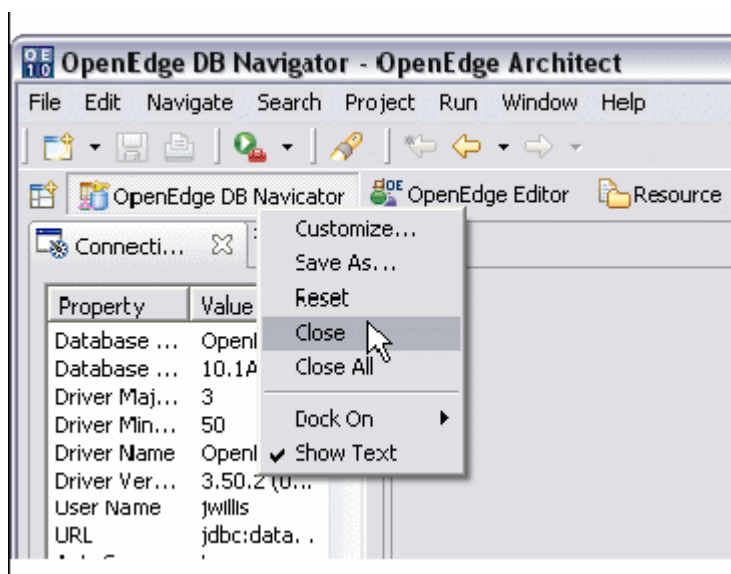
6. Таблица добавлена в базу, в окне DB Structure появился новый пункт myTable.



7. Раскройте новый пункт myTable, а также пункты полей и индексов, содержащихся в нем. Вы увидите добавленные вами новые элементы схемы.



8. Закройте Database Navigator Perspective, нажав правой кнопкой мыши по метке с названием окна на панели инструментов и выбрав опцию Close из контекстного меню.



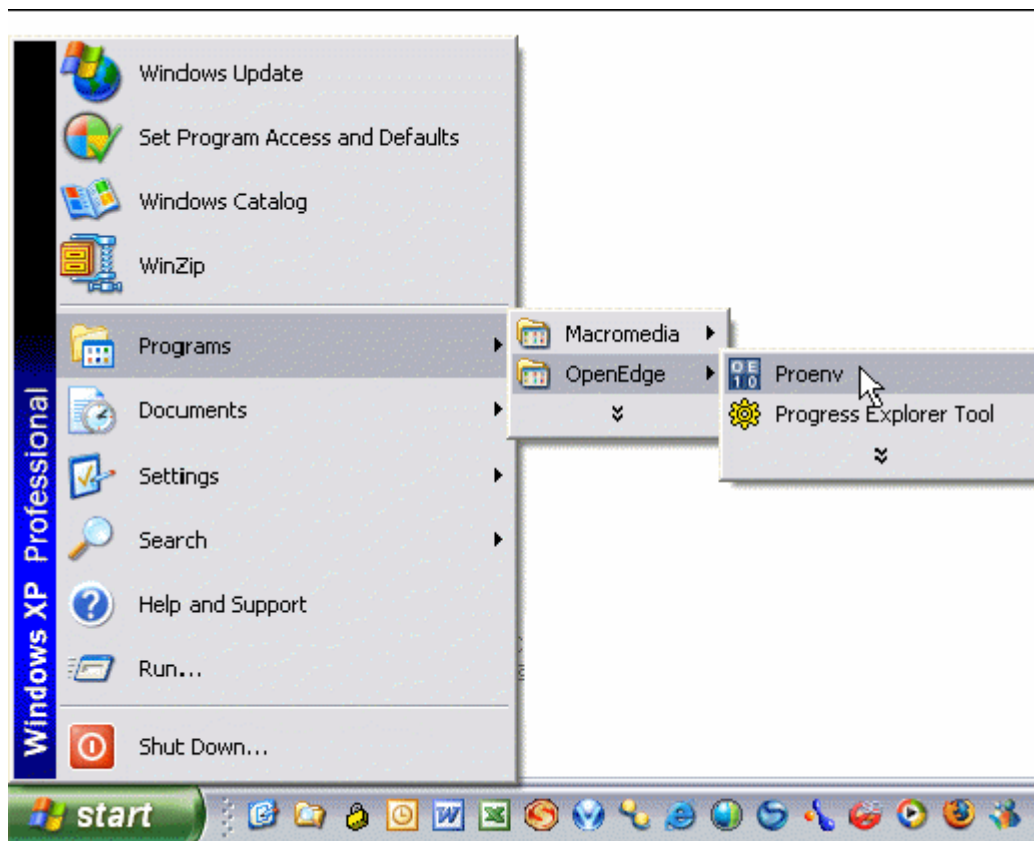
Программа отладки: OpenEdge Debugger

Настройка среды отладки

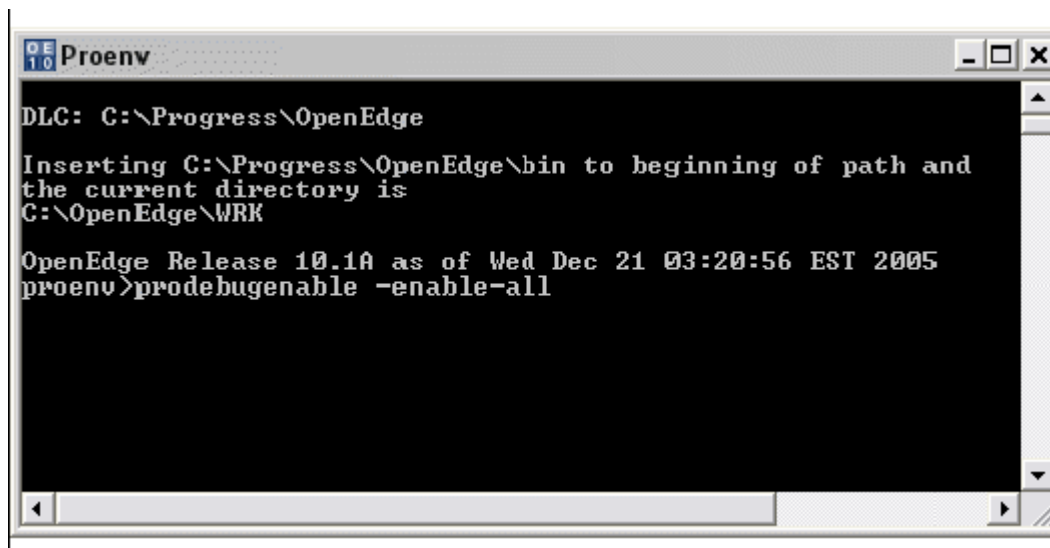
Разблокирование функции отладки

Функции отладки не доступны по-умолчанию. В среде OpenEdge их потребуется принудительно установить, выполнив следующее.

1. Закройте OpenEdge Architect.
2. Выберите из меню Start опции Progress->OpenEdge->Proenv.



3. В командной строке Proenv наберите “prodebugenable –enable-all”. Нажмите клавишу Enter. В OpenEdge станут доступными функции отладки.



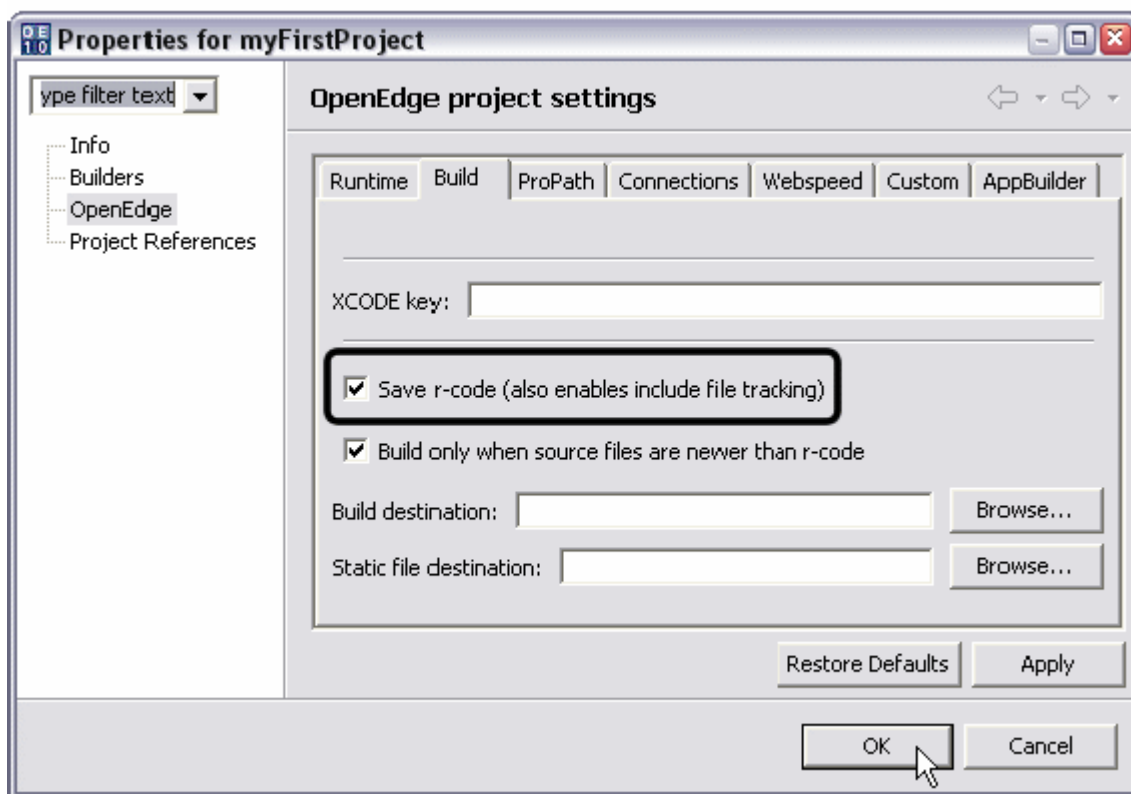
```
OE Proenv
DLC: C:\Progress\OpenEdge
Inserting C:\Progress\OpenEdge\bin to beginning of path and
the current directory is
C:\OpenEdge\WRK
OpenEdge Release 10.1A as of Wed Dec 21 03:20:56 EST 2005
proenv>prodebugenable -enable-all
```

4. Наберите “exit” и нажмите Enter для закрытия среды Proenv.
5. Откройте снова OpenEdge Architect. Если потребовалось объявить рабочую область Workspace, оставьте значение по умолчанию и нажмите кнопку ОК.

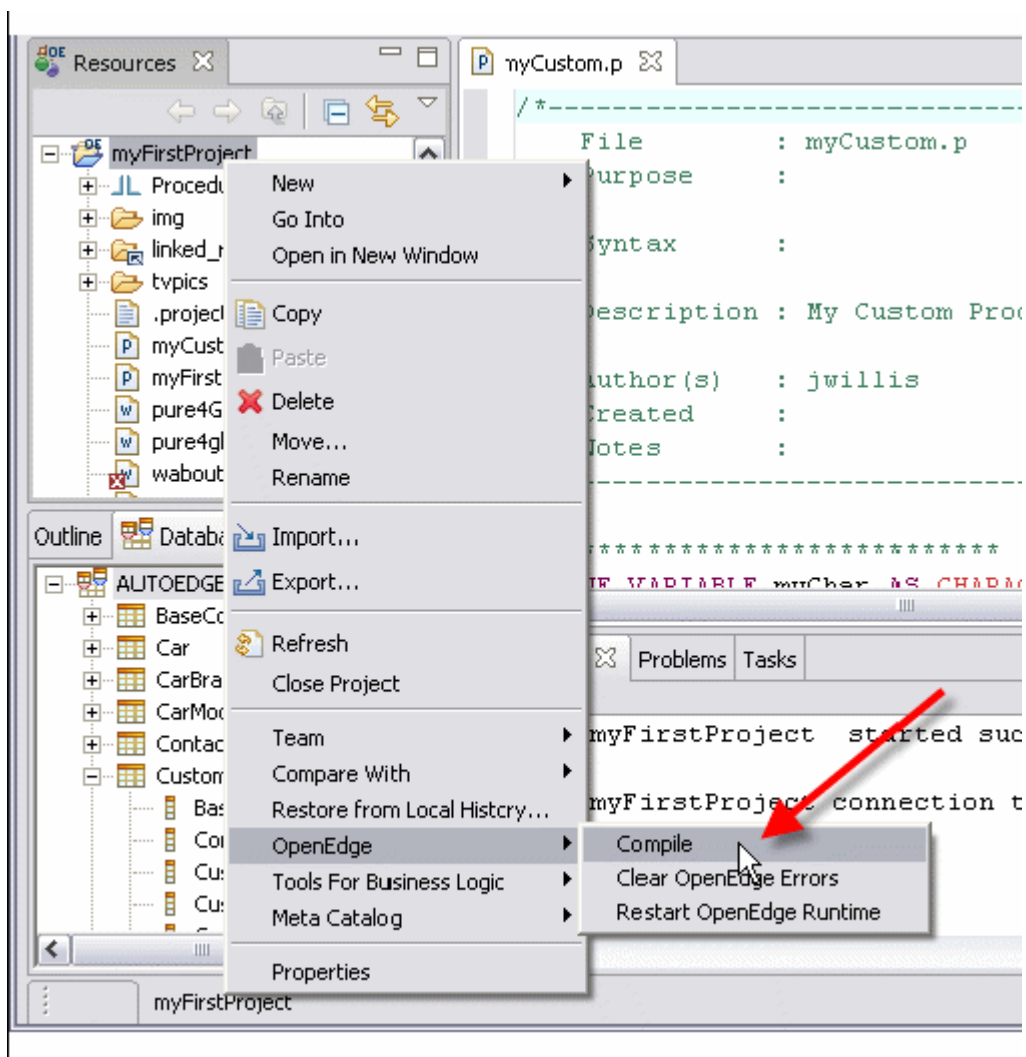
Сохранение и генерация R-кода

R-код – это скомпилированный файл ABL программы, он должен быть доступным для Debugger до начала процесса отладки. Каждый раз при сохранении программы Architect автоматически выстраивает R-код файл для нее. Следующие шаги описывают как установить эту опцию и сгенерировать r-код для файлов проекта, которые уже созданы и сохранены.

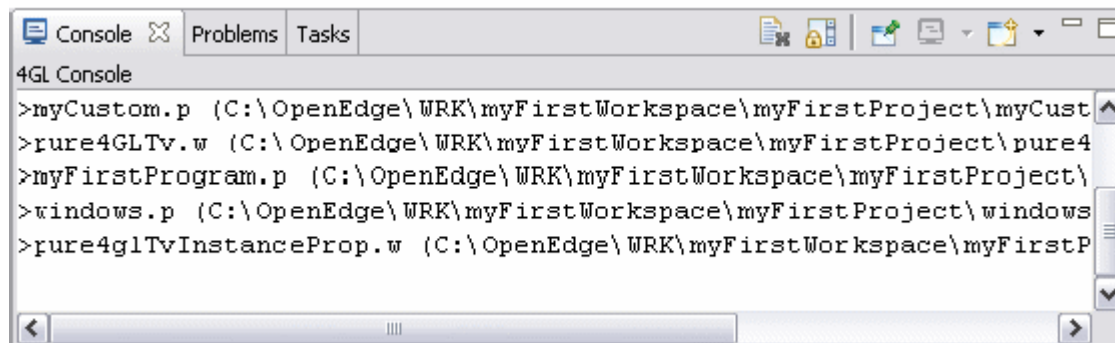
1. Чтобы сделать доступной опцию R-code Save/Reference Tracking, выберите Project > Properties Main Menu. В окне свойств Properties слева перейдите на OpenEdge, справа – на вкладку Build. Отметьте опцию Save r-code, нажмите кнопку ОК.



2. Для создания r-кода для уже существующей программы, кликните правой кнопкой мыши по пункту myFirstProject в окне Resources и выберите OpenEdge -> Compile.



- По окончании компиляции сформируется список сообщений в окне Console указывающий на обработанные программы. Пройгнорируйте ошибки, которые могли возникнуть в процессе компиляции, так как необходимо прежде завершить процесс написания кода для некоторых программ проекта.

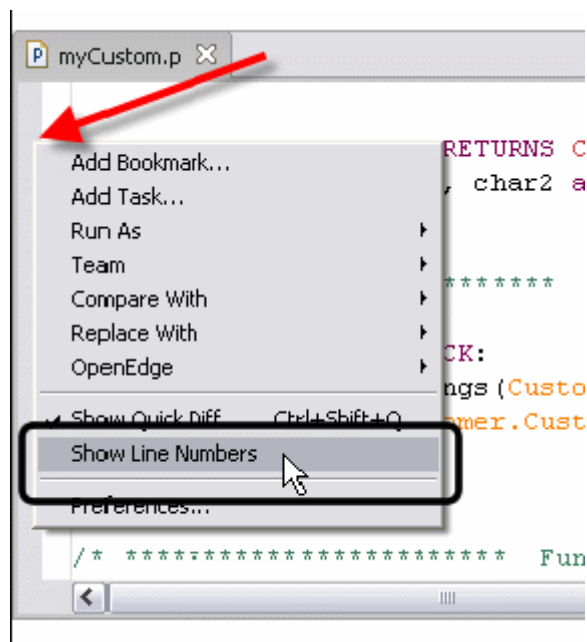


Отладка AVL приложения

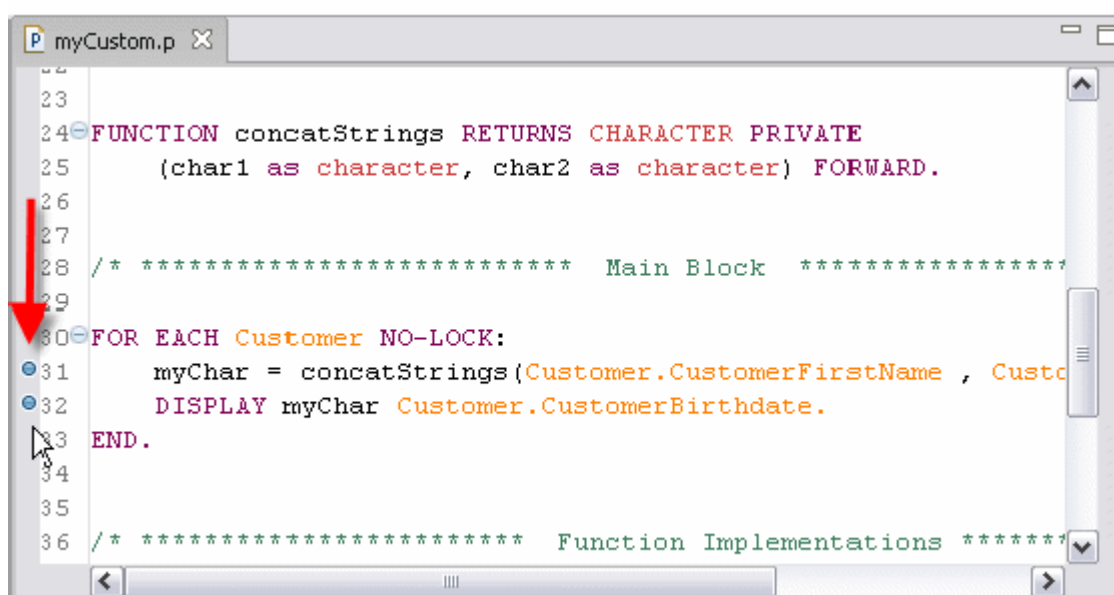
Использование отладчика Debugger

Для проверки функций Architect Debugger на примере программы myCustom.p выполните следующее.

1. Программа myCustom.p все еще должна быть открыта в редакторе OpenEdge Editor. В ином случае, кликните дважды по myCustom.p ниже пункта myFirstProject в окне Resources.
2. Правой кнопкой мыши кликните по Overview Ruler и выберите опцию Show Line Numbers. Это упростит нахождение кода при отладке.

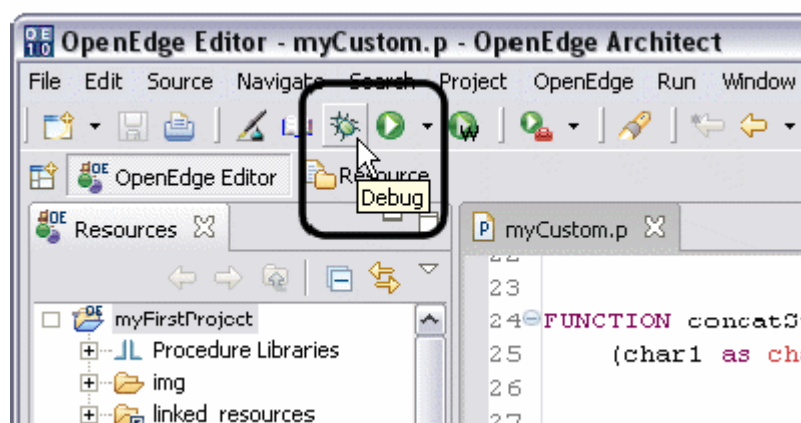


3. Кликните дважды по Overview Ruler, чтобы переместить точки прерывания на линии 31 и 32 как указано ниже. Убедитесь, чтобы синтаксис линий на рисунке совпадал с вашим. Иначе, запустите маркеры, начиная с кода "cFullName = concatStrings".

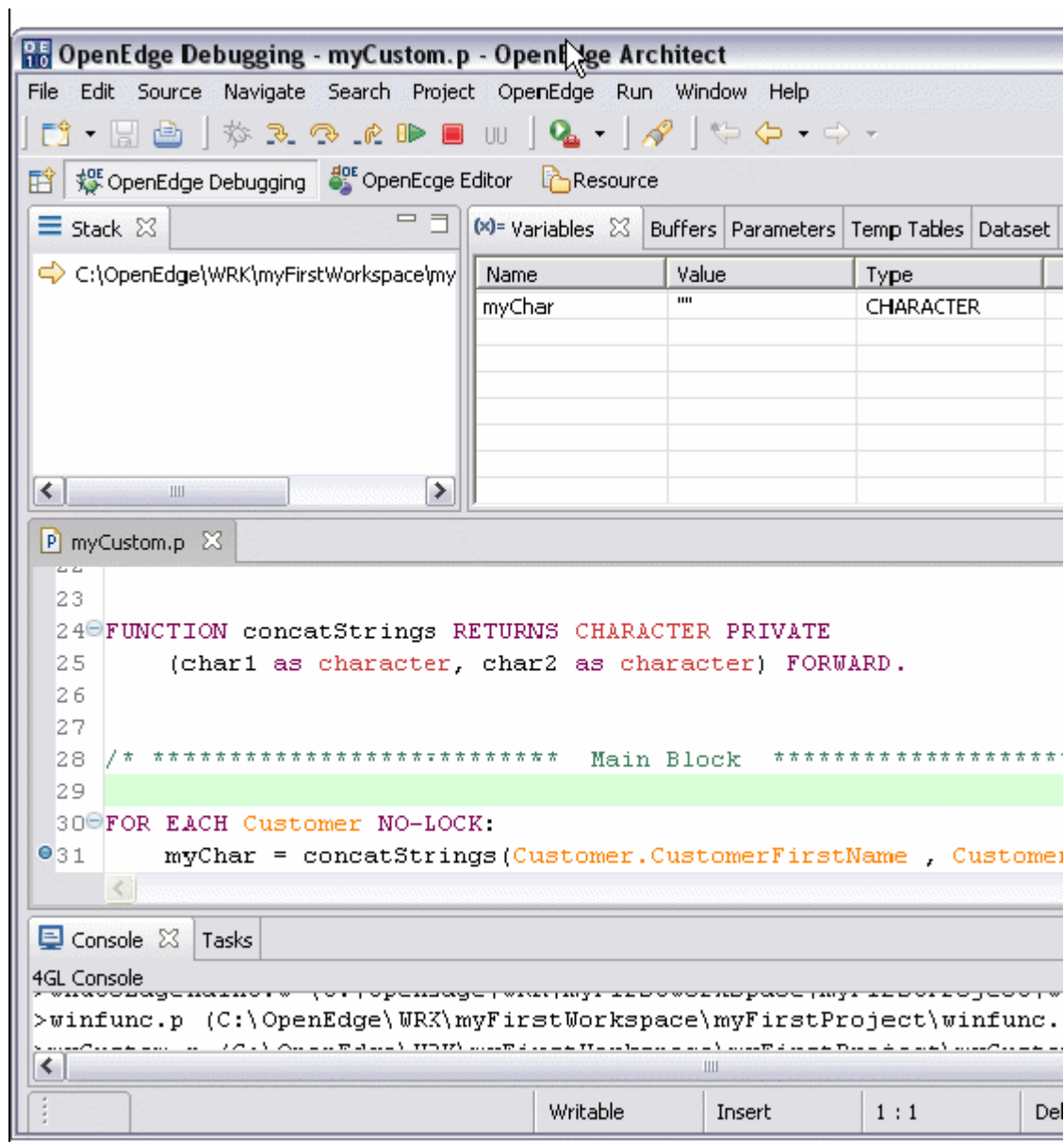


```
23
24 FUNCTION concatStrings RETURNS CHARACTER PRIVATE
25     (char1 as character, char2 as character) FORWARD.
26
27
28 /* ***** Main Block *****
29
30 FOR EACH Customer NO-LOCK:
31     myChar = concatStrings(Customer.CustomerFirstName , Custo
32     DISPLAY myChar Customer.CustomerBirthdate.
33 END.
34
35
36 /* ***** Function Implementations *****
```

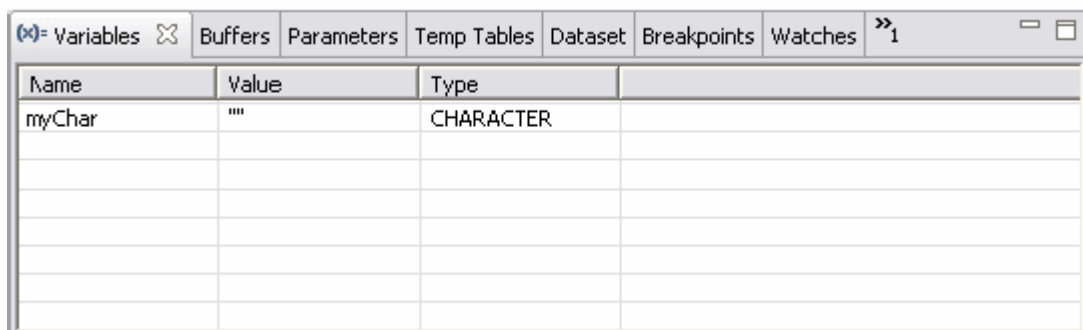
4. Нажмите кнопку Debug в главном меню.



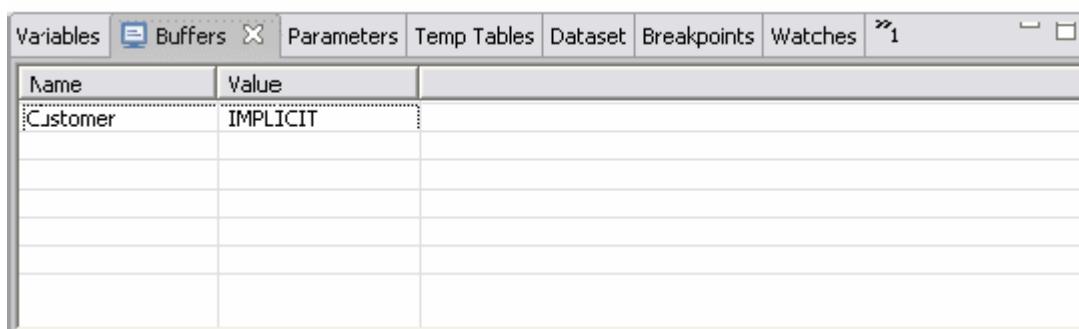
5. Откроется окно OpenEdge Debugging и начнется исполнение программы.



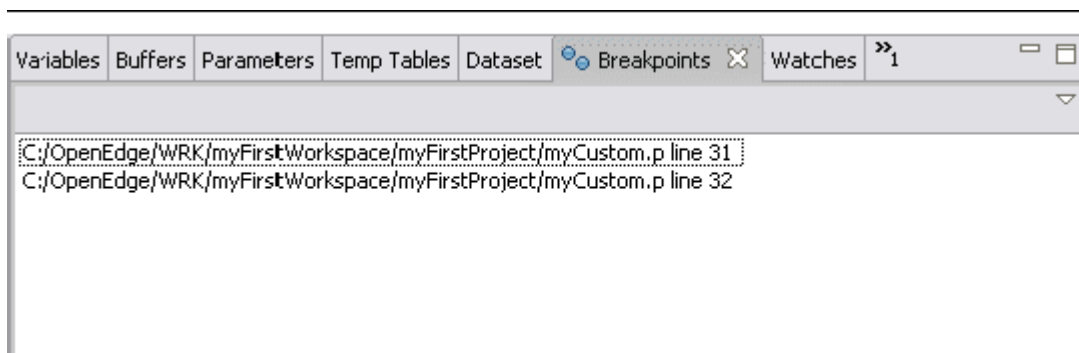
6. В окне Variables вы увидите переменную myChar с пустым текущим значением.



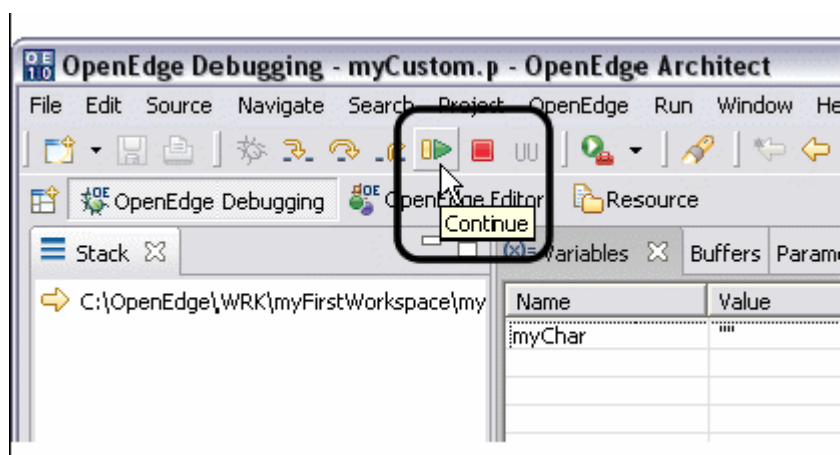
7. Перейдите к окну Buffer. Обратите внимание, что буфер Customer был сформирован, ссылаясь на буфер, использовавшийся в операторе For Each на линии 30.



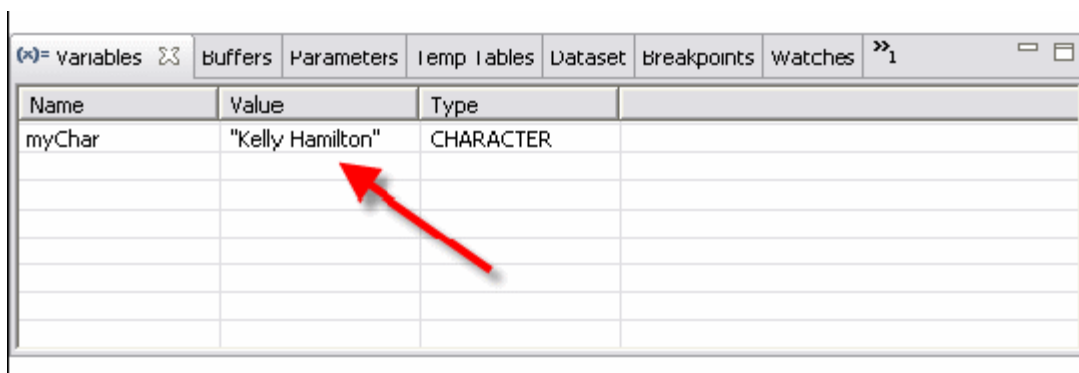
8. Перейдите на вкладку Breakpoints. Вы увидите две контрольные точки, определенные в списке.



9. Вернитесь на вкладку Variables.
10. Нажмите кнопку Continue в главном меню.

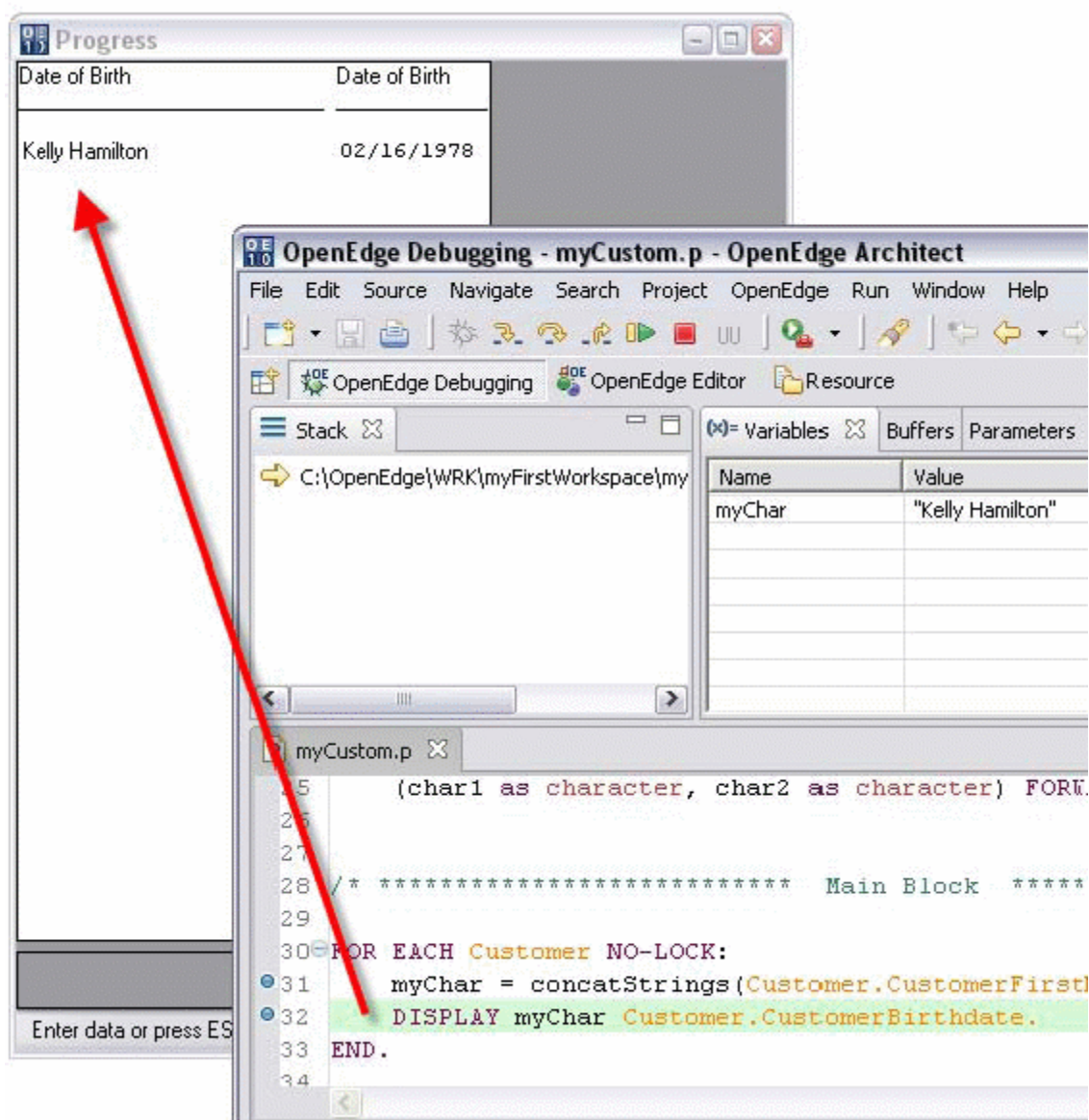


11. Во вкладке Variables появилось значение для myChar.



Name	Value	Type
myChar	"Kelly Hamilton"	CHARACTER

12. Нажмите кнопку Continue. Подкрашенная строка отладки - зеленый цвет линии 32 – указывает на последнюю линию, обработанную отладчиком. Результат применения оператора Display в линии 32 отразился в окне приложения Progress.



Progress

Date of Birth	Date of Birth
Kelly Hamilton	02/16/1978

OpenEdge Debugging - myCustom.p - OpenEdge Architect

Stack

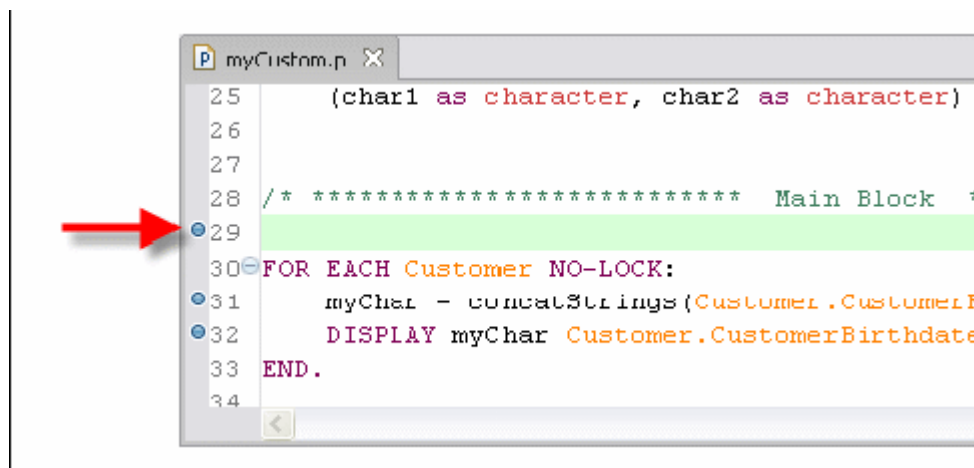
C:\OpenEdge\WRK\myFirstWorkspace\my

Variables

Name	Value
myChar	"Kelly Hamilton"

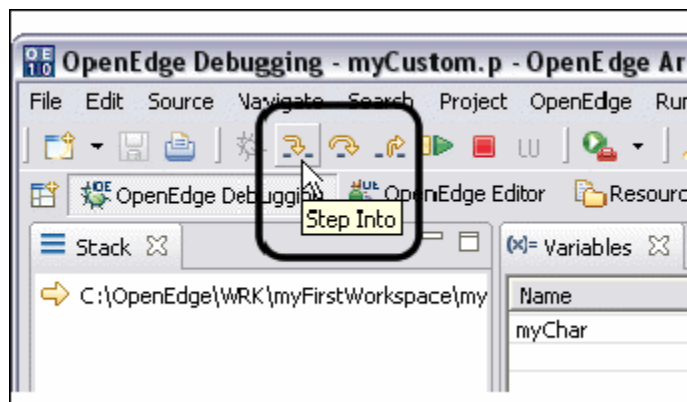
```
myCustom.p
25 (char1 as character, char2 as character) FORU
26
27
28 /* ***** Main Block *****
29
30 FOR EACH Customer NO-LOCK:
31 myChar = concatStrings(Customer.CustomerFirst
32 DISPLAY myChar Customer.CustomerBirthdate.
33 END.
34
```

13. Нажмите кнопку Continue. Отладчик продолжит исполнение и завершит For Each блок. Он переместит подкрашенную строку на свободное место над кодом, выше оператора For Each, показывая, что процесс стартует сначала. Зеленый маркер подсвечивает линию 29 и указывает на начало отлаженного цикла.

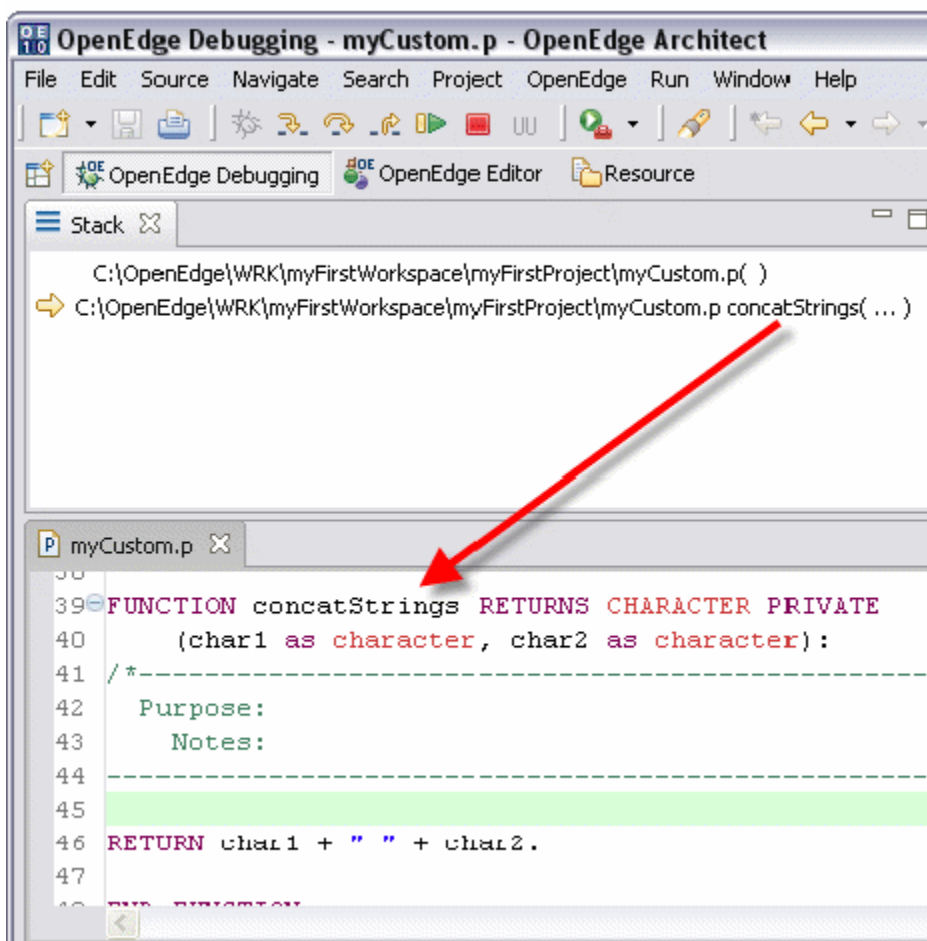


```
myCustom.p
25 (char1 as character, char2 as character)
26
27
28 /* ***** Main Block *****
29
30 FOR EACH Customer NO-LOCK:
31   myChar - concatStrings(Customer.CustomerID
32   DISPLAY myChar Customer.CustomerBirthdate
33 END.
34
```

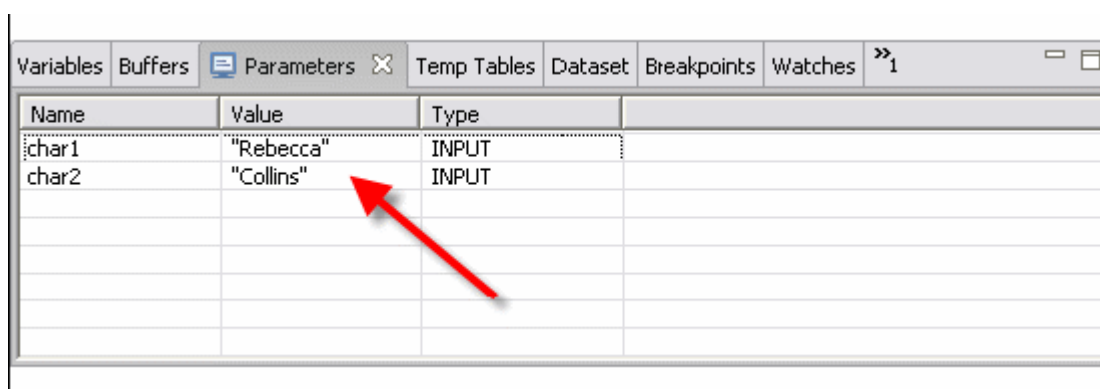
14. Кликните дважды по кнопке Step Into.



15. Это позволит вам перейти к коду функции concatString. Проверьте, что в окне Stack отображаемая функция concatString является следующей по списку.

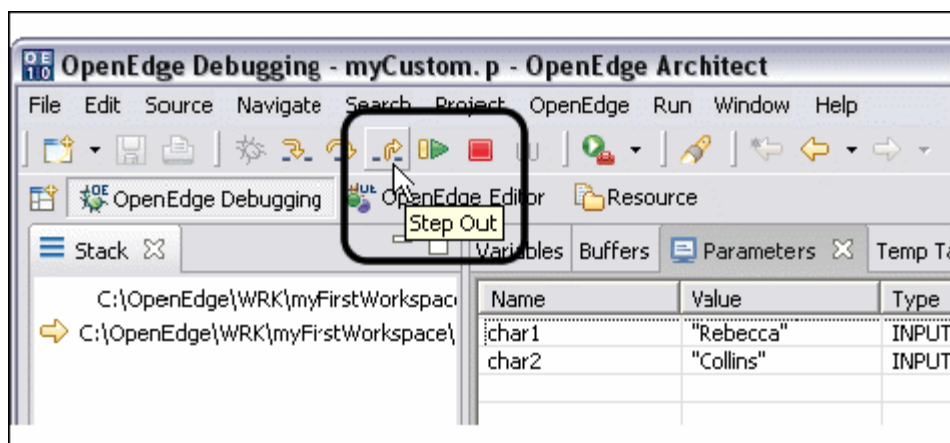


16. Перейдите во вкладку Parameters. Вы увидите два значения параметра, которые были переданы функции `concatString`.

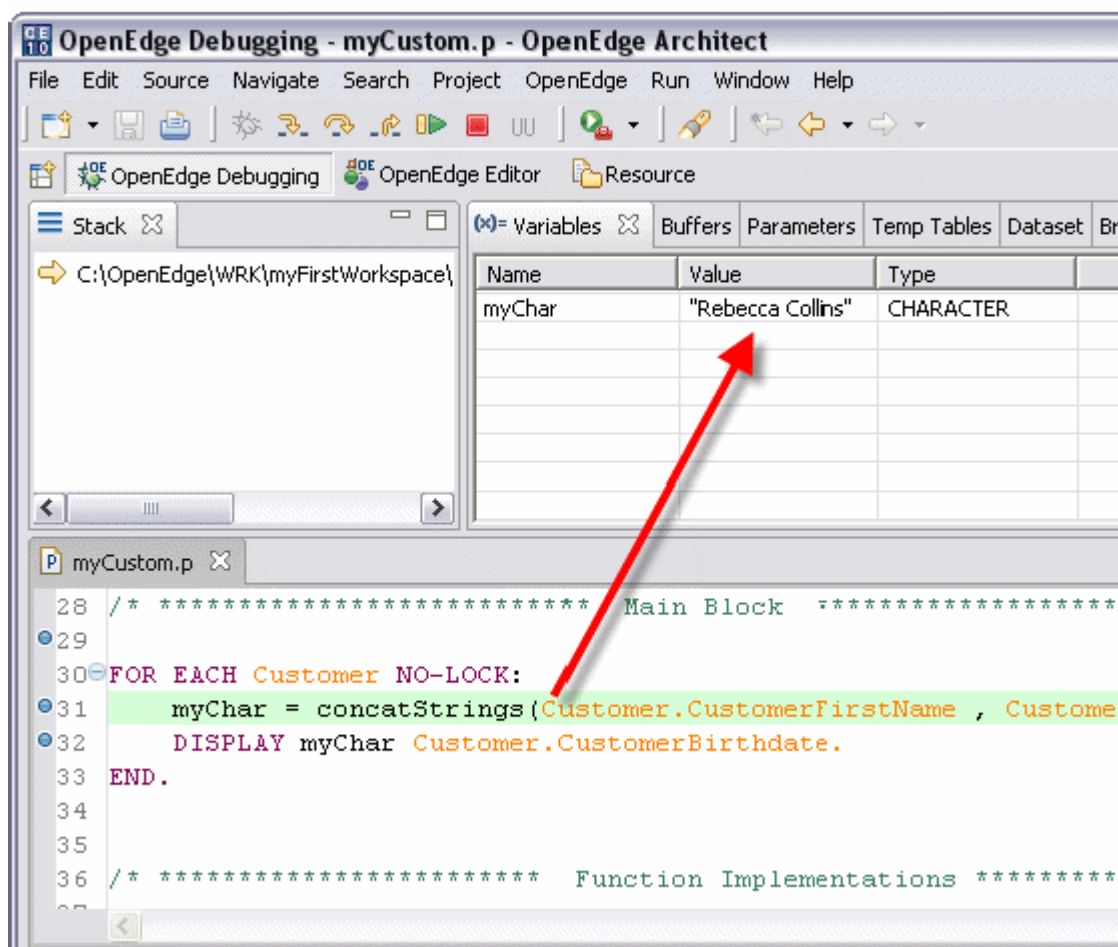


17. Вернитесь в окно Variables.

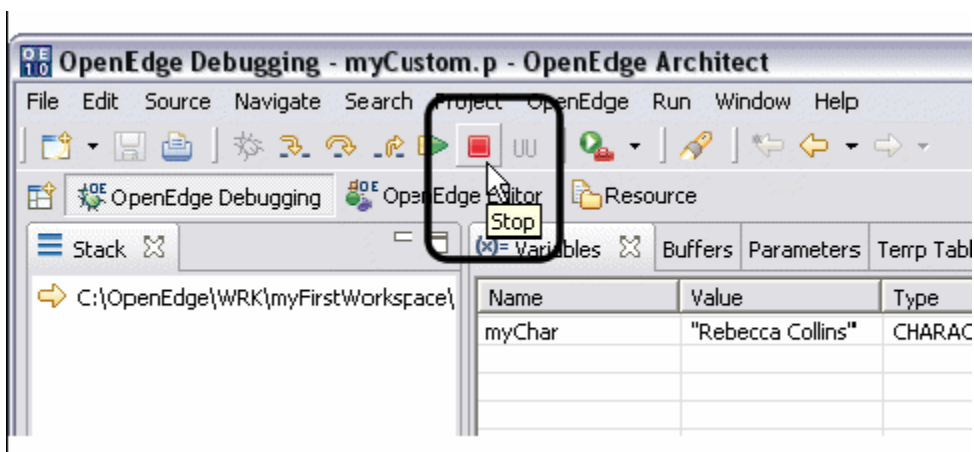
18. Нажмите клавишу Step Out для выхода из кода функции.



19. Убедитесь, что в результате действия функции, значение переменной myChar изменилось на ожидаемое.



20. Нажмите дважды кнопку Stop для остановки процесса отладки и очищения окна Results.



21. Закройте окно OpenEdge Debugging.

22. Закройте файл myCustom.p.

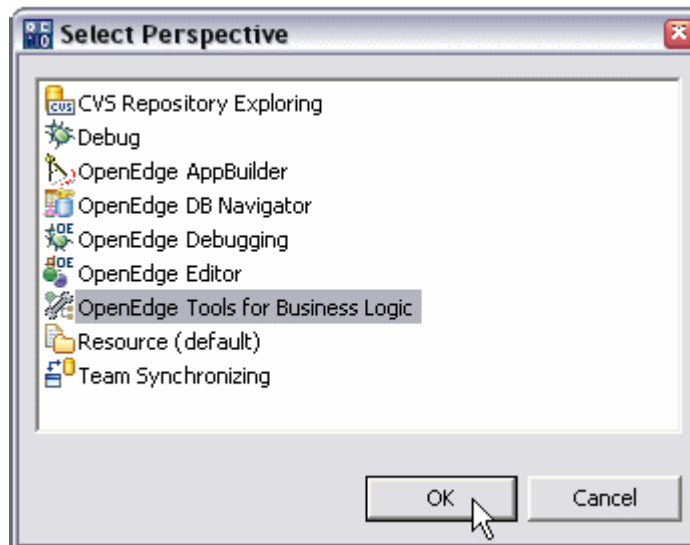
Инструменты для бизнес-логики

Создание компонентной диаграммы

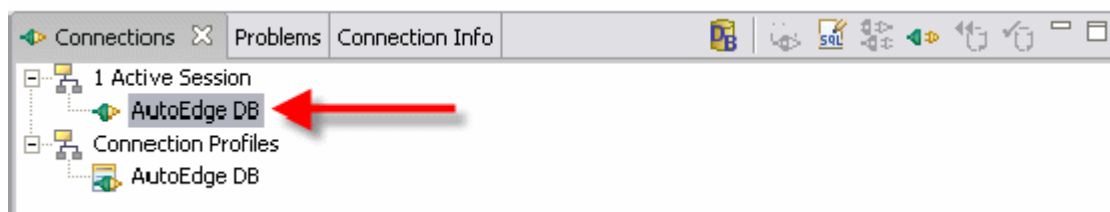
Создание файла диаграммы и ProDataSet компонента

Выполните следующие шаги для создания файла диаграммы, в которой будет наглядно представлена информация с помощью ProDataSet компонента о покупателях, тесте по вождению и машинах.

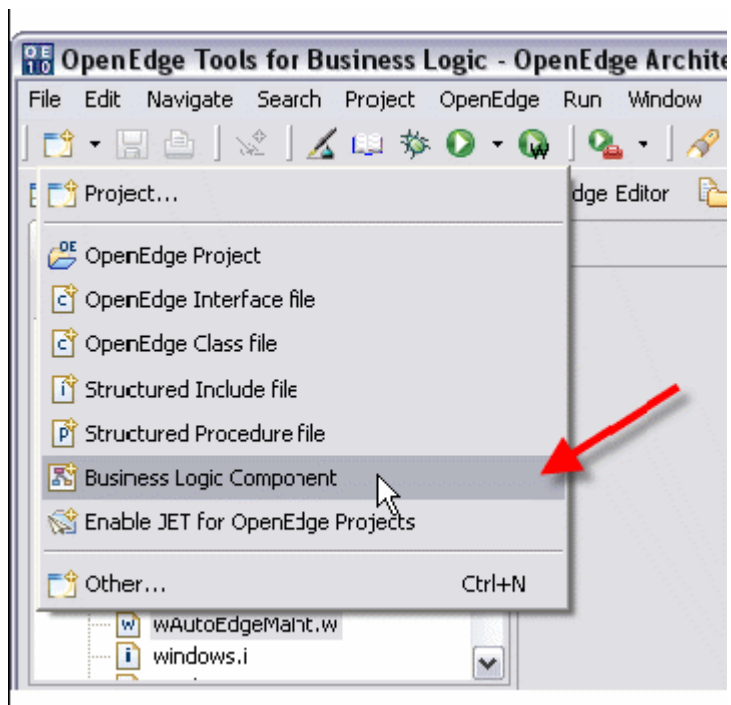
1. Вызовите Tools for Business Logic Perspective, открыв Window -> Open Perspective -> Other Main Menu. Далее выберите пункт OpenEdge Tools for Business Logic и нажмите OK.



2. Так как Tools for Business Logic использует SQL соединение для доступа к схеме базы данных, убедитесь, что в окне Connection соединение AutoEdge DB активно.

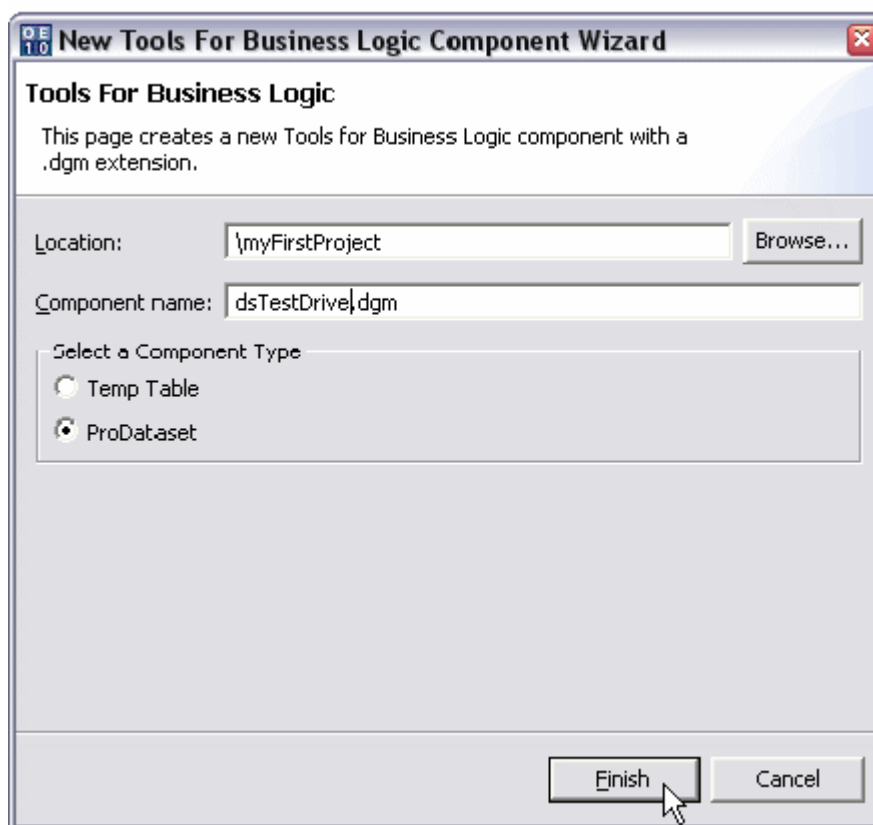


3. Создайте новый компонент, нажав на стрелку, направленную вниз, возле кнопки New и выбрав опцию Business Logic Component.

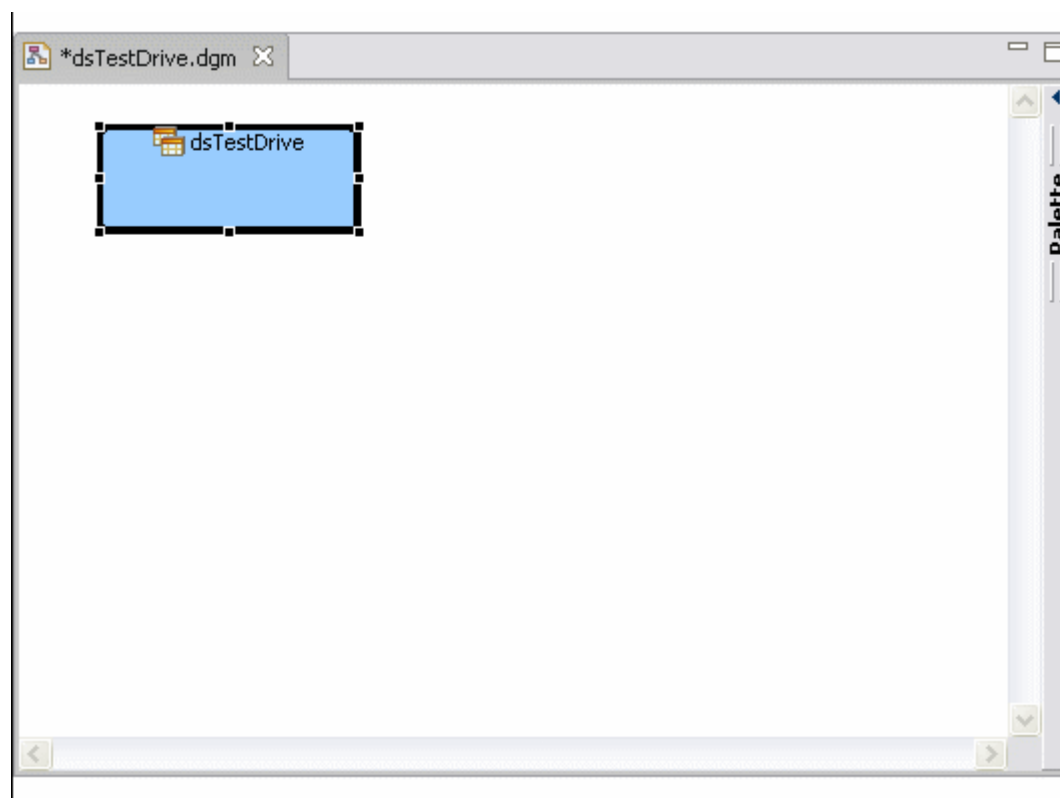


4. В открывшемся окне мастера новых компонентов, установите следующие значения:
 - Location: \myFirstProject
 - Component Name: dsTestDrive.dgm
 - Component Type: ProDataset

Закройте окно, нажав кнопку Finish.



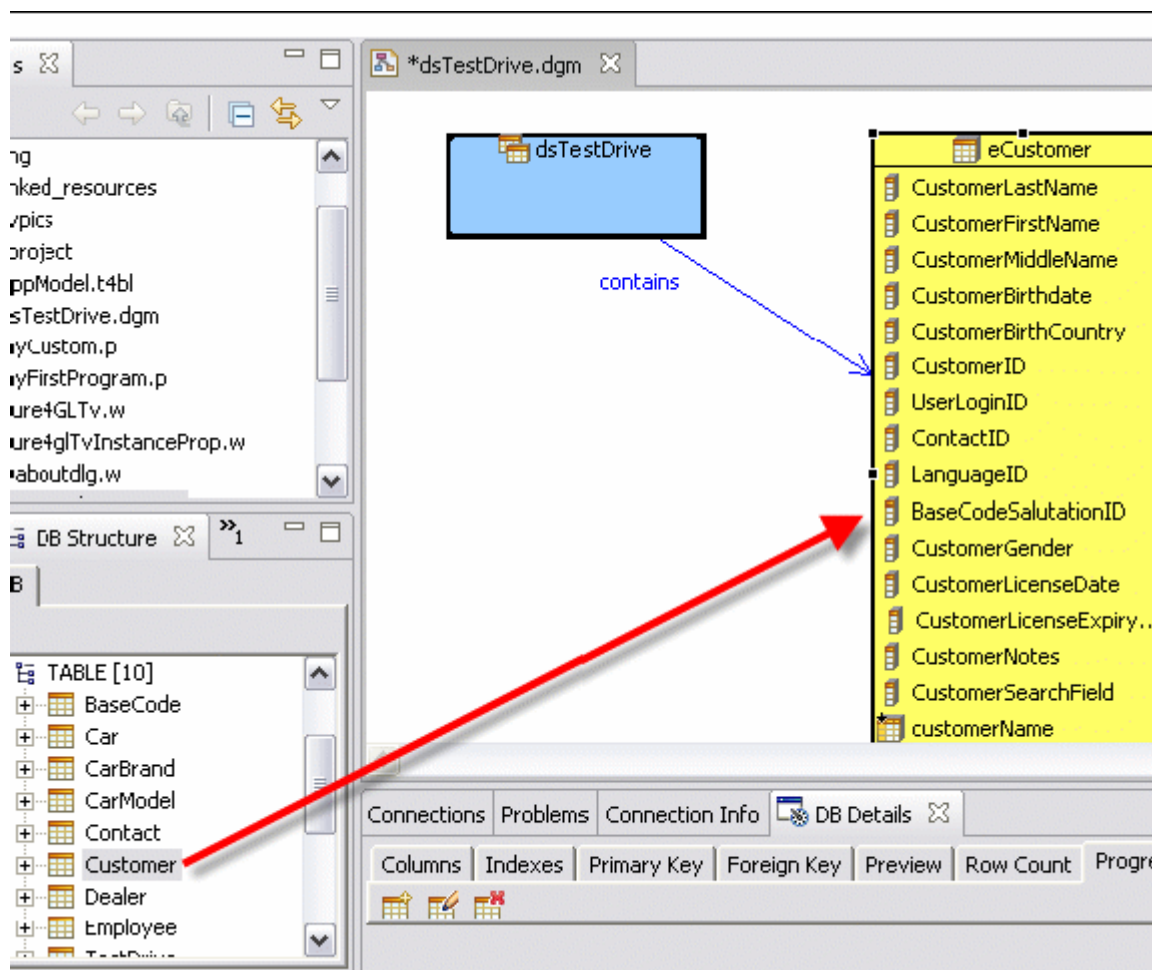
5. Создан файл диаграммы. Открылось окно, где в области для размещения элементов дизайна находится ProDataset компонент с названием dsTestDrive.



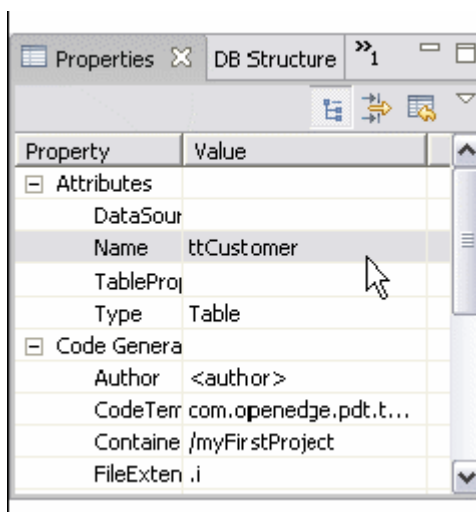
Создание временной таблицы с информацией о структуре схемы базы данных

Существует много случаев, когда создается и повторно используется структура компонентов диаграммы. Один быстрый способ для создания компонента временной таблицы – это копирование структуры схемы с уже существующей базы данных. Для создания компонента временной таблицы диаграммы dsTestDrive с помощью структуры базы данных выполните следующие действия:

1. В окне DB Structure раскройте пункты Database -> Pub -> Table и найдите таблицу Customer.
2. Удерживая мышью, переместите таблицу Customer в область для размещения компонентов. Будет создан новый компонент временной таблицы Customer.



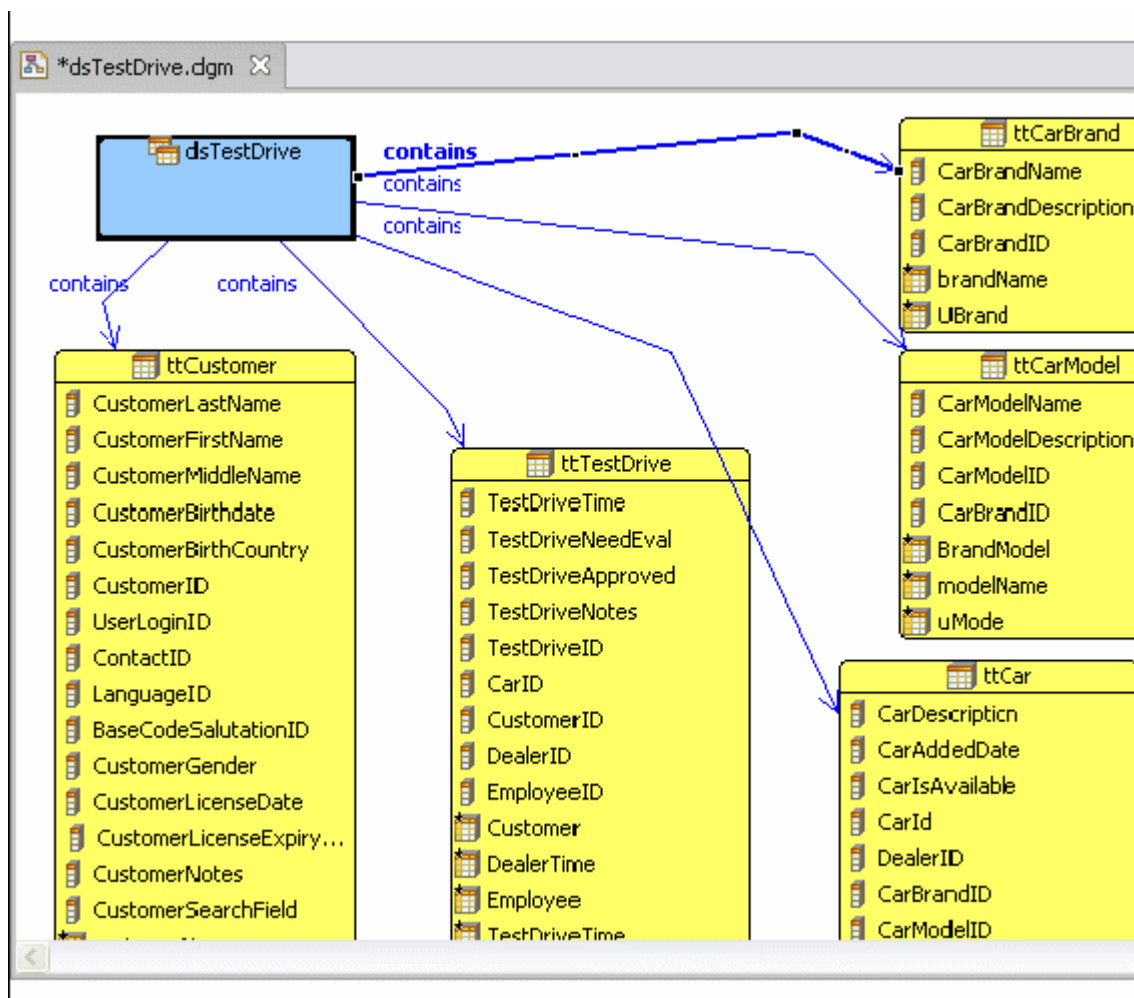
3. Выберите новый компонент временной таблицы Customer и вкладку Properties. Переместитесь к Name Property ниже пункта Attributes. Кликните по ячейке со значением Value и измените его на "ttCustomer".



4. Вернитесь в окно DB Structure и повторите шаги 2 и 3 для следующих таблиц.

Table	Temp-Table Name
TestDrive	ttTestDrive
Car	ttCar
CarBrand	ttCarBrand
CarModel	ttCarModel

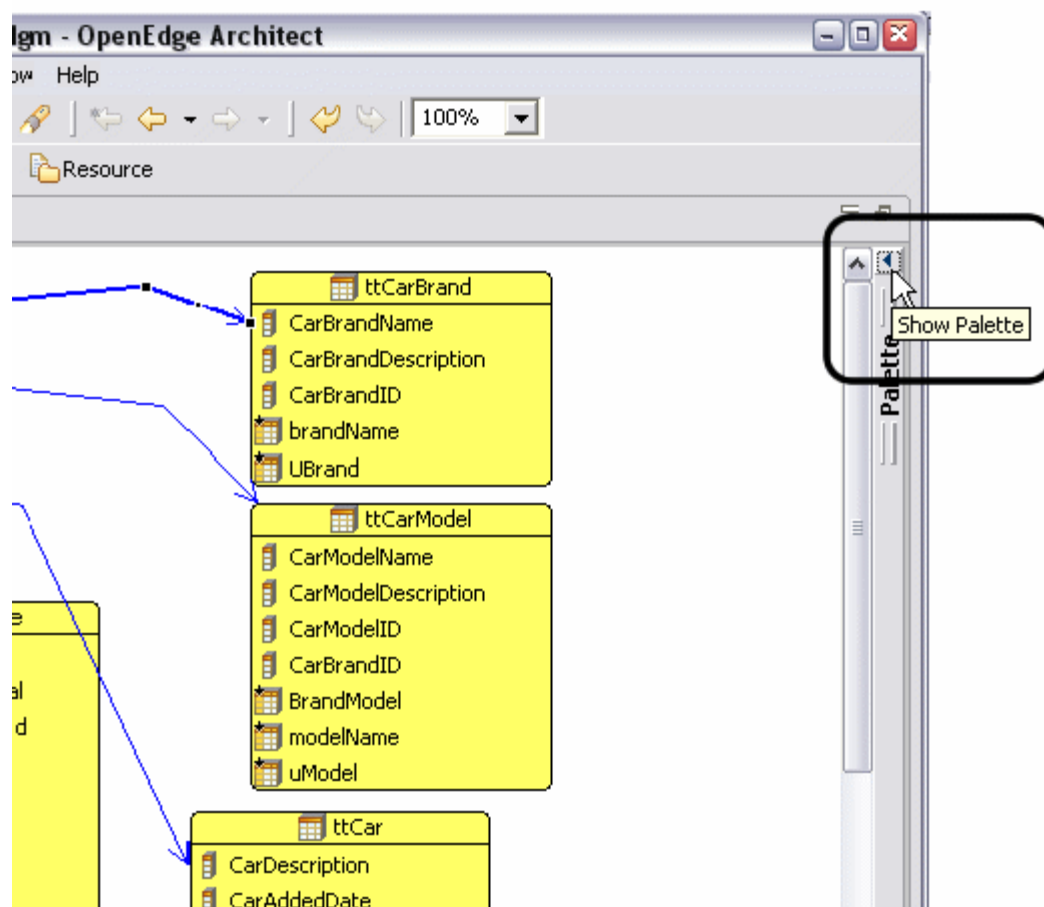
5. Компоненты временной таблицы на диаграмме располагаются примерно так, как показано на рисунке ниже. Вы можете развернуть диаграмму для просмотра на всю видимую область, кликнув дважды по ее названию.



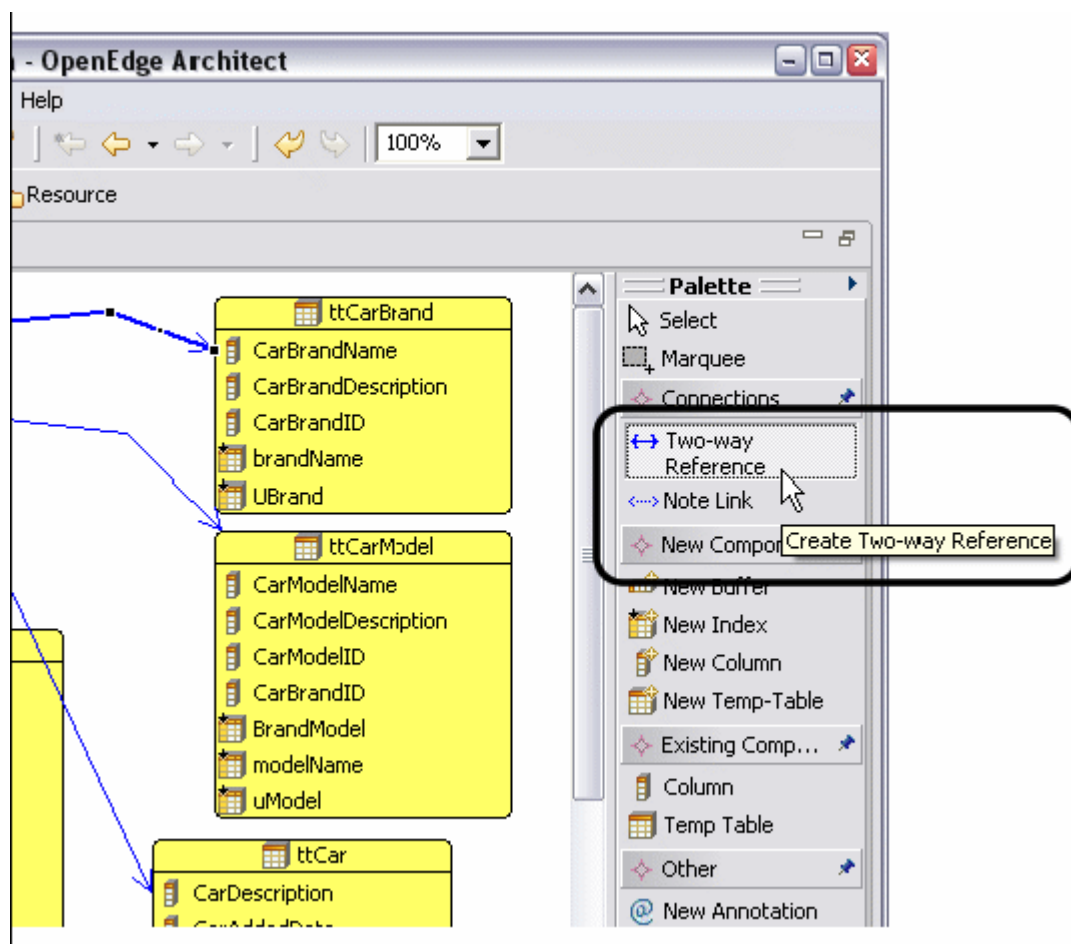
Добавление связей отношений

Для добавления отношений между компонентами временных таблиц в диаграмме dsTestDrive, выполните следующее.

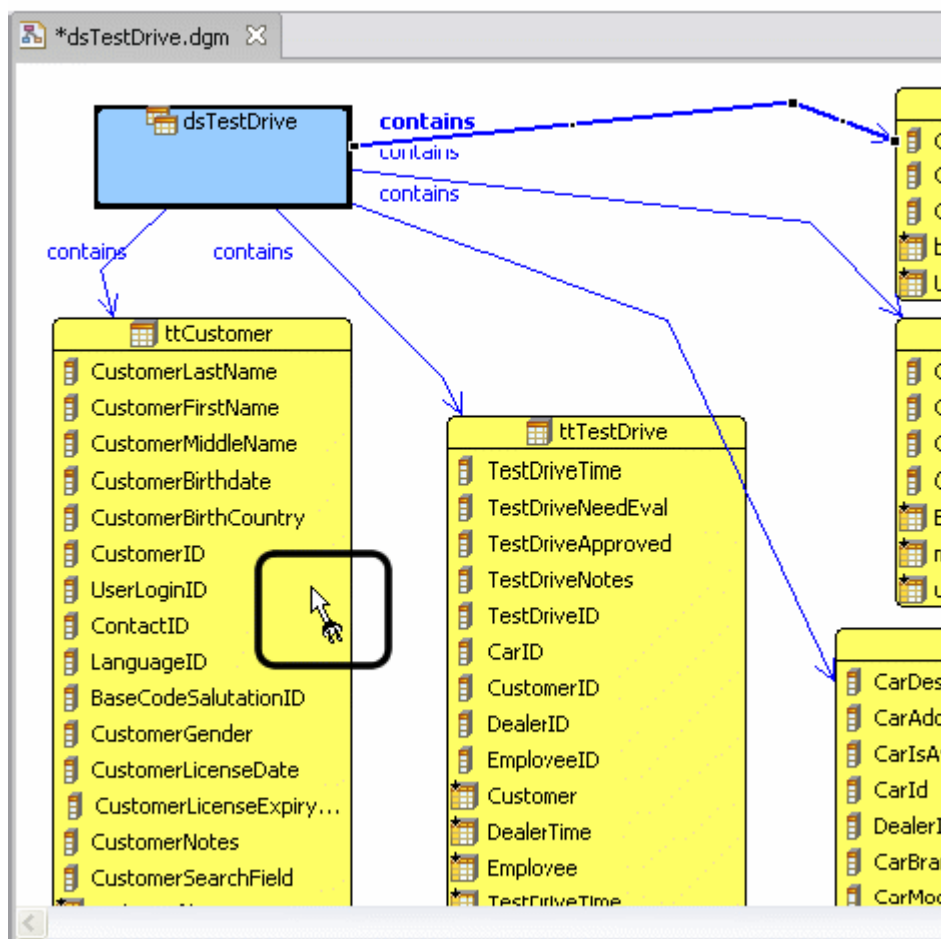
1. Откройте палитру Palette, выбрав левую стрелку в правой части области для размещения компонентов. Обратите внимание, что при наведении мыши на Palette всплывает надпись раскрытия палитры Show Palette.



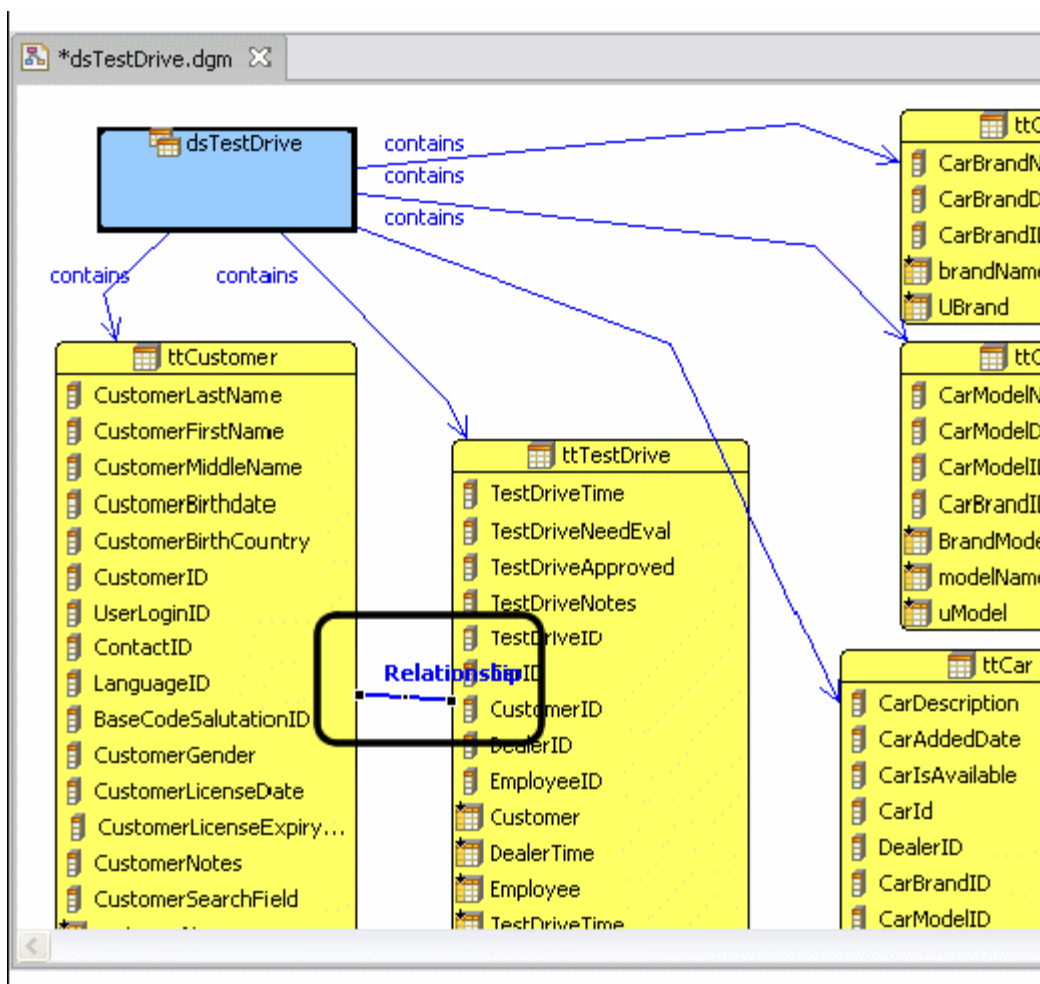
2. В палитре под меткой Connections выберите опцию Two-way Reference.



3. Наведите стрелку мыши на компонент ttCustomer.



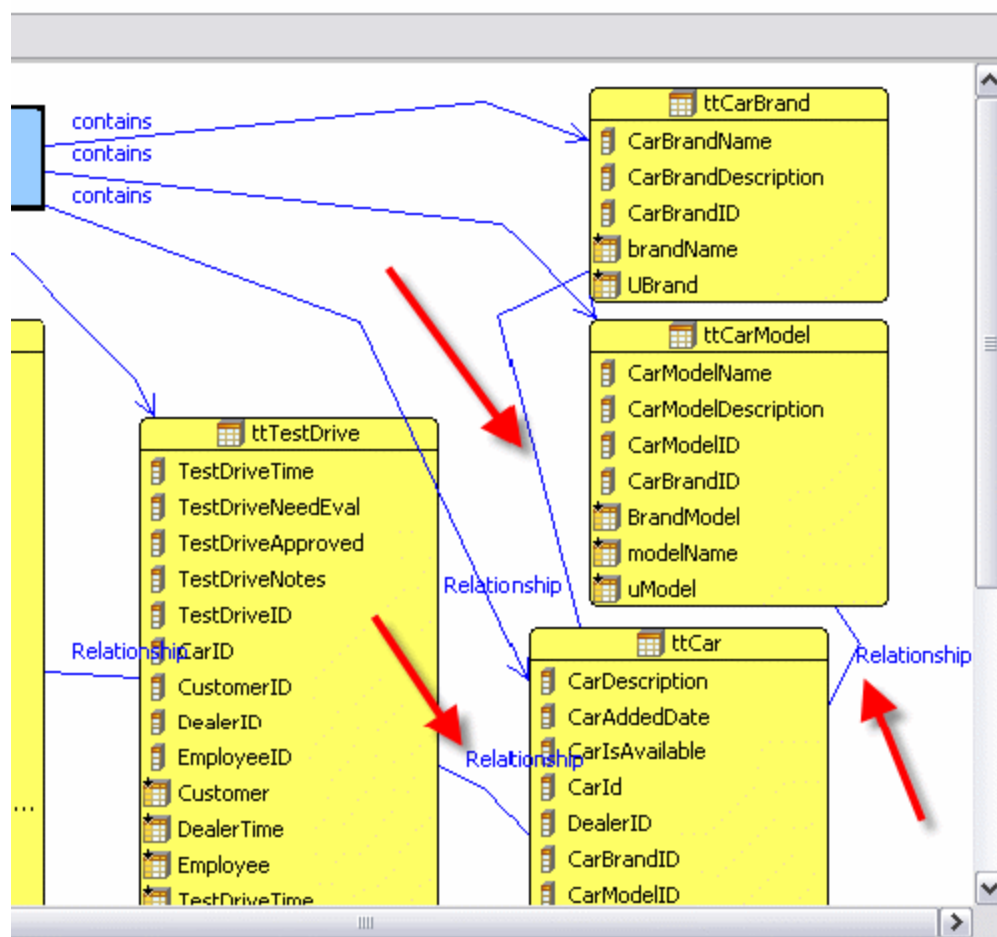
4. Кликните и протяните линию связи от ttCustomer (источника) к ttTestDrive (конечному элементу). Будет создано отношение.



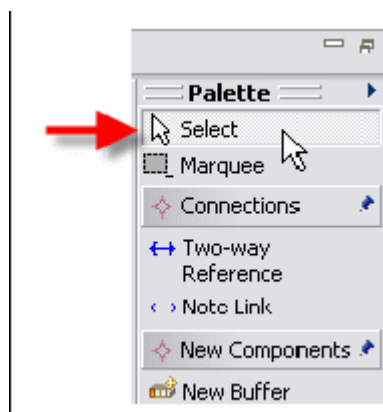
5. Повторите шаг 4 для аналогичного создания отношений между следующими компонентами:

Источник	Конечный элемент
ttTestDrive	ttCar
ttCar	ttCarBrand
ttCar	ttCarModel

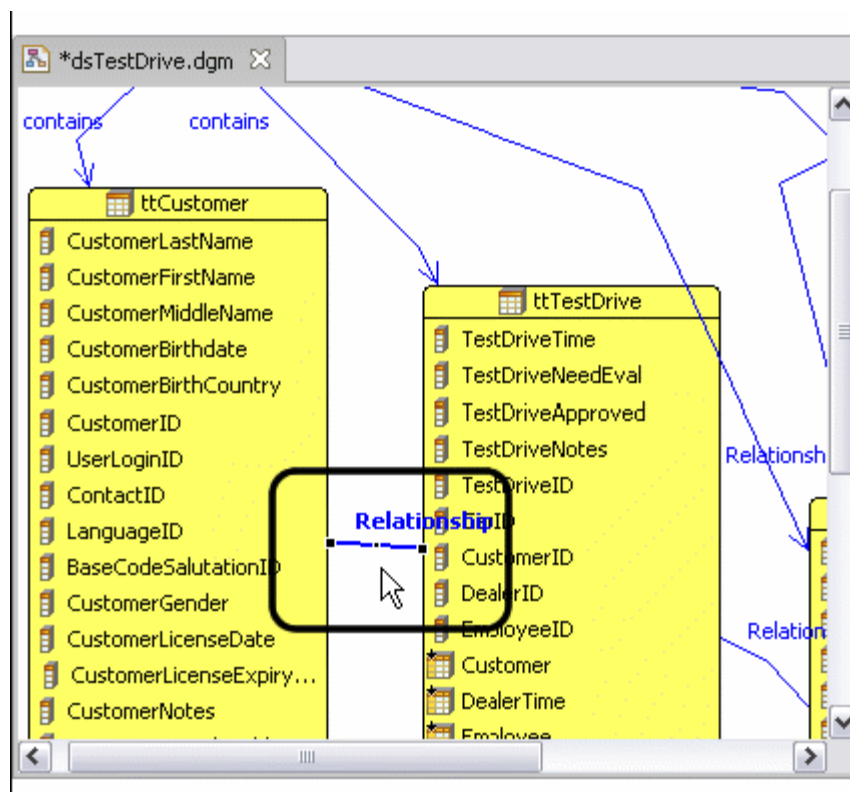
6. Диаграмма после установления связей будет выглядеть примерно следующим образом:



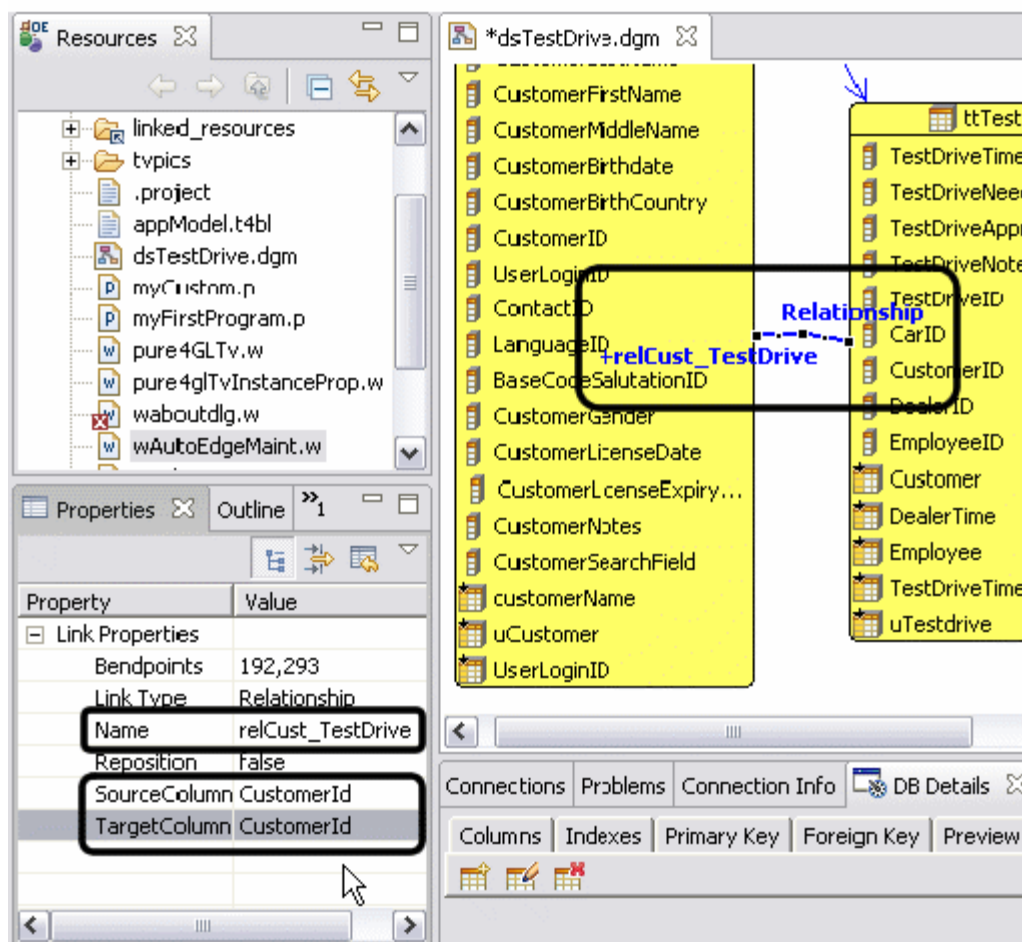
7. Вернитесь в палитру и выберите опцию Select.



8. Выберите линию связи между ttCustomer и ttTestDrive.



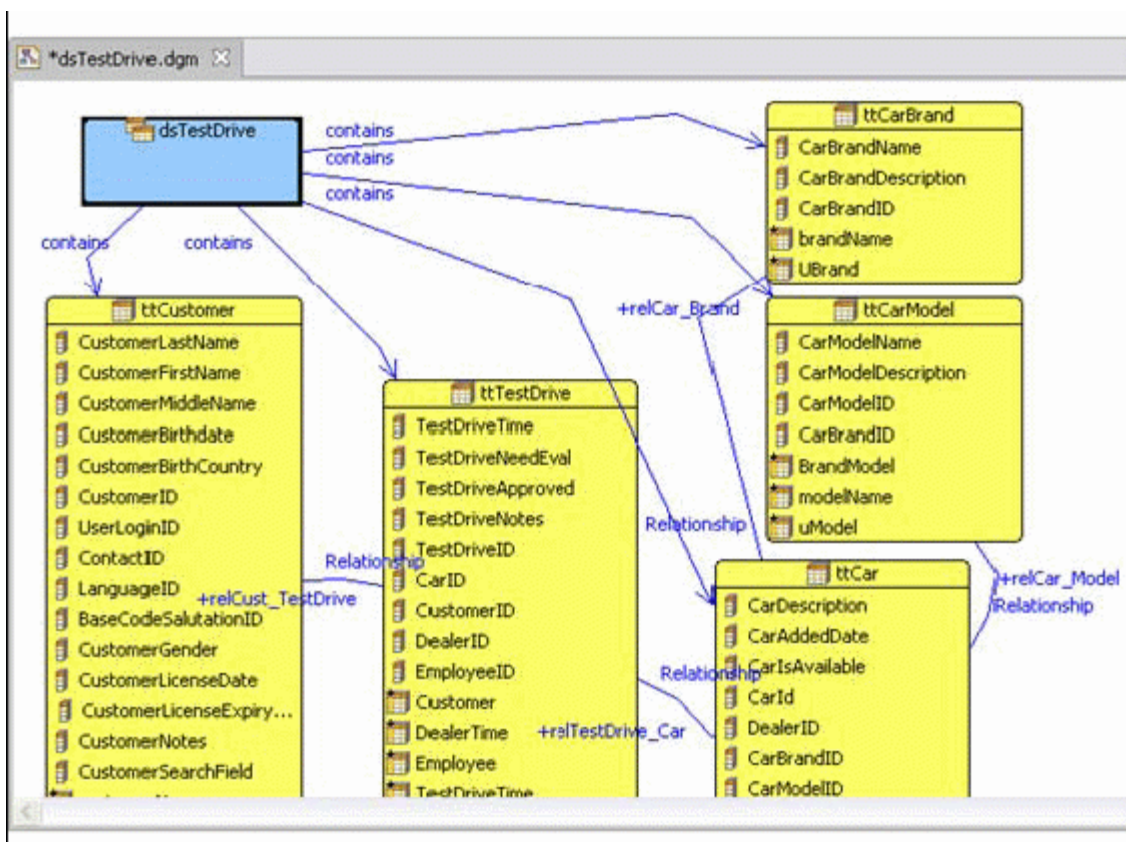
9. Выберите вкладку Properties. В поле Name впишите “relCust_TestDrive” и нажмите Enter. Результат отразится на диаграмме. Измените значения для SourceColumn и TargetColumn на CustomerID. Установится ключевое отношение по столбцу между двумя компонентами.



10. Повторите шаги 8 и 9, установив следующие свойства для отношений:

Source->Target	Name	Source/Target Column
ttTestDrive->ttCar	relTestDrive_Car	CarID
ttCar->ttCarModel	relCar_Model	CARModelID
ttCar->ttCarBrand	relCar_Brand	CarBrandID

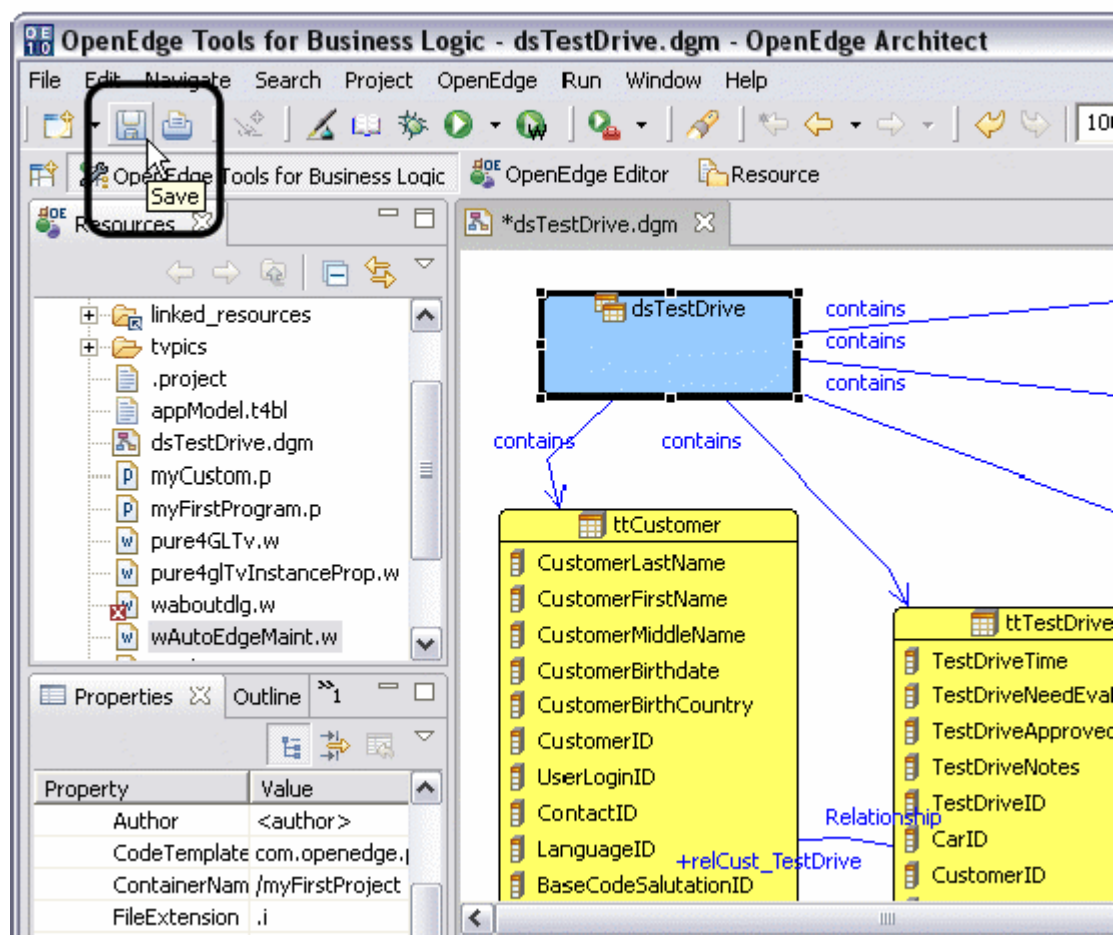
11. Компонентная диаграмма теперь завершена и выглядит следующим образом.



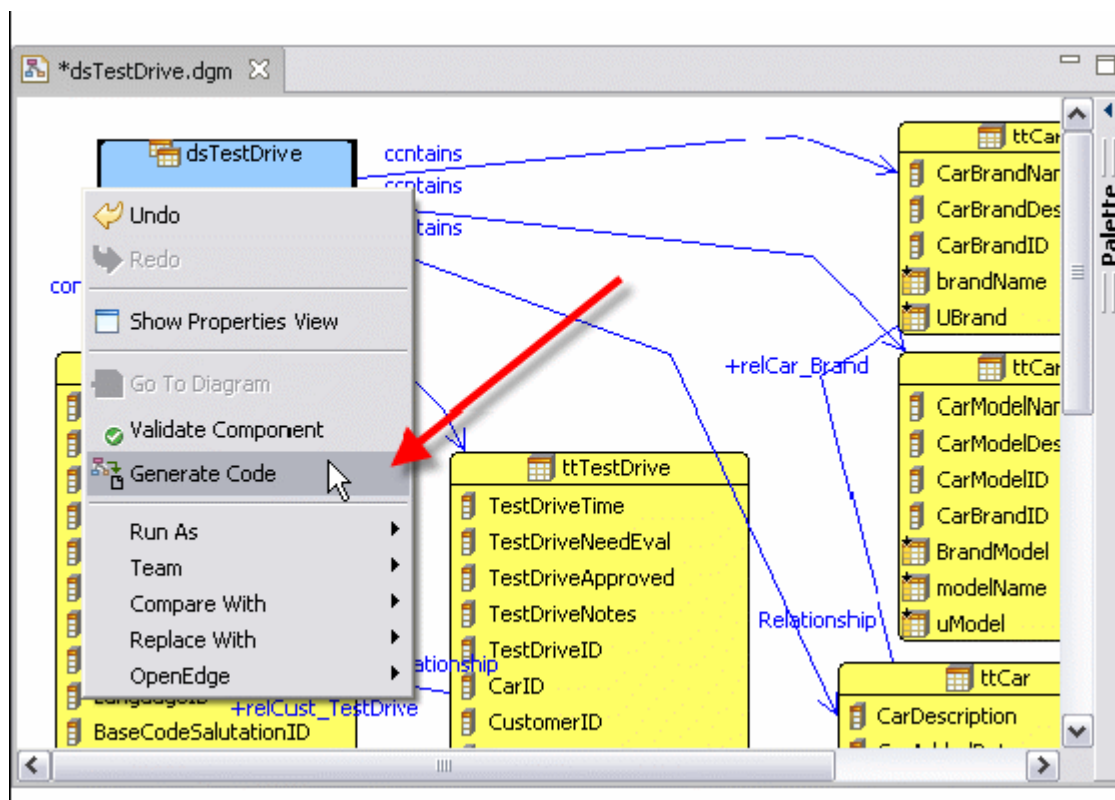
Заблаговременный прикладной код

Так как компонентная диаграмма сформирована, стало возможным использование модели для генерации ABL кода, базирующегося на структуре и свойствах ее составляющих. Чтобы сгенерировать код для ProDataSet dsTestDrive, включающего все связанные определения временных таблиц, выполните следующее.

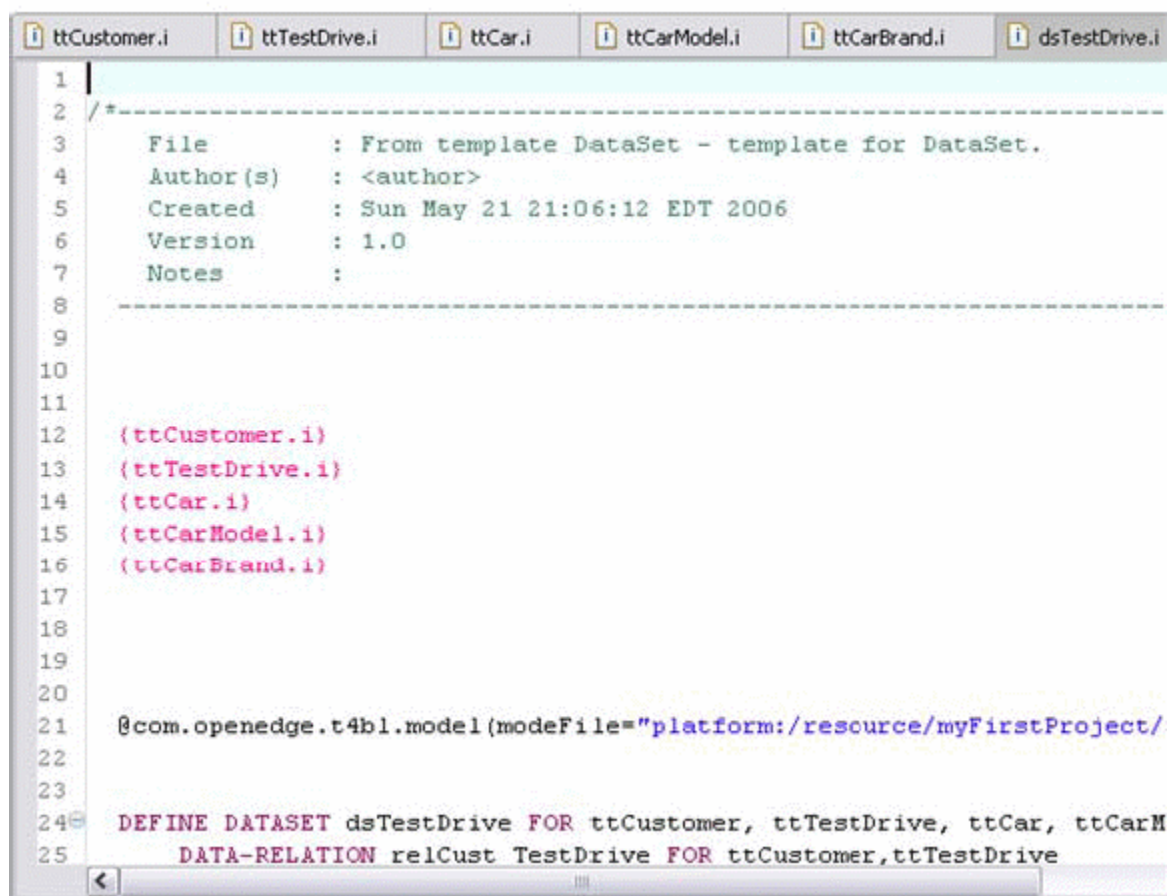
1. Сохраните диаграмму, нажав на кнопку Save на главной панели инструментов.



2. Кликните правой кнопкой мыши по области для размещения компонентов и в появившемся меню выберите опцию создания кода Generate Code.



3. Подключаемые файлы, содержащие ABL-код, будут созданы для ttCustomer, ttTestDrive, ttCar, ttCarModel, ttCarBrand, dsTestDrive компонентов. Проверив, закройте их.



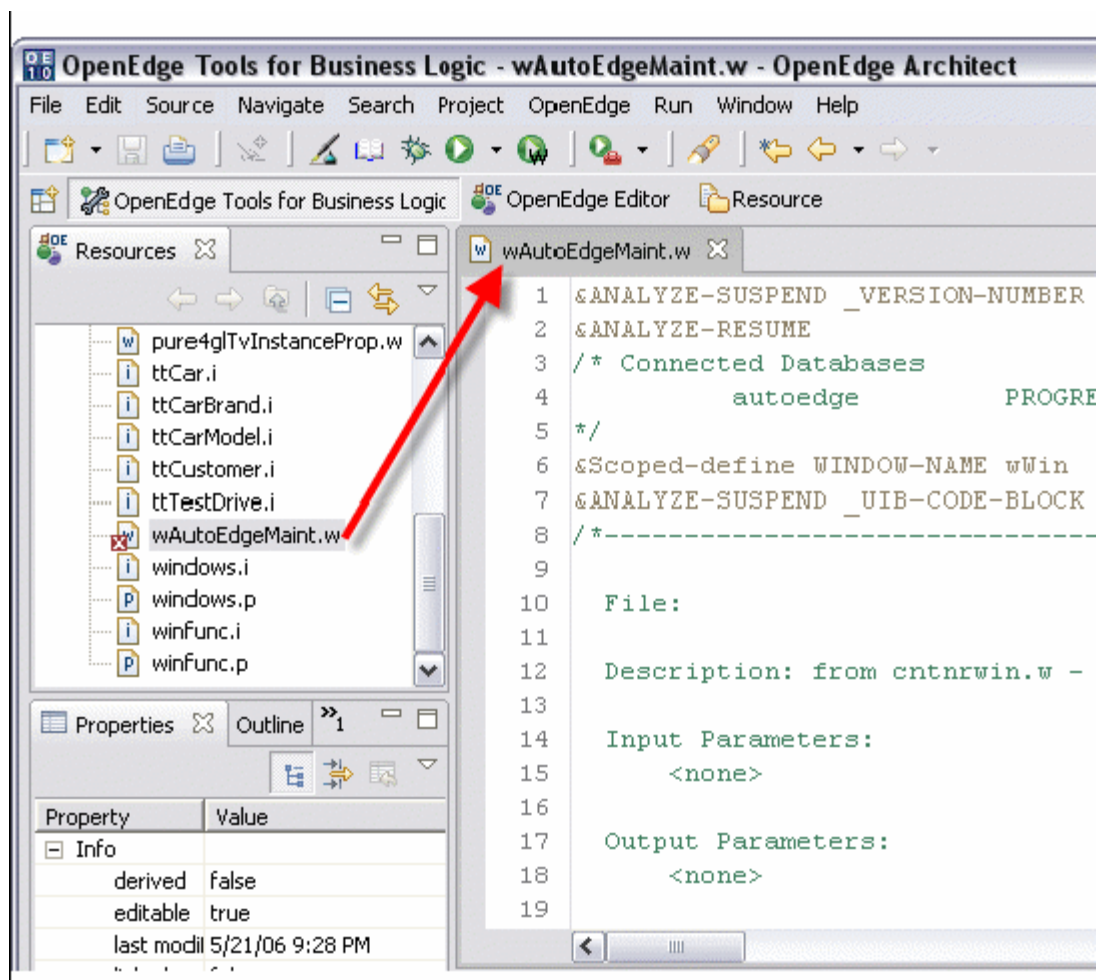
```
1 |
2 /*-----
3   File      : From template DataSet - template for DataSet.
4   Author(s) : <author>
5   Created   : Sun May 21 21:06:12 EDT 2006
6   Version   : 1.0
7   Notes    :
8   -----
9
10
11
12 {ttCustomer.i}
13 {ttTestDrive.i}
14 {ttCar.i}
15 {ttCarModel.i}
16 {ttCarBrand.i}
17
18
19
20
21 @com.openedge.t4b1.model(modeFile="platform:/resource/myFirstProject/
22
23
24 DEFINE DATASET dsTestDrive FOR ttCustomer, ttTestDrive, ttCar, ttCarM
25   DATA-RELATION relCust TestDrive FOR ttCustomer,ttTestDrive
```

Интеграция созданного кода в приложение

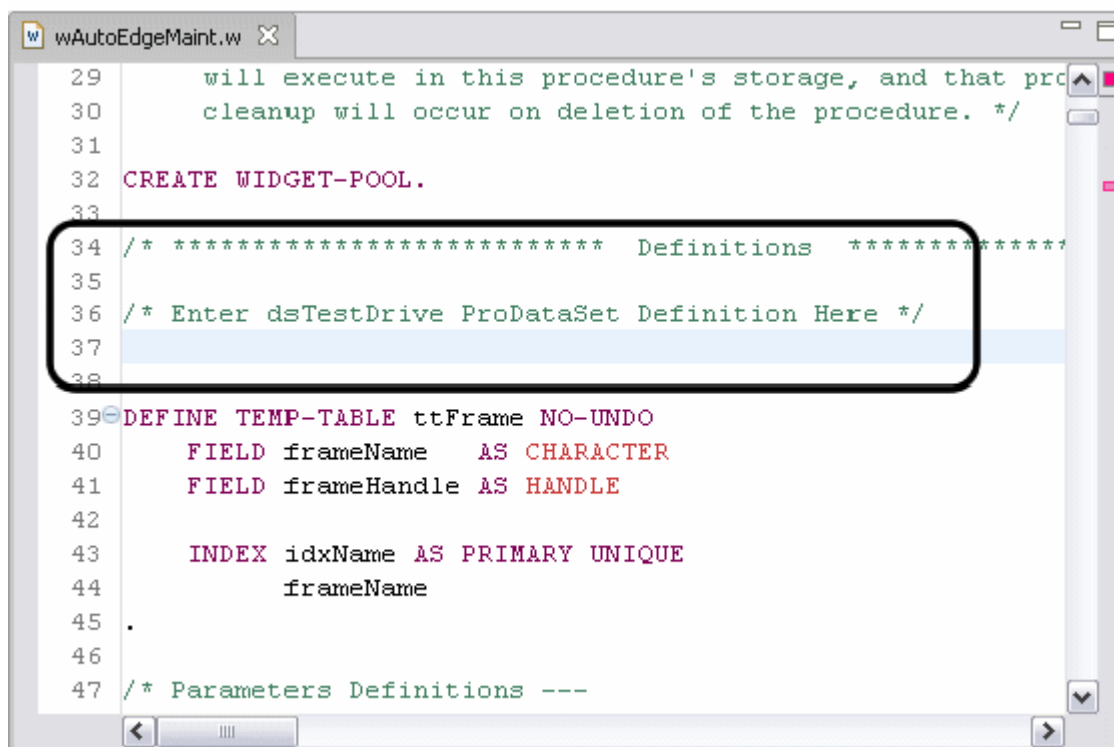
Использование созданного кода в приложении

Чтобы использовать сгенерированный код как часть приложения, необходимо добавить ссылку на подключаемый файл в общий код там, где это требуется. Для добавления компонента dsTestDrive в приложение и его тестирования выполните следующее:

1. Двойным щелчком мыши кликните по программе wAutoEdgeMaint.w в окне Resources для ее открытия в редакторе Editor.

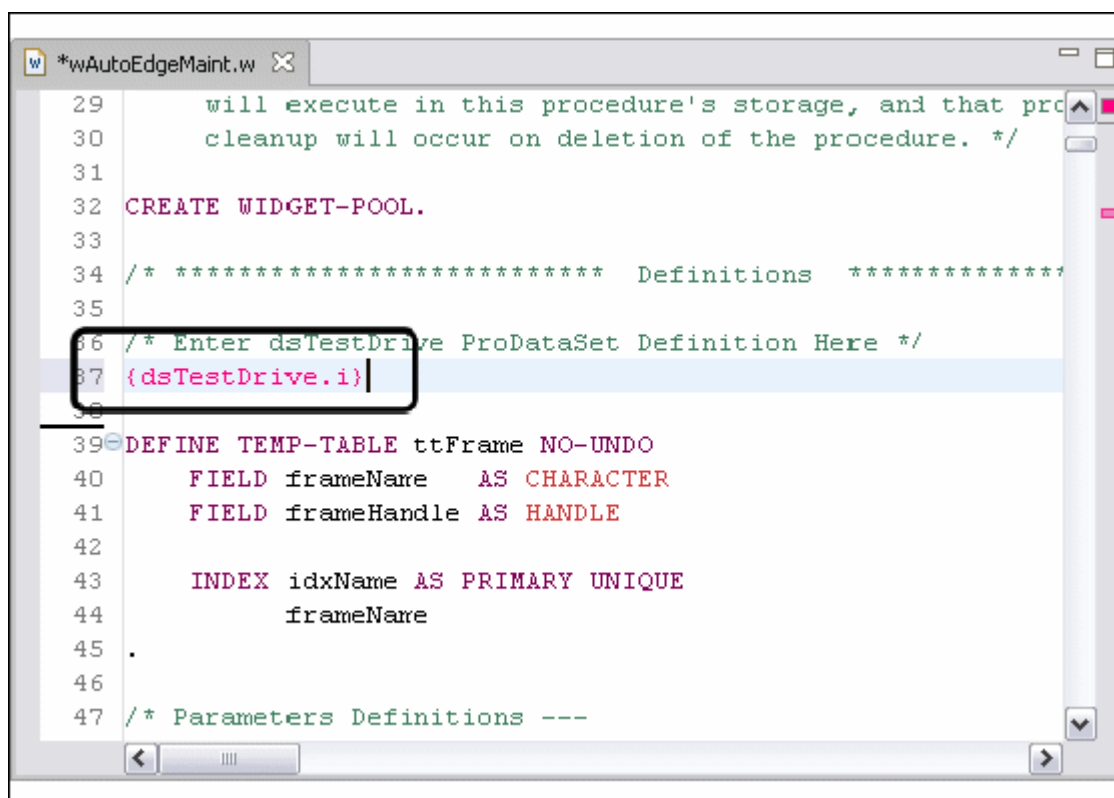


2. Переместитесь с помощью полосы прокрутки в раздел Definitions и найдите комментарий, который определяет место для описания ProDataSet.



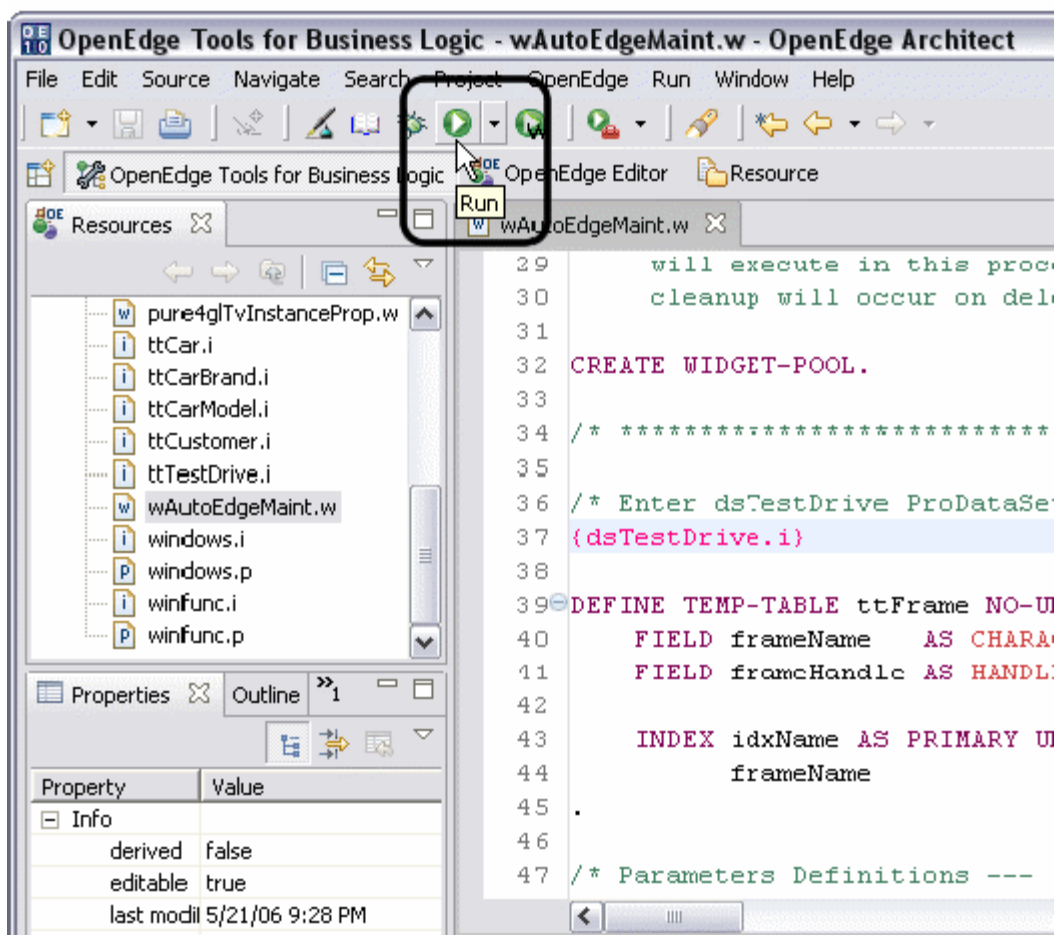
```
wAutoEdgeMaint.w X
29 will execute in this procedure's storage, and that pro
30 cleanup will occur on deletion of the procedure. */
31
32 CREATE WIDGET-POOL.
33
34 /* ***** Definitions *****
35
36 /* Enter dsTestDrive ProDataSet Definition Here */
37
38
39 DEFINE TEMP-TABLE ttFrame NO-UNDO
40     FIELD frameName AS CHARACTER
41     FIELD frameHandle AS HANDLE
42
43     INDEX idxName AS PRIMARY UNIQUE
44         frameName
45 .
46
47 /* Parameters Definitions ---
```

3. Ниже комментария впишите ссылку на подключаемый файл “{dsTestDrive.i}”.



```
*wAutoEdgeMaint.w X
29 will execute in this procedure's storage, and that pro
30 cleanup will occur on deletion of the procedure. */
31
32 CREATE WIDGET-POOL.
33
34 /* ***** Definitions *****
35
36 /* Enter dsTestDrive ProDataSet Definition Here */
37 {dsTestDrive.i}
38
39 DEFINE TEMP-TABLE ttFrame NO-UNDO
40     FIELD frameName AS CHARACTER
41     FIELD frameHandle AS HANDLE
42
43     INDEX idxName AS PRIMARY UNIQUE
44         frameName
45 .
46
47 /* Parameters Definitions ---
```

4. Сохраните файл. Запустите программу на исполнение, нажав на кнопку Run в главном меню.



5. Появилось окно поддержки TestDrive. Вы можете перемещаться по информации, поддерживаемой ProDataSet, раскрывая записи, отражаемые в дереве слева и, выбирая информационные вкладки, справа. Функции поддержки в этом простом окне не доступны. По-окончании тестирования, закройте окно.

