

## **Дополнительная техника программирования для Web**

# 6

Изложенной информации о создании Web-приложений и преобразования существующих Clarion-приложений в Web-приложения достаточно для того, чтобы публиковать приложения баз данных в Интернет.

Но это не полная информация о возможностях Internet Connect. Эта глава продемонстрирует Вам несколько технических приемов, которые Вы сможете использовать в Ваших Web-приложениях.

В оставшейся части данного руководства продолжим работу с примером, который мы использовали в предыдущей главе.

В этой главе Вы будете:

- ◆ Добавлять окно Login (вход в систему) и использовать Cookies, чтобы “вспоминать” пользовательское имя входа при каждом новом старте приложения.
- ◆ Использовать Code Template (шаблона) для встроенного статического (Embed Static) HTML.
- ◆ Использовать Code Template ( шаблона) для встроенного динамического (Embed Dynamic) HTML, используя переменную.
- ◆ Использовать вставку Internet Embed для написания условного кода для HTML.
- ◆ Защищать процедуру паролем.
- ◆ Добавить всплывающее Web окно для того, чтобы информировать пользователя -новичка о том, что загружена Java Support Library.
- ◆ Использовать встроенный HTML для выравнивания изображения в Web
- ◆ Использовать Individual Control Options для обеспечения встроенных кодовых вставок, исполняемых через Web.
- ◆ Использовать встроенные кодовые вставки для ограничения редактирования по месту (Edit-In-Place) во время исполнения через Web.

**Это должно занять около тридцати минут.** К концу этой главы Вы изучите большую часть методов доведения Ваших Web-приложений до максимального соответствия поставленным задачам.

Давайте продолжим!

## **Использование Cookies (переменные, используемые для координации в межадачном пространстве)**

В этом разделе мы будем добавлять окно для регистрации в системе, позволяющее пользователям идентифицировать самих себя. Приложение будет использовать cookies для записи имени и для его “запоминания”. В следующий раз, когда пользователь запустит приложение, приглашение ввести имя не появится.

**Отправная точка: У Вас должен быть открыт в среде разработки Clarion файл WebTree.app.**

Это руководство предполагает, что Вы установили Clarion в директорию C:\CW20, а Application Broker - в директорию C:\CWICWEB. Если Вы используете другие директории, то должны будете, соответственно, изменить команды.

### **Добавление процедуры входа в систему**

1. В дереве приложения установите курсор на процедуре main и нажмите кнопку Properties (свойства).

Появится окно Procedure Properties.

2. Нажмите кнопку Embeds.

Появится окно Embedded Source (код-вставка).

3. Подсветите строку WindowManager Executable Code Section - Init-(),BYTE.

4. Нажмите кнопку Insert.

Появится окно Select Embed Type (выберите тип вставки)

5. Подсветите строку Call a Procedure и нажмите кнопку Select.

6. В поле Procedure to call (вызываемая процедура) наберите Login Window и нажмите кнопку ОК.

7. Установите значение Priority равным 1.

Это обеспечит LoginWindow перед открытием окна.

8. Нажмите кнопку Close в окне Embedded Source и кнопку ОК в окне Procedure Properties.

Это добавит процедуру LoginWindow с типом ToDo.

### **Добавление окна входа в систему.**

1. В дереве приложения выберите процедуру LoginWindow и нажмите кнопку Properties. Появится окно Select Procedure Type.

2. Выберите Window-Generic Window Handler (общий обработчик окна) и нажмите кнопку Select.

Появится окно Procedure Properties.

3. Нажмите кнопку Window.

Появится окно New Structure.

4. Выделите Window и нажмите ОК.

### **Разработка окна входа в систему.**

1. Выберите Populate > Field.

Появится File Schematic

2. В списке с названием Files, расположенном в левой части, выделите Global Data, а затем в списке (справа) с названием Fields выберите LoginName и нажмите кнопку Select.

Эта переменная была создана для Вас в учебном приложении.

3. Для размещения текста приглашения введите имя и поля ввода, щелкните над окном кнопкой мыши .

4. Выберите Populate > Control Template.

Появится окно Select Control Template.

5. Выделите CancelButton и нажмите кнопку Select.

6. Для размещения кнопки Cancel, щелкните над окном кнопкой мыши.

7. Выберите Populate > Control Template.

8. Выделите CloseButton и нажмите кнопку Select.

9. Для размещения кнопки Close щелкните над окном кнопкой мыши.

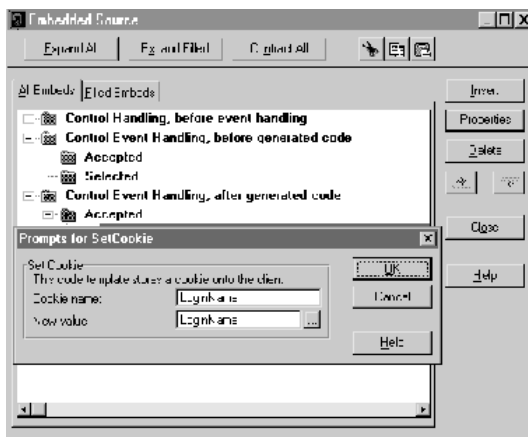
10. Измените текст на кнопке Close на ОК.

11. Расположите управляющие элементы как на рисунке.



### Добавление “Cookie” - кода для сохранения LoginName

1. Произведите двойной щелчок мышью на кнопке ОК для доступа к пунктам вставки кода для этой кнопки.
2. Выделите пункт Control Event Handling, After Generated Code, Accepted и нажмите кнопку Insert.
3. Выберите шаблон (template) SetCookie и нажмите кнопку Select
4. В поле Cookie name введите LoginName.
5. В поле New Value введите LoginName (или выберите глобальную переменную LoginName из File schematic используя кнопку (...)).



6. Нажмите кнопку ОК.
7. Нажмите кнопку Close в окне Embedded Source.

### Добавьте “Cookie”-код для получения LoginName.

1. Произведите двойной щелчок мышью над окном для получения доступа к точкам вставки встроенного кода для этого окна.

2. Выделите точку вставки кода Accept Loop, Before CASE EVENT() handling (цикл обработки оператора Accept, перед предложением CASE EVENT()) и нажмите кнопку Insert.

3. Выделите Source и нажмите кнопку Insert.

4. Наберите следующий исходный код:

```
IF EVENT() = Event:NewPage                !Если web-страница новая
  LoginName = Broker.Http.GetCookie('LoginName') !Получить cookie
  DISPLAY
END
IF LoginName                               !Если значение было установлено
  POST(Event:CloseWindow)                 ! закрыть окно
END
```

Этот код “получает” cookie когда окно активно. Если код успешно находит cookie и устанавливает переменную LoginName, то он закрывает окно (до того как пользователь увидит его).

Это означает что пользователь должен регистрироваться при входе только один раз, а после этого сервер “распознает” пользователя при следующих входах в систему.

5. Покиньте редактор исходных текстов и сохраните изменения.

6. Нажмите кнопку Close в окне Embedded Source.

7. Выйдите из форматера окна (Window Formatter) и сохраните изменения.

8. Нажмите кнопку ОК в окне Procedure Properties.

### **Создание и размещение**

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).

Ваше web-приложение готово к размещению.

2. Нажмите кнопку ОК в окне компилятора.

3. Откройте Windows Explorer (проводник) (или Windows NT Explorer).

4. Скопируйте WebTree.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\.

### **Запуск приложения**

1. Запустите Application Broker двойным щелчком на файле CWBrokr1.exe или CWBroker.exe (если у Вас полная версия Application Broker) из директории C:\CWICWEB\.

#### **Примечание**

Как и в первом руководстве, мы будем использовать исполняемую версию Application Broker. Версия ISAPI работает похожим образом, но с некоторыми отличиями. Это обсуждается в главе Application Broker.

2. Запустите Ваш браузер.
3. Запустите в браузере приложение WebTree.  
(http://localhost/exec/webtutor/webtree.exe)

### **Проверка приложения**

Вы запускаете приложение в первый раз. Вам выдается предложение ввести имя для регистрации в системе. При следующем запуске предложение ввести имя не будет выдаваться, так как система прочитает cookie и значение глобальной переменной будет установлено равным значению cookie.

1. Введите имя, когда появится экран Login и нажмите ОК.
2. Выйдите из приложения.
3. Перезапустите в браузере приложение WebTree.

Заметьте, что при втором запуске приглашение зарегистрироваться в системе не появляется.

4. Выйдите из приложения.

Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

Давайте сделаем следующее изменение для отображения имени пользователя (LoginName), используя шаблон (template) Dynamic HTML.

## ***Встроенный HTML***

Одна из наиболее мощных возможностей Internet Developer's Kit - это способность вставлять HTML-код в HTML-страницы, порождаемые web-приложением.

Когда Вы встраиваете HTML-код (используя специальные точки для вставки, добавленные шаблоном Global Internet Application Template), он вставляется в определенные места в HTML, возвращаемый в браузер, выполняющий приложение.

**Отправная точка:** У Вас должен быть открыт в среде разработки Clarion файл WebTree.app.

### **Добавление динамического HTML, используя переменную**

Мы написали код, необходимый для установки, восстановления входного имени пользователя и записи его в глобальную переменную. Теперь мы обеспечим отображение этого имени на web-странице ниже HTML-представления окна.

1. В дереве приложения выделите процедуру Main и нажмите кнопку Properties.

Появится окно Procedure Properties.

2. Нажмите кнопку Embeds.

Появится окно Embedded Source.

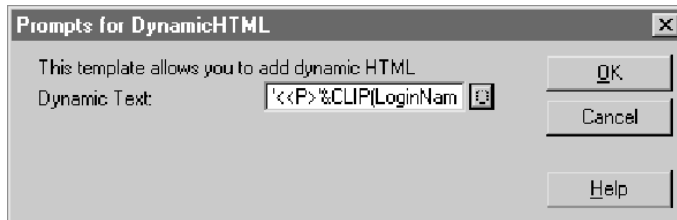
3. Выделите пункт для вставки кода Internet-Before the Closing </BODY> tag, затем нажмите кнопку Insert.

Появится окно Select Embed Type.

4. Выделите Dynamic HTML и нажмите кнопку Select.

5. В поле Dynamic Text наберите следующее:

'<<P>' & CLIP(LoginName) & ' is logged in <</P>'



6. Нажмите кнопку ОК в окне шаблона.

7. В окне Embedded Source нажмите кнопку Close, затем в окне Procedure Properties кнопку ОК.

### **Создание и размещение**

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).

Ваше web-приложение готово к размещению.

2. Нажмите кнопку ОК в окне компилятора.

3. Откройте Windows Explorer (или Windows NT Explorer).

4. Скопируйте WebTree.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\.

### **Проверка приложения**

1. Перезапустите в браузере приложение WebTree (щелкните по гиперсвязи Restart).

Если Вы уже запускали приложение на этой машине, то у Вас не появится окно

регистрации. Вместо этого сервер прочитает Ваш “cookie” и занесет это значение в глобальную переменную LoginName. Переменная LoginName теперь будет отображаться на web странице ниже кнопок панели инструментов.

## 2. Выйдите из приложения.

Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

Давайте произведем еще несколько изменений в приложении, используя встраиваемый HTML.

## **Добавление статического HTML.**

В последнем разделе мы добавляли HTML-код, который составлялся с использованием комбинации текста и переменных. В этом разделе мы будем использовать шаблон Static HTML для добавления HTML-кода, который будет оставаться статическим.

Мы будем использовать это для добавления ссылки (Link) в конце страницы, которая позволит пользователям отправлять E-mail для Web-мастера с комментариями или вопросами к приложению.

1. В дереве приложения выделите процедуру Main, затем нажмите кнопку Properties.

Появится окно Procedure Properties.

2. Нажмите кнопку Embeds.

Появится окно Embedded Source.

3. Выделите пункт для вставки кода Internet-Before the Closing </BODY> tag и нажмите кнопку Insert.

Появится окно Select Embed Type (выберите тип вставки).

4. Выделите Static HTML и нажмите кнопку Select.

5. В поле HTML to Insert наберите следующее:

```
<P> <A HREF="mailto:nobody@topspeed.com">Comments?</A></P>
```

6. Нажмите кнопку ОК в окне шаблона.

7. Нажмите кнопку Close в окне Embedded Source и кнопку ОК в окне Procedure Properties.

## **Создание и размещение**

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).



Ваше web-приложение готово к размещению.

2. Нажмите кнопку ОК в окне компилятора.
3. Откройте Windows Explorer (или Windows NT Explorer).
4. Скопируйте WebTree.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\.

### **Проверка приложения**

1. Перезапустите в браузере приложение WebTree (щелкните по гиперсвязи Restart).

Вы заметите новую связь(Link). Если Вы щелкните по связи, то Ваш браузер откроет Вашего E-mail “клиента” с новым сообщением с предопределенным адресом.

2. Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

### **Добавление условного HTML в исходный код Clarion**

Третьим методом вставки HTML-кода в Ваши web-страницы является использование во вставках исходного кода метода Target.WriteLine. Это позволит Вам использовать исходный код Clarion для написания HTML - кода. Одним из преимуществ использования исходного кода Clarion является возможность управлять тем, какой код HTML Вы хотите написать. Другими словами, Вы можете использовать управляющие структуры языка Clarion для управления тем, что пишется. Вы можете написать одну или другую строку, используя структуры языка IF.THEN..ELSE или CASE.

Мы будем использовать этот прием для случайного отображения рекламы в конце страницы, используя структуру EXECUTE.

1. В дереве приложения выделите процедуру Main и нажмите кнопку Properties.

Появится окно Procedure Properties.

2. Нажмите кнопку Embeds.

Появится окно Embedded Source.

3. Выделите пункт вставки кода Internet-Before the Closing </BODY> tag и нажмите кнопку Insert.

Появится окно Select Embed Type.

4. Выделите Source, затем нажмите кнопку Select.

5. В Embedded Source editor (редактор вставок кода) наберите следующий

ИСХОДНЫЙ КОД:

```
Str1" = '<<A HREF=http://www.'
```

```
Str2" = '.com"><<IMG SRC=""
```

```
Str3" = ""BORDER=0><</A'
```

```
EXECUTE RANDOM(1,5)
```

```
Target.WriteLine(CLIP(Str1") & 'topspeed' & clip(Str2") & SELF/Files.GetAlias('1.GIF') & Str3")
```

```
Target.WriteLine(CLIP(Str1") & 'icetips' & clip(Str2") & SELF/Files.GetAlias('2.GIF') & Str3")
```

```
Target.WriteLine(CLIP(Str1") & 'finatics' & clip(Str2") & SELF/Files.GetAlias('3.GIF') & Str3")
```

```
Target.WriteLine(CLIP(Str1") & 'flpathers' & clip(Str2") & SELF/Files.GetAlias('4.GIF') & Str3")
```

```
Target.WriteLine(CLIP(Str1") & 'flamarlins' & clip(Str2") & SELF/Files.GetAlias('5.GIF') & Str3")
```

```
END
```

**Примечание:** Вы можете скопировать и вставить этот текст из chap4.txt из директории \webtutor.

6. Выйдите из редактора и сохраните изменения.

7. Нажмите кнопку Close в окне Embedded Source и OK в окне Procedure Properties.

### Создание и размещение

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).

Ваше web-приложение готово к размещению.

2. Нажмите кнопку OK в окне компилятора.

3. Откройте Windows Explorer(Проводник) (или Windows NT Explorer).

4. Скопируйте WebTree.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\.

5. Скопируйте GIF файлы (\*.gif) из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\EXEC\Public.

### Проверка приложения

1. Перезапустите в броузере приложение WebTree (щелкните по гиперсвязи Restart).

Вы заметите новое изображение и связь(Link). Каждый раз при запуске приложения будет показываться реклама выбираемая по случайному закону.

2. Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

## **Защита операции подгрузки с помощью Splash окна**

Для того, чтобы запустить приложение, приспособленное для Web, браузеру клиента должна быть доступна JSL (Java Support Library). Начинаящие пользователи должны подгрузить Clarion.CAB(для Microsoft Internet Explorer) или Clarion.ZIP (для Netscape). Для большинства браузеров JSL подгружается единовременно и сохраняется в кэше (до тех пор, пока пользователь не очистит его). Хотя JSL чрезвычайно компактна для той функциональности, какую она обеспечивает, ее подгрузка займет несколько минут. Имея это ввиду, мы будем использовать всплывающее окно для предупреждения начинающего пользователя о том, что идет подгрузка. Размещая кнопки Java в этом окне, мы можем воспрепятствовать работе пользователя до тех пор, пока не загрузится JSL и кнопки Java не будут инициализированы.

1. В Application Tree подсветите Main процедуру и нажмите клавишу Properties.

Откроется окно Procedure Properties

2. Нажмите клавишу Embeds.

Откроется окно Embedded Source.

3. Подсветите WindowManagerExecutable Code Section - Init-(),BYTE.

4. Нажмите клавишу Insert.

Откроется окно Select Embed Type.

5. Выделите подсветкой Source, нажмите клавишу Select.

6. В редакторе Embedded Source введите следующее:

```
IF WebServer.Active THEN Splash
```

Это позволит процедуре Splash быть вызванной, когда приложение выполняется в Web.

7. Установите Priority равным единице.

8. Убедитесь, что эта вставка внесена в список перед вызовом процедуры LoginWindow, использующей клавиши "вверх" и "вниз".

Это обеспечивает то, что Splash процедура вызывается перед открытием других окон.

9. Нажмите клавишу Button в окне Embed Source.

10. Нажмите клавишу Procedure.

Откроется окно Procedure.

11. Подсветите Splash, нажмите кнопку ОК.

Это связывает Splash процедуру с Main процедурой в Application Tree.

Последнее необходимо, если ваше приложение используется в LocalMAPs.

## **Переименование BUTTON в Java Button**

Окно Splash содержит некоторый текст, клавишу и управление IMAGE. К клавише BUTTON "подвязан" управляющий шаблон CloseButton с текстом, измененным на Continue. Так как кнопка создана как HTML кнопка по умолчанию, вы будете специфицировать ее иначе. Так как мы хотим, чтобы это была Java кнопка, она не будет доступна конечному пользователю до тех пор, пока JSL будет подгружаться.

1. В Application Tree подсветите Splash процедуру, нажмите кнопку Properties.
2. Нажмите кнопку Internet Options.
3. Выберите закладку Controls.
4. Подсветите ?Close в списке Individual Control Overrides, нажмите кнопку Properties.
5. Выберите закладку Classes.
6. Пометьте признак Override default Class, выберите WebJavaButtenClass из выпадающего списка ClassName.
7. Нажмите ОК.

Оставьте открытым окно Internet Options. Мы будем использовать его далее.

## **Центрирование поля Image в окне Splash.**

Управляющий элемент IMAGE окна Splash располагается по центру горизонтали окна. В этой части руководства будет добавлен некоторый HTML код, обеспечивающий отцентрованное положение IMAGE во время исполнения в Web.

1. Подсветите ?Image в списке Individual Control Overrides (индивидуальные настройки управления), затем нажмите кнопку Properties.
2. Выберите закладку Tab.  
Это окно позволит вам ввести HTML код перед и после управляющего элемента. Код будет влиять на управляющий элемент только во время исполнения в Web.
3. В поле HTML before control введите:  
<CENTER>
4. В поле HTML after control введите:  
</CENTER>
5. Нажмите кнопку ОК во всех окнах (3 раза) и вернитесь в Application Tree.

### **Создание и размещение**

1. Выберите Project-Make (или нажмите кнопку Make панели).  
Ваше Web приложение готово к размещению.
2. Нажмите кнопку ОК в окне компилятора.
3. Откройте Windows Explorer (или Windows NT Explorer).
4. Скопируйте WebTree.exe из директории C:\Clarion4\Examples\WebTutor в директорию директорию C:\CWICWEB\EXEC\WebTutor.

### **Проверка приложения**

1. Перезапустите в броузере приложение WebTree (щелкните по гиперсвязи Restart).  
Вы можете обнаружить, что Splash окно появилось перед другим окном.
2. Выйдите из приложения.  
Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

## ***Применение частичного обновления к управляющим элементам Update***

В приложениях под Windows программисты часто вставляют код для изменения одного элемента управления, когда значение другого меняется. Например, вы могли вставить код для изменения суммы переменных, когда меняется количество переменных. Код, подобный этому, дерево приложений для Web(Webtree) имеет в процедуре UpdateItems. Встроенный код связан EVENT:Accepted на каждом элементе управления. Другими словами, когда пользователь изменяет значение в управляющем элементе или выбирает другой управляющий элемент щелчком мыши, код выполняется.

Когда приложение выполняется через Web, управляющий элемент ENTRY обрабатывается в броузере по умолчанию. Другими словами, нет взаимодействия между браузером и сервером приложений, пока вы не меняете вручную опции для этого элемента управления. В этой главе вы будете изменять действие для трех управляющих элементов, чтобы обеспечить исполнение кодовой вставки на сервере для Event:Accepted для этих элементов.

1. В Application Tree подсветите UpdateItems процедуру и нажмите клавишу Properties.  
Откроется окно Procedure Properties
2. Нажмите клавишу Internet Options.
3. Выберите закладку Controls.
4. Подсветите ?Item:ProdCode в списке Individual Control Overrides, затем нажмите клавишу Properties.
5. Выберите закладку Events.
6. Выделите подсветкой Accepted, нажмите клавишу Select.
7. Поставьте признак Override default action, выберите Partial page refresh из выпадающего списка Action on Event.
8. Нажмите ОК во всех окнах, пока не вернетесь в окно Internet Options.(2 раза)
9. Повторите последние 5 шагов для ?ITEM:Quantity и ?ITEM:Price.
10. Нажмите ОК во всех окнах до возвращения в Application Tree (2 раза).

### **Создание и размещение**

1. Выберите Project-Make (или нажмите кнопку Make панели).  
Ваше Web приложение готово к размещению.
2. Нажмите кнопку ОК в окне компилятора.
3. Откройте Windows Explorer (или Windows NT Explorer).
4. Скопируйте WebTree.exe из директории C:\Clarion4\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.

### **Проверка приложения**

1. Перезапустите в браузере приложение WebTree (щелкните по гиперсвязи Restart).
2. Нажмите кнопку Orders.
3. Нажмите кнопку Expand All.
4. Подсветите одну из тем (зеленая линия).
5. Нажмите кнопку Change.
6. Измените итог(количество) в поле Quantity, нажмите Tab.  
Отметьте изменения Extended Total. Если вы меняли поле Price или Product Code, поле Extended Total также изменится.
7. Выйдите из приложения.  
Покиньте Ваш браузер, оставив открытой страницу перезапуска. Вы будете использовать это для перезапуска приложения после ввода изменений.

## **Ограничение доступа к процедуре**

В следующей части руководства мы будем ограничивать доступ к процедуре, используя встроенную в браузер поддержку аутентификации и возможности шаблона Internet Procedure Extension по парольной защите. При вызове процедуры, защищенной паролем, на экране появляется окно аутентификации браузера. От Вас не требуется создавать окно сбора информации для входа. Парольная защита основана на области, имени пользователя и пароле. “Область” - это защищенная процедура.

Браузер запрашивает у пользователя имя и пароль. Затем они посылаются в программу для проверки. Если программа принимает пароль (то есть возвращает TRUE из метода WebWindow.ValidatePassword), предъявляется новая страница, в противном случае браузер повторяет запрос. После трех попыток браузер выдает сообщение, информирующее пользователя о том, что доступ запрещен. Эта страница автоматически возвращает пользователя к последнему активному месту в программе.

**Примечание:** если страница уже посещалась в текущем сеансе, браузер обычно обеспечивает имя и пароль без опроса пользователя. Это свойство встроено в большинство браузеров.

Существует несколько методов парольной защиты (смотрите Using Passwords Web в главе Application Design Considerations). Мы будем использовать более передовой метод - переопределим метод WebWindow.ValidatePassword.

**Исходное положение:** у Вас должны быть открыты Среда разработки Clarion и приложение WebTree.app.

## **Парольная защита**

---

Мы написали код, необходимый для установки и восстановления имени пользователя для входа. Сейчас мы покажем это имя на web-странице ниже HTML-представления окна.

## **Добавьте файл проверки**

1. В Дереве приложения выделите процедуру UpdateProducts, затем нажмите кнопку Properties.

Появляется окно Procedure Properties.

2. Нажмите кнопку Files.

Появляется File Schematic.

3. Выделите Other Files, затем нажмите кнопку Insert.

Появляется Insert File.

4. Выделите Userlist, затем нажмите кнопку Select.

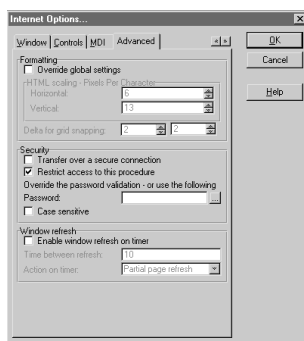
5. Нажмите кнопку OK на окне File Schematic.

## **Добавьте вызов пароля**

1. Нажмите кнопку Internet Options.

2. Выделите закладку Advanced.

3. Установите флаг Restrict access to this procedure (ограничить доступ к данной процедуре).



4. Нажмите кнопку OK.

5. Нажмите кнопку Embeds (вставки).

Появляется окно Embedded Source.

6. Выделите точку вставки кода Internet- Password validation Code Section, затем нажмите кнопку Insert.

Появляется окно Select Embed Type.

Вводя код в точке вставки Internet- Password validation Code Section, Вы преопределяете принятый по умолчанию метод для проверки пароля.

Эта точка вставки помещает код внутри метода с двумя параметрами: Именем пользователя и Паролем, которые получены от броузера. Вы должны будете вернуть TRUE, если пароль правильный, и FALSE - если неправильный. Это



позволяет Вам просматривать информацию в файле или использовать другой метод, выбранный вами для проверки пароля.

7. Выделите Source, затем нажмите кнопку Select.

8. В Embedded Source editor напишите следующий код:

```
USE:UserID -UserName
Get(Userlist.Use:KeyUserID)           !lookup UserName in file
IF ERRORCODE() THEN RETURN(false).    !Return false on invalid user
IF USE:UserPassword=Password          !Check the password
    RETURN(True)
ELSE
    RETURN(False)
END
```

9. Выйдите из редактора текста и сохраните изменения.

10. Нажмите кнопку Close на окне Embedded Source и кнопку ОК на окне Procedure Properties.

### **Создание и размещение**

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).

Ваше web-приложение готово к размещению.

2. Нажмите кнопку ОК в окне компилятора.

3. Откройте Windows Explorer(Проводник) (или Windows NT Explorer).

4. Скопируйте WebTree.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.

### **Проверка приложения.**

1. Перезапустите в броузере приложение WebTree (щелкните по гиперсвязи Restart).

2. Нажмите кнопку Products.

3. Нажмите кнопку Insert, чтобы добавить новый продукт.

Появляется окно аутентификации броузера.

4. В поле UserName напишите Fred.

5. В поле Password напишите Wilma.

Введенные вами значения находятся в файле Userlist. Этот файл был создан заранее с двумя пользователями. Заметьте, что в этом приложении нет процедуры для редактирования этого файла. Это обычный способ обращения с паролем пользователя, при котором только у системного администратора есть разрешение добавлять пользователей. Можете создавать процедуры для изменения этого файла, как Вы это сочтете нужным.

6. Выйдите из приложения.

## **Ограничение редактирования по месту**

Шаблоны ABC в Clarion4 позволяют вам использовать редактирование по месту (Edit-In-Place) в простом элементе выбора. Это свойство, однако, не поддерживается во время исполнения через Web. В этом случае вы должны иметь отдельную форму для изменений. Это простой метод - альтернатива между редактированием по месту при локальном исполнении под Windows и вызовом формы при исполнении в Web.

Если вы используете редактирование по месту и уточняете процедуру шаблоном управления BrowseBox, вы уже выполнили две трети работы. Шаблон, сгенерировавший код, или вызывает отдельную процедуру обновления, или осуществляет редактирование по месту в зависимости от значения BRWn.AskProcedure. Установите значение BRWn.AskProcedure в ноль, и получите редактирование по месту; установите в единицу и получите процедуру обновления (update).

Для того, чтобы использовать редактирование по месту для локального Windows и видоизменить при исполнении в Web:

1. Выберите процедуру Browse, затем нажмите клавишу Properties.
2. В разделе UpdateButton окна Procedure Properties установите флаг Use Edit in Place (использовать редактирование по месту).

Отметьте, что процедура замены всегда уточняется.

Далее, задайте код, чтобы установить обновление при редактировании по месту, когда исполняется в Windows и вызовите form при исполнении через Web.

3. Нажмите кнопку Button.

Откроется окно Embedded Source.

4. Выделите точку вставки кода Window Manager Method-Init-BYTE(), затем нажмите кнопку Insert.
5. Выделите Source, нажмите кнопку Select.
6. В редакторе Embedded Source введите следующий код:  
IF WebServer.Active  
BRW1:AskProcedure=1  
END
7. Выйдите из редактора и сохраните изменения.
8. Нажмите кнопку Close в окне Embedded Source и ОК в окне Procedure Properties.

Оставьте значения в Property в качестве значений по умолчанию.

### **Создание и размещение**

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).
- Ваше web-приложение готово к размещению.
2. Нажмите кнопку ОК в окне компилятора.
  3. Откройте Windows Explorer(Проводник) (или Windows NT Explorer).
  4. Скопируйте WebTree.exe из директории C:\Clarion4\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.

### **Проверка приложения.**

1. Перезапустите в броузере приложение WebTree (щелкните по гиперсвязи Restart).
2. Нажмите кнопку Products.
3. Нажмите кнопку Insert, чтобы добавить новый продукт.  
Появляется окно аутентификации броузера.
4. В поле UserName напишите Fred. В поле пароля введите Wilma. Отметьте, что появилась форма Update Products.
5. Выйдите из приложения.
6. Запустите приложение "под Windows".
7. Нажмите клавишу Products.
8. Нажмите Insert, чтобы добавить новый продукт.  
Отметьте, что Edit-In-Place доступна.

Выйдите из приложения.

Поздравляем! Вы успешно завершили учебный курс данного руководства. У Вас теперь должно быть достаточно опыта, чтобы создавать крепкие web-приложения для базы данных.

Оставшаяся часть книги разъясняет шаблоны IBC, библиотеку IBC и приемы создания приложений. Читайте дальше.