

CLARION
Internet
Connect

Руководство пользователя

COPYWRITE 1997 TopSpeed Corporation

Это издание защищено от копирования, и все права принадлежат компании TopSpeed.

Оно не может быть полностью или частично скопировано или переснято, размножено, переведено в любой электронный вид без предварительного письменного разрешения компании TopSpeed.

Это руководство сопровождает программный продукт Clarion Internet Connect. Возможно, оно содержит технические или типографские ошибки. TopSpeed Corporation дает эту публикацию без каких либо гарантий, явных или подразумеваемых.

TopSpeed Corporation
150 East Sample Road
Pompano Beach, Florida 33064
(954) 785-4555

Зарегистрированные торговые марки:

TopSpeed ® зарегистрированная торговая марка TopSpeed Corporation.

Clarion ™ торговая марка TopSpeed Corporation.

Microsoft ® Windows ® , Windows 95 ® и Windows NT ® зарегистрированная торговая марка Microsoft Corporation.

Все прочие названия продуктов и фирм (компаний) являются собственными торговыми марками владельцев.

Оглавление

Введение	1
Добро пожаловать!	1
Что такое Internet Connect?	2
Internet Connect и Среда разработки Clarion	2
Что Вы найдете в этой книге	4
Где найти дополнительную информацию	5
Соглашения в документации	6
Соглашения по написанию шрифтов:	6
Соглашения о клавиатуре:	6
Информация о продукте	6
Регистрация данного продукта	6
Техническая поддержка	7
Установка	9
Системные требования	9
Система разработки	9
Система клиента	9
Server System	10
Программа установки	11
Начало установки	11
Параметры установки	11
Регистрация набора шаблонов IBC	12
Брокер приложений	13
Запуск брокера приложений	14
Использование версии на одно соединение	14
Подключение к Вашему приложению	15
Версия ISAPI брокера приложений	16
Как он работает	17
Установка и подготовка к работе Clarion Application Broker (брокера приложений Clarion) ISAPI DLLs	17
Утилита установки для ISAPI Broker	20
Файлы, размещаемые программой инсталляции установки брокера приложений Clarion (Application Broker)	20
Директории	21

Параметры установки Application Broker (брокера приложений)	22
Удаленный доступ к брокеру приложений	23
Порядок разработки и размещения приложений	24
Локальное тестирование	27
Проверка Вашего TCP/IP соединения	28
Руководство по созданию Web-приложений	29
Мастер Web приложений	30
Создание гибридных Web/Windows приложений	30
Размещение приложения	33
Чем быстрее, тем лучше - оптимизация Вашего приложения	36
Внешний вид очень важен - Добавление графики	40
Превращение готовых	
Clarion-приложений в Web-приложения	43
Использование шаблона расширения для глобальных	
Интернет-приложений	44
Перенос приложения в web	44
Дополнительная техника	
программирования для Web	49
Использование Cookies (переменные, используемые для	
координат в межзадачном пространстве)	50
Встроенный HTML	54
Ограничение доступа к процедуре	58
Парольная защита	59
Шаблоны Internet Builder Class	63
Global Internet Extention Template	63
Настройки страницы	64
Настройки окна	65
Помощь	65
Управляющий элемент	66
Многодокументный интерфейс	68
Дополнительно	69
Классы	71
Глобальные параметры компоненты окна	71
Заголовок	71

Меню	73
Панель инструментов	73
Клиентская область	74
Распределенный шаблон процедуры для Internet	76
Установки страницы	76
Установки окна	77
Помощь	78
Управляющие элементы	79
Многодокументный интерфейс	81
Дополнительно	81
Индивидуальные переопределения настроек для управляющего элемента	84
Отображение	84
HTML	85
События	86
Классы	87
Процедурные параметры компоненты Окно (Global Window Component Options)	88
3 заголовков	88
Меню	89
Панель инструментов	90
Клиентская область	91
MDI-параметры процедуры Frame	92
Меню приложения	92
Панель инструментов приложения	92
Dynamic Html Code Template	93
Static Html Code Template	94
GetCookie Code Template	94
SetCookie Code Template	94
Cookies (Persistent Client Data)	95
AddServerProperty Code Template	96
GetServerProperty Code Template	96
Дизайн Web - приложений: соображения	97
Соображения пропускной способности	97
Используйте частичное обновление, где возможно	97
Будьте экономны с управляющими элементами	98
Используйте графику умеренно	98

Соображения о косметическом дизайне	98
Использование групп	98
Использование изображений	99
Разработка интерфейса пользователя	101
MDI window access	101
Неподдерживаемые стандартные Windows-диалоги	101
Использование параметров командной строки	102
Изменение класса для индивидуального параметра	102
Вызовы API	103
Соображения секретности	104
Использование паролей	104
Использование SSL(Secure Socket Layer)	106
Использование встроенного HTML	106
Использование ссылок на файлы во встроенном HTML-коде	108
Реализация помощи в Ваших Web-приложениях	109
Использование базового документа с внутривстраничными ссылками	109
Использование индивидуальных документов помощи	110
Управляющие элементы Windows и их HTML-эквиваленты	110
Программирование приложений вручную	114
HelloWeb - пример программы	114
Обсуждение проекта, закодированного вручную	117
Краткое руководство по IBC Library	119
Классы и их шаблонно-сгенерированные объекты	119
Краткое руководство	120
Глоссарий	125

Введение

1

В этой главе:

- ◆ введение в Clarion Internet Connect;
- ◆ обзор содержания руководства пользователя по Internet Connect;
- ◆ шрифтовые и другие соглашения в документе;
- ◆ напоминание о регистрации продукта;
- ◆ краткий обзор наших программ технической поддержки.

Добро пожаловать!

Благодарим за приобретение Internet Connect - продукта компании TopSpeed, позволяющего решить проблему создания приложений для Web за один шаг!

Эта книга содержит три руководства - для трех очень сильно различающихся уровней:

- ◆ краткое руководство "Создание приложений для Web", описывающее наиболее быстрый путь к созданию приложений баз данных для Internet-Intranet;
- ◆ руководство "Превращение существующих приложений в Web-пригодные", которое показывает, как подготовить Ваши приложения на Clarion к Web;
- ◆ руководство "Углубленная техника программирования для Web", которое показывает несколько методов расширения функциональности с использованием шаблонов и встроенного HTML.

Эта книга предполагает, что Вы закончили изучение руководства "Getting Started" для Clarion. Если Вы этого еще не сделали, мы настаиваем, чтобы Вы ознакомились с ним до начала работы с нашим руководством.

Было бы полезно также представлять, как работает Web-браузер. Применимы, но необязательны знания основ HTML.

Что такое Internet Connect?

Clarion и Clarion Internet Connect совместно создают приложение, способное работать локально (то есть под Windows, Windows 95 или Windows NT), а также в Web с использованием любого браузера с поддержкой Java.

Вы можете запустить Web-приложение на любой платформе, допускающей работу браузера с поддержкой Java.

Internet Connect и Среда разработки Clarion

Среда разработки функционирует на нескольких уровнях и поддерживает каждый уровень пользователя/разработчика.

Автоматический разработчик приложений для Windows или Web

Когда необходимо “простое” приложение для ведения базы данных, Вы можете создать его буквально за минуты, используя Clarion. Секрет - в словаре базы данных. Если генератор приложений “знает”, какие файлы и таблицы Вы желаете иметь в приложении и как они связаны, он может построить приложение. Поэтому все, что вы должны сделать - это выбрать один или несколько файлов и установить (если файлов несколько) связи: “один ко многим” или “многие к одному” (если эти связи существуют).

Application Broker (брокер приложений) затем может создать полноценное приложение, и, посредством простой пометки поля в одном из диалогов Мастера, Вы можете трансформировать приложение в Web-приложение. Полученное приложение может выполняться локально или в Web, используя Clarion Application Broker (брокер приложений).

Это может сделать каждый. Только сначала надо выбрать файл из списка.

Визуальная Среда разработки для Windows или Web

В Clarion размещение управляющего элемента в окне дает Вам гораздо больше, чем в других средствах быстрой разработки приложений, которые обычно позволяют добавить интерфейсный управляющий элемент пользователя, но предполагают самостоятельное написание кода для обеспечения его функционирования. В Clarion Вы добавляете шаблон (template), который содержит

элемент, данные и исполняемый код. Это означает, что Вы не должны писать код - одно нажатие размещает полный рабочий вариант решения: управляющий интерфейсный элемент для пользователя и код, который позволяет ему выполнить его задачу. Кроме того, каждый шаблон имеет свой собственный интерфейс для настройки. При просмотре свойств шаблона Вы можете увидеть закладку “Actions”. Помечая поля, выбирая варианты из выпадающих списков или заполняя поля редактирования, Вы можете задать поведение шаблона в точном соответствии с Вашими потребностями. В процессе изучения этой книги Вы еще неоднократно будете устанавливать “Actions” для шаблонов.

После того, как Вы опишете поведение шаблона через его интерфейс, генератор приложений создаст код (исходный код на языке Clarion), который обеспечит это поведение. Используя шаблоны, Вы можете выполнить устрашающие объемы программистской работы , не написав вручную ни единой строки кода.

Этот пример распространяется на приложения, создаваемые для Web. Вся упомянутая функциональность используется при интерпретации Вашего приложения браузером. Проверка корректности обновления записи при одновременном доступе к ней нескольких пользователей и ссылочной целостности автоматически включается в Ваше приложение, и похожим образом выполняется при работе через Web. Дополнительные параметры для Internet позволяют Вам управлять обработкой событий таким образом, что Вы имеете возможность определять, при каких условиях событие произойдет на сервере.

Квалифицированные пользователи, не имеющие возможности писать программы, легко смогут все это делать.

Это совершенный язык программирования

На базовом уровне Clarion Вы можете создавать приложения, целиком написанные вручную. Clarion - это язык четвертого поколения. У него есть высший уровень абстракции, имеющий компактную и очень хорошо читаемую грамматику для баз данных, которая позволяет Вам легко обработать запись или наборы записей. Вы не обязаны знать язык Clarion для создания приложений ... но знание того, как он работает, позволит Вам понять, что делает Ваше приложение, и поможет в создании лучших приложений.

Библиотека Internet Builder Class открыта и пригодна для приспособления к Web проектов, написанных вручную.

Профессиональные разработчики по достоинству оценят язык Clarion. Он с самого начала проектировался как язык для разработки бизнес-приложений. В дополнение к сравнительно небольшому времени изучения, Вы получите блестящую производительность. Компилятор TopSpeed превратит весь ваш код, писали ли Вы его вручную или он был написан генератором приложений, в высокооптимизированный машинный код.

Не имеет значения, на каком уровне вы собираетесь работать, - значительно легче будет начать работу, прочитав эту книгу до конца.

Что Вы найдете в этой книге

Ниже перечисляются части этой книги и их краткое содержание:

- Введение** Глава первая - которую Вы читаете сейчас. Это введение в программирование для Web с использованием Clarion.
- Установка** Глава вторая: инструкции “шаг за шагом” - как установить Clarion Internet Connect в Вашу среду разработки Clarion .

Clarion Application Broker

Глава третья: описывает Clarion Application Broker (брокер приложений) - ключевой продукт для запуска Web- пригодных приложений через Web.

Руководство по Application Wizard

Глава четвертая: несколько быстрых шагов с Мастером Приложений позволят Вам создать полноценное Web-приложение за пять минут.

Эта глава покажет, как использовать Мастер Приложений и установить некоторые параметры для Internet с целью создания простого приложения, работающего как локально, так и через Internet. Вы выполните все это, не написав вручную ни единой строки кода.

Приспособление к Web существующих приложений

Глава пятая: использование шаблонов IBC для приспособления

Clarion-приложений к Web.

Передовая техника программирования для Web

Глава шестая: представляет возможности, предлагаемые шаблонами IBC. Она проведет Вас через модифицирование Вашего приложения для достижения оптимальной производительности и функциональности при работе в Web.

Использование шаблонов Internet Builder Class

Глава седьмая: справочник к интерфейсу шаблона.

Обсуждение разработки приложения

Глава восьмая: советы и техника разработки приложений для Web.

Internet Builder Class Library - справочное руководство

Глава девятая: краткое руководство к реализации в шаблонах объектов из библиотеки IBC. Эта глава перечисляет свойства и методы, обычно используемые в Web-приложениях.

Глоссарий

Глоссарий терминов.

Где найти дополнительную информацию

Первое место для поиска дополнительной информации - это раздел How do I ...? в Help-e. Новички Clarion найдут здесь ответы на многие вопросы. Нажмите на How do I ...? на странице Help Contents для обращения к этому разделу.

Reference (глава третья) - это руководство по инсталляции, конфигурированию и использованию Брокера приложений Clarion.

Internet Builder Class Library Reference (на CD, в формате .PDF) - это полное руководство по классам, используемым шаблонами IBC. Оно представляет описание ко всем классам, методам и свойствам, с примерами к каждому.

Внимание: если какая-либо из частей Online-Help конфликтует с печатной документацией, то преимущество имеет информация в Online-Help.

Корпорация TopSpeed делает все возможное, чтобы информация в печатной документации соответствовала текущему моменту. Однако время, требуемое для ее подготовки, вызывает запаздывание документации в сравнении с информацией, содержащейся в Online-файлах, поставляемых с очередными версиями продукта.

Печатные материалы будут поступать “вдогонку”.

Соглашения в документации

Соглашения по написанию шрифтов:

<i>Italics</i>	показывает, что набирать на клавиатуре, например, Enter This.
Small Caps	отмечает нажатие клавиш на клавиатуре, например, Enter или Escape, или Click для мыши.
BoldFace	отмечает команды или параметры из выпадающего меню, или текст в окне диалога. Заметим: этот стиль также использует другое начертание шрифта для совпадения с helvetica bold face, который Windows использует как системный шрифт.
Letter Gothic	используется для диаграмм, листингов исходного кода, для аннотации к примерам и для примеров использования фрагментов исходного кода.

Соглашения о клавиатуре:

F1	означает единичное нажатие на клавиатуре. В данном случае нажать и отпустить клавишу F1.
Alt+X	означает комбинацию нажатий клавиатуры. В данном случае нажать клавишу Alt и, не отпуская, нажать клавишу X. Затем отпустить обе клавиши.

Информация о продукте

Регистрация данного продукта

До того, как Вы начнете использовать Clarion Internet Connect, заполните и отошлите по почте регистрационную карту, находящуюся в комплекте поставки. Эта карта (Business Reply Card) даст Вам возможность пользоваться службами технической поддержки TopSpeed и получать информацию о новых продуктах и сигналы о возможностях произвести обновление версий.

Техническая поддержка

Вы можете получить свободную неограниченную техническую поддержку по Clarion Internet Connect на CompuServe Information Service или в Internet. Служащие компании TopSpeed, а также TopSpeed Certified Support Partners (группа добровольцев из пользователей Clarion, известная как Team TopSpeed), ответят на ваши вопросы в оперативном порядке.

- В CompuServe - GO TOPSPEED.
- В Internet используйте любой NewsReader для:
 - доступа к группе новостей comp.lang.clarion в USENET.

- доступа к внутренней группе новостей TopSpeed. Подключитесь к серверу новостей tsnews.clarion.com и подпишитесь на группу новостей TopSpeed.products.ic (или на любую подгруппу).

Вы можете также посетить Web-страницу TopSpeed по адресу <http://www.TopSpeed.com> для получения списка других источников информации о Clarion.

Существует также платная телефонная техническая поддержка от компании TopSpeed (954) 785-4556. Запрашивайте службу работы с покупателями по телефону (800) 354-5444 или по телефону (954) 785-4555 - для получения дополнительной информации о различных предложениях по программам поддержки компании TopSpeed.

Установка

2

Системные требования

Система разработки

Вы можете запустить среду разработки Clarion на любой системе, удовлетворяющей минимальным системным требованиям для Windows 95 или Windows NT. Web-пригодные приложения должны быть откомпилированы в 32 разряда, поэтому Вы должны использовать Windows 95 или Windows NT.

- Windows 95 рекомендуется 16 или больше мегабайт RAM.
- Windows NT рекомендуется 32 или больше мегабайт RAM.
- Минимум от 7 до 13 мегабайт свободного дискового пространства, в зависимости от параметров SetUp, которые Вы выберете.

Система клиента

Система клиента может быть запущена на любой платформе, которая поддерживает работу Java-совместимого браузера. Приложения, которые Вы разработаете с использованием Clarion Internet Connect, будут успешно выполняться на компьютерах, удовлетворяющих минимальным требованиям для этих браузеров. Производительность зависит от скорости соединения в Internet, но большинство приложений хорошо выполняется через модемное соединение 28,8 kb.

Рекомендуемые браузеры

Хотя Web-пригодные приложения должны работать под любыми Java-совместимыми браузерами, лучшие результаты показывают следующие браузеры:

Windows 95/NT

Microsoft Internet Explorer 3.0x или более поздние (в версии 3.02 введено усиление виртуальной машины Java, которая улучшает распределение памяти).

Netscape Navigator 3.0 или более поздние.

Netscape Communicator 4.0 или более поздние (в версию 4.04 внесены исправления, которые улучшили производительность).

Windows 3.1x

Microsoft Internet Explorer 3.0x или более поздние.
Netscape Navigator 3.0 или более поздние.
Netscape Communicator 4.0 или более поздние.

Unix

Существует много вариантов UNIX , и некоторые из них имеют броузеры, написанные специально для них. Их слишком много, чтобы перечислять их здесь. Для получения более полной информации смотрите Home Page компании Netscape по адресу <http://www.Netscape.com> .

Netscape Navigator 3.01 или более поздние.
Netscape Communicator 4.03 или более поздние.

Apple/Macintosh

Microsoft Internet Explorer 3.0x или более поздние.
Netscape Navigator 3.0 или более поздние.
Netscape Communicator 4.0 или более поздние.

OS/2

Netscape Communicator 4.0 или более поздние.

Server System

Вы можете запустить Брокер приложений и Web- приложение на Windows 95, однако мы строго рекомендуем Windows NT для размещения рабочего комплекса.

- Для Windows 95 рекомендуется 32 мегабайт RAM и статическое соединение с Internet/Intranet.
- Для Windows NT рекомендуется 64 мегабайт RAM и статическое соединение с Internet/Intranet.

Производительность зависит от скорости канала подключения Вашего сервера с Internet и трафика, который Вы предполагаете при использовании Вашего приложения. Приложение может быть доступно через модемное соединение со скоростью 28,8 kb, но мы рекомендуем ISDN и выше.

Программа установки

Программа установки разворачивает и копирует файлы Clarion на Ваш жесткий диск. Для всех операционных систем она предоставляет Вам возможность выбора установки различных компонент, таких как файлы примеров.

Начало установки

Для запуска программы установки в Windows 95 или Windows NT:

1. Вставьте инсталляционный CD в Ваш CD-ROM.
2. Из меню Start выберите Settings > Control Panel.
3. Выберите Add/Remove Programs, затем нажмите кнопку Install.

Windows 95/NT Withard проведет Вас по процессу инсталляции.

Параметры установки

После запуска Setup Вы увидите набор экранов мастера, содержащих некоторое число параметров.

1. Выбирайте параметры для установки, предлагаемые экранами мастера (такие как диск и директория установки) и перемещайтесь от экрана к экрану, используя кнопку Next.

Программа установки установит основные компоненты Clarion Internet Developer's Kit в соответствующие поддиректории определенной Вами директории. Программа установки Clarion устанавливает все файлы в указанную директорию и в поддиректории, лежащие ниже ее. Она не устанавливает файлы в другие директории, за одним исключением. Версия ISAPI Application Broker-а требует, чтобы один файл (CWISAPI.DLL) был установлен в директорию \WinNT\System32 на компьютере с Windows NT, или \Windows\System на машине с Windows 95.

В процессе инсталляции индикаторы выполнения процессов отображают, как программа установки копирует файлы.

Нажмите кнопку ОК, когда программа установки закончит свою работу.

Регистрация набора шаблонов IBC

Для того, чтобы зарегистрировать шаблоны IBC из Internet Developer's Kit, зарегистрируйте ICONNECT.TPL в Clarion Template Registry. Файл устанавливается в поддиректорию ..\TEMPLATE.

Для регистрации набора шаблона Internet Developer:

1. Выберите Setup>Template Registry из меню Clarion.
2. Нажмите кнопку Register из диалога Template Registry.
3. В диалоге Template Registry выберите файл ICONNECT.TPL из директории ..\TEMPLATE, после чего нажмите ОК.
4. Нажмите кнопку Close.

Брокер приложений

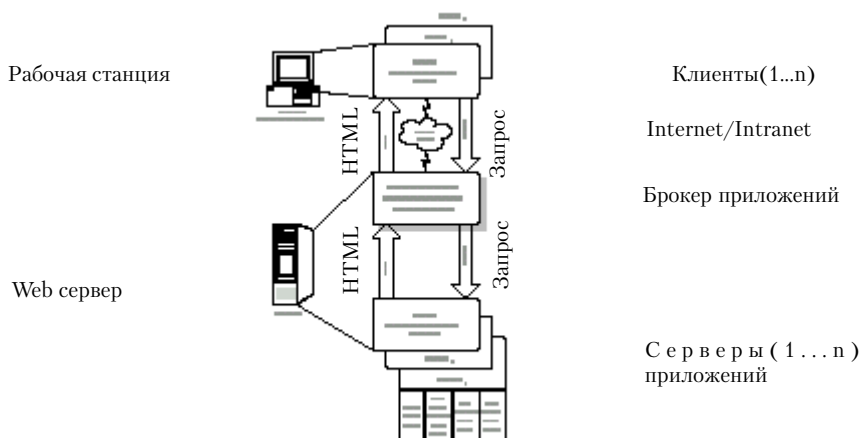
3

Эта глава содержит описание Application Broker (брокера приложений), обеспечивающего Вам связь с приложениями, созданными при помощи Internet Connect.

Internet Developers Kit состоит из двух частей: IBC Templates/Library и Application Broker.

Шаблоны (templates) делают Web-пригодными Ваши Clarion-приложения, а Брокер приложений разрешает доступ к ним через Web.

При использовании Брокера приложений Clarion отсутствует необходимость в каком-либо дополнительном программном обеспечении для Web-сервера. При этом Вы позволите конечным пользователям при помощи браузера манипулировать Web-приложениями, загруженными на сервере.



Запуск брокера приложений

Есть две версии Брокера приложений- в виде исполняемого модуля (.EXE) и версия ISAPI DLL (.DLL). Первый вариант инициируется запуском исполняемого модуля. Версия ISAPI вызывается по требованию. Для получения дополнительной информации смотри раздел “Версия ISAPI брокера приложений”.

1. Запустите брокер приложений двойным нажатием мыши на машине-сервере. По умолчанию он использует порт 80. Если Вы запустили какой-либо другой Web-сервер, использующий порт 80, то можете определить другой порт (например 8080) для Брокера приложений - в параметрах установки или передавая порт в качестве параметра при вызове из командной строки.

Внимание: Если Вы определите номер порта в командной строке, Вы не сможете изменить его в параметрах установки брокера. При запуске брокера приложений разрешите ему предупреждать Вас, если он не может использовать порт, определенный по умолчанию, затем измените параметры порта, после чего он запомнит эти установки и не будет требовать определения порта при очередном запуске.

Использование версии на одно соединение

Internet Developer's Kit содержит версию брокера приложений на одно соединение (CWBrokr1.EXE). Он содержит также ISAPI- версию брокера приложений на одно соединение.

Версия брокера приложений на одно соединение полностью совпадает по функциям с полной версией брокера приложений, за исключением того, что разрешает доступ только к одному IP-адресу за один сеанс. Это означает, что один пользователь способен запустить несколько копий одного исполняемого модуля или несколько других, открывая новые окна броузера.

Для того, чтобы обеспечить доступ другому пользователю, все приложения, запущенные первым, должны быть закрыты.

Пользователь должен выйти из приложения, чтобы приложение закрылось; простое закрытие броузера не закрывает приложения. Если пользователь закроет броузер, приложение не закроется, пока не истечет timeout. По умолчанию шаблоны устанавливают timeout для приложений 10 минут. Вы можете изменить этот параметр в Global Internet Application Extention Template.

Вы можете тестировать многопользовательский доступ (т.е. моменты в приложении, связанные с одновременным доступом к приложению более чем одного пользователя) с помощью однопользовательской версии брокера приложений, запуская два экземпляра броузера и запуска в каждом из них своей копии приложения.

Для запуска нового экземпляра Вашего броузера:

В Internet Explorer 3.0x:

1. Выберите File > NewWindow.

В Internet Explorer 4.0x:

1. Выберите File > New > Window.

В Netscape Navigator 3.0x:

1. Выберите File > NewWebBrowser.

В Netscape Navigator 4.0x:

1. Выберите File > New > NavigatorWindow.

Подключение к Вашему приложению

Для запуска приложения из броузера обеспечьте Uniform Resource Location в следующем формате :

`http://nnn.nnn.nnn.nnn/appname.exe.0`

где appname.exe- имя Вашего исполняемого файла, а nnn.nnn.nnn.nnn -Ваш IP-адрес (не забудьте добавить (.0) после имени исполняемого модуля), или

`http://domain.ext/appname.exe.0`

если Ваш IP-адрес имеет имя, зарегистрированное на Domain Name Server.

Если Вы запустили broker через порт, отличный от порта 80 (HTTP-порт по умолчанию), пользователи должны включить адрес порта в URL.

Примеры:

http://nnn.nnn.nnn.nnn:8080/appname.exe.0
http://domain.com:8080/appname.exe.0

Использование гиперссылок для запуска приложений

Вы можете также создать гиперссылки на другую Web-страницу для запуска Web-пригодных приложений.

Примеры:

<HREF="http://nnn.nnn.nnn.nnn/appname.exe.0">Click to start app

или

<HREF="http://nnn.nnn.nnn.nnn:8080/appname.exe.0">Click to start app

Firewalls и альтернативные порты

Firewalls и проху-сервера могут быть сконфигурированы с ограничением доступа к портам, отличным от порта по умолчанию (80). Для обеспечения доступа пользователям, находящимся за firewall и проху-серверами, у Вас есть следующие возможности:

1. Использовать ISAPI-версию брокера приложений и запустить Ваш web-сервер через порт 80.
2. Запустить брокер приложений через порт 80.
3. Поручить сетевому администратору удалить ограничение на порт 8080(или на порт, который Вы используете).

Версия ISAPI брокера приложений

Internet Connect содержит также Application Broker, который использует расширение сервера. ISAPI (Internet Server Application Programming Interface). Вы можете использовать эту версию с Microsoft Internet Information Server(IIS) или другими ISAPI-совместимыми web-серверами.

Эта версия брокера приложений Clarion разработана для обеспечения секретности и использует шифрование Secure Socket Layer (SSL). Здесь применяется мощная технология для шифрования данных, передаваемых через Internet. Непосвященному невозможно расшифровать информацию. Когда Вы

видите золотую иконку “замок” в строке состояния Вашего броузера - знайте, что Вы находитесь на секретном уровне, и все отправляемые и получаемые данные шифруются.

Примечание: Personal Web Server не обеспечивает SSL шифрование, поэтому секретность не будет обеспечена, когда Вы используете ISAPI Брокер приложений Personal Web Server. Именно это позволяет разрабатывать и запускать приложения на машине с Windows 95 и размещать его позже на компьютере с Windows NT под IIS. Personal Web Server можно переписать с www.microsoft.com

Как он работает

Брокер приложений Clarion выполнен в виде ISAPI Server extention Dynamic Link Library(DLL). Запрос от клиента(броузера) производится к web-серверу, определяя базовый DLL и web-приложение для выполнения. После этого в память загружается брокер приложений, и он поддерживает связь между клиентом и серверным приложением так же, как это делает версия с исполняемым модулем.

Когда Ваше web-приложение производит секретный запрос (это определяется для процедуры в Internet Options), брокер приложений принимает запрос через уровень secure sockets и остается в секретном режиме до тех пор, пока процедура активна.

Шифрование надо использовать разумно. Secure Sockets Layer (SSL) замедляет производительность между HTTP-серверами и броузерами. По этой причине Internet Connection позволяет Вам переключаться между нормальным и секретным режимами в любой момент работы в приложении. Использование шифрования только в те моменты, когда это необходимо, очень сильно повысит производительность.

Установка и подготовка к работе Clarion Application Broker (брокера приложений Clarion) ISAPI DLLs

1. Установите Microsoft Information Server (IIS) или Personal Web Server (PWS).
2. Получите Secure Sockets Layer (PCT/SSL) Digital Certificate и установите его. Вы можете получить сертификат по адресу <http://www.verisign.com>. Для тестов предлагается свободный пробный сертификат (free trial certificate). За подробной информацией по установке сертификата обращайтесь к файлу помощи для IIS Key

Manager.

Примечание: Вы можете использовать ISAPI DLLs без обеспечения SSL. Если Вы не хотите использовать SSL, установите PWS-версию Брокера приложений и пропустите шаг 2. PWS не поддерживает SSL или digital certificates.

3. Установите ISAPI DLL-версию брокера приложений Clarion. Программа установки предлагает Вам определить следующее:

Устройство для установки

Устройство на которое будет установлен брокер.

Корневая директория для установки

Корневая директория для установки. Это не должна быть корневая директория Вашего web site.

Пароль на брокер приложений

Это пароль, необходимый для доступа к настройке брокера. Вы должны его обеспечить.

IP-адрес или Domain Name сервера

Domain Name сервера или его IP-адрес.

Используемый сервер

Тип web-сервера, с которым будет работать программа установки.

Варианты для выбора - Personal Web Server или Internet

Information Server. Если Вы не хотите устанавливать Digital Certificate, выберите PWS.

Программа установки создает эти директории внутри корневой установочной директории:

- \scripts
- \sec_scr
- \secure
- \public
- \exec

Инсталляционная программа также создает текстовый файл (CWISSET.TXT), перечисляющий созданные директории и установки, которые Вы должны сделать на Вашем Web Server-е.

4. Используя утилиту IIS Internet Service Manager или PWS Administrator (из Control Panel), установите права доступа для директорий, перечисленных в CWISSET.TXT.

Например, при использовании C: как устройства для инсталляции и CWICWeb - как корневой инсталляционной директории, Вы должны установить следующие права в IIS:

<u>Directory</u>	<u>Alias</u>	<u>Rights</u>
C:\CWICWeb\public	/cwpub	Read
C:\CWICWeb\secure	/cwsec	Read & SSL
C:\CWICWeb\scripts	/cws	Execute
C:\CWICWeb\sec_scr	/cwss	Execute & SSL
C:\CWICWeb\exec	none*	none*

Примечание: Установки для SSL нужны только при использовании digital certificate.

5. Используя утилиту User Manager (для NT), предоставьте пользователю Internet Guest права WRITE и DELETE в директориях \PUBLIC и \DELETE.

Примечание: При использовании Personal Web Server нет необходимости в установке прав доступа для пользователя.

6. Поместите Ваше web-приложение в директорию C:\CWICWeb\Exec (или в директорию ниже ее). Вместе с ними Вы должны поместить в ту же директорию все необходимые для работы приложения файлы .DLL (драйверы файлов, runtime библиотеки и др.).

Примечание: 32-х-разрядная Clarion Runtime Library (CW2RUN32.DLL), 32-х-разрядный DOS-файл-драйвер (CW2dos32.dll) и 32-х-разрядный ASCII-файл-драйвер (CW2ASC32..DLL) также необходимы, даже если Вы не используете файлы этого типа в Вашем .APP (если только приложение не скомпилировано как Local Link .EXE-модуль).

7. Разместите все необходимые для Вашего приложения файлы пиктограмм в C:\CWICWeb\Public. В эту же директорию можете поместить и .GIF или .JPG файлы для Вашего приложения. Web-приложение будет автоматически извлекать файлы изображений, но если они размещены здесь, то этот шаг может быть пропущен. Смотрите Using Images в главе Application Design Considerations. Пиктограммы, отображаемые в списках или на кнопках, должны быть размещены в

этой директории.

8. Создайте Вашу Web-страницу для вызова Ваших Web-приложений, используя следующее соглашение для гипер-связей:

Click to start app

Утилита установки для ISAPI Broker

Для запуска утилиты установки брокера приложений:

1. Обратитесь из броузера к следующему URL:

http://nnn.nnn.nnn/cws/cwlaunch.dll/AppBroker/

http://domainname.com/cws/cwlaunch.dll/AppBroker/

Примечание: строка должна содержать завершающий бэк-слэш.

Появляется диалог аутентификации пользователя Броузера-а.

2. Введите в Quick имя пользователя и пароль (смотрите параметры установки брокера приложений), затем нажмите кнопку ОК.

Появится страница удаленного доступа установки брокера приложений. Эта страница содержит гиперсвязи для удаленного управления брокером.

3. Нажмите на гиперсвязь Setup. Это вызовет страницу удаленной настройки. Дополнительную информацию смотрите в Remote Access для установки брокера приложений. Это позволит Вам редактировать установки Вашего установки брокера приложений. Проверьте это сейчас, чтобы убедиться в корректности установок.

Файлы, размещаемые программой инсталляции установки брокера приложений Clarion (Application Broker)

Имя файла

CWLAUNCH.DLL

CWSECURE.DLL

CWISAPI.DLL

CLARION.ZIP

CLARION.CAB

*.class

Директория

C:\CWICWeb\scripts

C:\CWICWeb\sec_scr

\WinNT\System32or\Windows\System

C:\CWICWeb\public

C:\CWICWeb\public\

C:\CWICWeb\public

Примечание: программа инсталляции устанавливает файлы JSL в сжатом формате (Clarion.Cab и Clarion.Zip) и в развернутом виде - для поддержки 16-разрядных Броузеров и Browser для не-Windows- платформ.

Директории

Версия EXE

При запуске EXE-версии установки брокера приложений директория, из которой он запускается - это виртуальная корневая директория для исполняемых модулей, а директория \Public - это виртуальная корневая директория для всех получаемых файлов. Например, ссылка на HTML-документ в директории \Public должна адресоваться как:

<http://nnn.nnn.nnn.nnn/index.htm>,

- в то время как на исполняемый модуль, находящийся в той же директории, что и установки брокера приложений, ссылка определяется похожим URL:

<http://nnn.nnn.nnn.nnn/appname.exe.0>

Хотя эти файлы находятся в разных директориях, брокер приложений управляет маршрутизацией и не требуется никакой информации о директориях.

Если Вы хотите хранить информацию в разных директориях, можете вызывать приложение, используя его относительный адрес. Например, в случае исполняемой версии брокера, запущенного в C:\CWICWeb и приложения с именем myapp.exe, размещенного в C:\CWICWeb\Exec\Appl, Вы можете вызвать его, используя:

<http://mydomain.com/exec/appl/appname.exe.0>

Версия ISAPI

При использовании ISAPI-версии Application Broker-a, директории и их алиасы определяются в параметрах настройки и в установках web-сервера.

Если Вы хотите хранить приложение в нескольких директориях, вызывайте приложение, используя его относительный путь. Если, например, мы используем ISAPI-версию broker-a с C:\CWICWeb\Exec в качестве пути к исполняемым файлам, и приложение с именем myapp.exe, размещенное в C:\CWICWeb \Exec\Appl, то вызывать это приложение следует используя:

<http://mydomain.com/cws/cwlaunch.dll/app1/apname.exe.0>

Параметры установки Application Broker (брокера приложений)

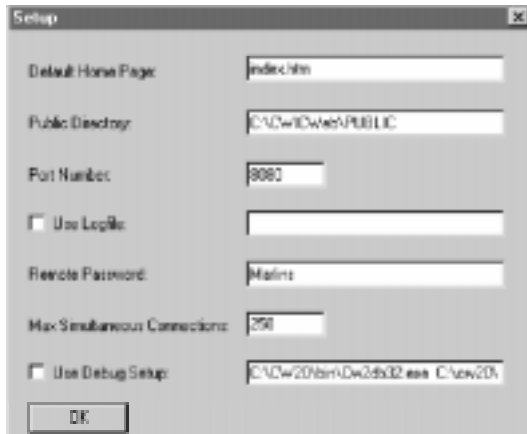
В Setup для установки брокера приложений Вы можете определить следующие параметры:

- Default Home Page** Документ, который предоставляется по умолчанию, если не определен конкретный URL. По умолчанию - это index.htm.
- Public Directory** Директория, в которой размещены совместно используемые файлы (файлы HTML, изображения и т.д.). Это также директория, в которой создаются временные директории, содержащие HTML-файлы, представляющие приложения. По умолчанию это /Public.
- Port Number** HTTP-порт, к которому "привязан" Application Broker. Если он определен в командной строке Broker-a, то эта опция отключается и не может быть изменена.
- Если порт отличен от 80 (HTTP порт по умолчанию), то клиенты (или Ваши гиперсвязи) должны определять номер порта в URL. Например, если broker подключен к порту 8080, Вы должны определить:
http://nnn.nnn.nnn.nnn:8080/appname.exe.0
- Use Log File** Включение этого признака разрешает трассировку сервера. Для каждого запроса, сделанного клиентом через Application Broker, создается запись в файле трассировки. Эта запись содержит IP-адрес источника запроса, дату и время запроса и сам запрос. Запомните, что файл трассировки непрерывно растет.
- Log File Name** Определяет имя файла трассировки.
- Remote password** Определяет пароль, разрешающий удаленный доступ к Application Broker. За дополнительной информацией обращайтесь к разделу "Удаленный доступ к брокеру приложений".
- Max Simultaneous Connections**

Определяет максимальное число одновременных подключений. Если пользователь пытается получить доступ к брокеру, превышая это

число, ему возвращается предупреждение.

Use Debug Setup Этот параметр определяет запуск приложений в режиме отладки. Для запуска отладчика у Вас должен быть обеспечен доступ к машине-серверу. В этом поле определите путь к файлам отладчика и переназначения (redirection). Например:
`C:\cw20\bin\cw2db32.exe c:\cw20\bin\cw20.red`



Удаленный доступ к брокеру приложений

Если Вы обеспечите пароль для удаленного доступа в Setup брокера приложений, то можно будет, используя браузер, получать доступ к брокеру приложений с любой удаленной площадки. Это позволит Вам модифицировать параметры установки, запрещать и разрешать доступ к приложениям, и смотреть статус последнего запроса.

Для получения доступа из браузера к брокеру необходимо:

1. Ввести следующий URL в поле Address Location Вашего браузера и нажать Enter.

`HTTP://domainname.ext/appbroker/`

или

`HTTP://nnn.nnn.nnn.nnn/appbroker/` ,

где domainname.ext - это имя Вашего сервера в домене, а nnn.nnn.nnn.nnn - его IP- адрес. Обязательно ставить заключающий слэш (/).

Появится диалог для ввода пароля.

2. Введите CWIC в поле определения имени пользователя и пароль, который Вы ранее определили.

Появится web-страница с гиперссылками на следующие функции:

Setup Обеспечивает доступ к модификации параметров установки. Вы можете модифицировать любой из параметров, за исключением номера порта и пароля для удаленного доступа. Эти параметры могут изменяться только на сервере.

Suspend (Запрещение) Позволяет запретить доступ к web-приложению, размещенному на сервере с брокером приложений. Если приложение находится в работе, сессия и приложение закрываются. Пользователи, работающие с программой, получают сообщение, что приложение закрыто для проведения обслуживания. Это позволяет проводить обновление версий приложений или файлов с данными.

Для запрещения доступа введите полное имя исполняемого файла (включая расширение), а затем нажмите кнопку ОК. Если приложение, для которого вводится запрещение, находится в поддиректории, то нет необходимости указывать здесь ее имя. Web-страница возвращает список приложений, которые запрещены в настоящий момент.

Enable Позволяет Вам снова сделать доступными web-приложения, размещенные на сервере с брокером приложений и имеющие статус запрещенных.

Чтобы разрешить работу с этими приложениями, введите полное имя исполняемого файла (включая расширение), затем нажмите кнопку ОК. Web-страница возвратит список все еще запрещенных приложений.

Status Отображает текущий статус приложений, запущенных на Вашем сервере с Application Broker, включая время доступа и IP-адреса клиентов, работающих с приложениями.

Порядок разработки и размещения приложений

1. Создайте приложение (или откройте существующее).
2. Превратите приложение в web-приложение, добавив Global Internet Application Extention Template. Смотрите “Использование Global Internet Extention Tem-

plate”.

3. Откомпилируйте приложение в 32-х-разрядном режиме (в local или standalone).

4. Установите и настройте Брокер приложений(CWBroker.exe, или CWBrokr1.exe, или версию ISAPI.DLL).

5. Разместите приложение и другие необходимые файлы (DLL, файлы данных, файлы изображений, и др.) в директорию, из которой будет вызываться Брокер приложений(или поддиректории).

Примечание: даже если Вы их не используете в своем приложении, необходимы (если только приложение не скомпилировано с параметром Local Link):

**32-х разрядная Clarion Rintime Library (CW2RUN32.DLL),
32-х разрядный драйвер для DOS-файлов (CW2dos32.DLL) и
32-х разрядный драйвер для ASCII-файлов (CWASC32.DLL).**

Исполняемые модули и необходимые для работы файлы должны располагаться в той же директории, что и приложение, или в директории, указанной в PATH (Windows 95), или в system path для пользователя (account) Internet Guest (Windows NT). Другими словами, все .DLL, необходимые для работы приложения, запущенного на сервере, должны быть видны приложению.

6. Инсталляционная программа создает поддиректорию \Public в директории самого брокера. Эта директория использует Брокер приложений для передачи файлов (таких как Java Classes из Java Support Library, графические или другие HTML файлы). не передает каких-либо файлов из своей собственной директории. Это предотвращает “скачивание” файлов с данными или исполняемых модулей с Вашего сервера. Ваше приложение на ходу создает HTML-файлы и автоматически размещает их во временных поддиректориях ниже \Public.

Примечание: Программа установки размещает JSL-файлы в сжатом формате (Clarion.CAB или Clarion.ZIP) и в несжатом формате - для поддержки 16-ти-разрядных browser-ов и броузеров-ов для не-Windows платформ.

В момент работы для каждого клиентского соединения создается временная директория. По завершению соединения эти директории автоматически удаляются. Изображения распаковываются в эту директорию, когда они не находятся в

директории \Public. Пиктограммы, отображаемые в списках (LIST) или на кнопках (BUTTON), должны быть размещены в директории \Public.

7. Запустите на сервере Application Broker. По умолчанию он использует порт 80. Если Вы запускаете какой-либо другой web-сервер, который использует порт 80, то для Брокера приложений Вы можете определить другой порт (например 8080), создав ярлык с номером порта в качестве параметра командной строки. Вы также можете изменить порт, к которому привязывается Application Broker, через параметры настройки.

Примечание: Если Вы определяете номер порта в командной строке, то не можете изменить его в параметрах настройки. Если Вы готовите к работе брокер приложений, то позвольте ему, если он не может использовать порт по умолчанию, предупредить Вас, дав затем Вам возможность изменить настройку порта и сохранить изменения. В результате не придется определять номер порта при каждом запуске брокера приложений.

8. Предоставьте пользователям, которые хотят иметь доступ к приложению, URL в следующем формате:

`http://nnn.nnn.nnn.nnn/appname.exe.0`

или

`http://domain.ext/appname.exe.0,`

где appname.exe - это имя Вашего исполняемого модуля. Убедитесь, чтобы пользователь добавлял (.0) после имени исполняемого модуля.

Если Вы запускаете Брокер приложений с использованием порта, отличного от 80 (порт HTTP по умолчанию), пользователи должны будут включать номер порта в URL. Например,
`http://nnn.nnn.nnn.nnn:8080/appname.exe.0`

Если Ваше приложение имеет параметры, передаваемые через командную строку, добавьте знак вопроса и сам параметр:
`http://nnn.nnn.nnn.nnn/appname.exe.0?myargument`

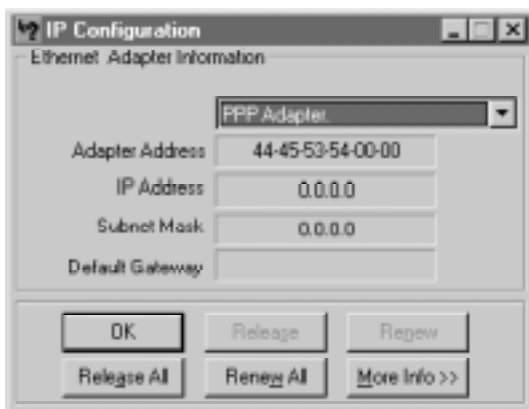
Смотрите также в разделе "Вопросы разработки приложений": "Использование параметров командной строки", "Локальное тестирование", "Тестирование Вашего TCP/IP соединения".

Локальное тестирование

Хоть для правильного размещения web-приложения Вы должны использовать постоянный IP-адрес, Вы также можете использовать IP адрес, динамически присваиваемый в момент соединения по коммутируемым линиям. Это употребимо для создания одноразовых демонстраций через Internet. Для нахождения Вашего динамического адреса Вы можете запустить applet, включенный в операционную систему.

Для локального тестирования (с брокером и броузером на одной машине), Вы можете использовать “заглушку” IP адреса: 127.0.0.1

В Windows 95 запустите утилиту Winipcfg.exe (из директории Windows). Она покажет Ваш адрес, как это показано ниже. Если Ваш TCP/IP привязан к сети и к модему, используйте выпадающий список для выбора модема.



В Windows NT используйте Dial Up Networking Manager для того, чтобы узнать Ваш IP-адрес. После дозвонки и подключения к Вашему провайдеру нажмите на пиктограмму в System Tray.

Проверка Вашего TCP/IP соединения

Существует несколько способов проверки Вашего TCP/IP соединения.

1. В приглашении DOS наберите:

PING 127.0.0.1

Это проверит TCP/IP соединение на Вашем локальном компьютере и покажет время, затраченное на ответ.

2. Запустите Ваш web браузер, введите следующий URL машины-сервера и тестовой страницы:

<http://server.domain.com/index.htm>,

где server.domain.com - это IP-адрес Вашего сервера.

Если Вы присвоили Вашему серверу порт, отличный от 80 (порт HTTP по умолчанию), Вы должны будете включить номер порта в URL, как показано ниже:
<http://server.domain.com:XX/index.htm>

где XX- номер порта, присвоенного Web-серверу.

Если у Вас нет TCP/IP соединения или Вы хотите проверить соединение на своей локальной машине, Вы можете использовать следующий URL:
<http://localhost/index.htm>

или

<http://127.0.0.1/index.htm>

Это внутренний локальный loopback (заглушка) адрес.

Руководство по созданию Web-приложений

4

В Clarion Вы можете создать приложение из словаря данных без необходимости ручного программирования. Все, что Вы должны сделать - это создать Data Dictionary (Словарь Данных) и использовать Application Wizard (Мастер Приложений) для создания полного Windows приложения - в минуты!. При установленном Internet Connect, в Application Wizard появляется добавочный check box, который позволяет превратить создаваемое Вами приложение в web-приложение. Это позволит Вам создать web-приложение с помощью единственного дополнительного нажатия кнопки мыши.

В этой главе Вы будете:

- ◆ Использовать Application Wizard для создания гибридных Web/Windows приложений из Clarion Data Dictionary, а затем выполнять программу, используя Ваш браузер.
- ◆ Компилировать и размещать приложение, запускать его через браузер.
- ◆ Оптимизировать приложение для web, используя интерфейс шаблона (template), перекомпилировать, размещать и запускать его снова.
- ◆ Модифицировать вид приложения для web, перекомпилировать, размещать и запускать его повторно.

Все это займет у Вас около 30 минут - безо всякого кодирования с Вашей стороны. В конце этой главы у Вас будет простое приложение для ввода заказов.

Давайте начнем!

Мастер Web приложений

Создание гибридных Web/Windows приложений

Отправная точка: Вы должны иметь установленную среду разработки Clarion.

Это руководство предполагает, что Вы установили Clarion в директорию C:\CW20, а Брокер приложений- в директорию C:\CWICWEB. Если Вы используете другие директории, то должны будете соответственно изменить команды. Это руководство предполагает, что Вы освоили Clarion Getting Started и имеете основные понятия о среде разработки Clarion.

Создайте Ваше первое Clarion приложение для Web.

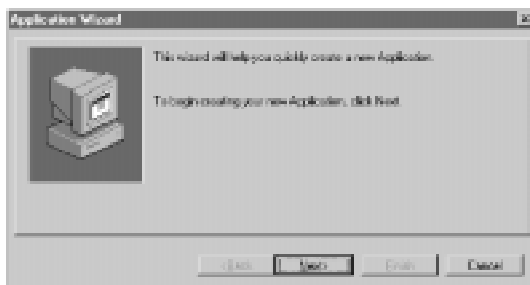
1. В диалоге Pick нажмите кнопку New...
Появится диалог New.
2. Выберите закладку (tab) Application.
3. Выберите C:\CW20\Examples\WebTutor из списка Directories.
4. Наберите WebOrder в поле File Name.
5. Убедитесь, что не отмечен признак Use Quick Start Wizard, затем нажмите кнопку Create.

Появится диалог Application Properties.

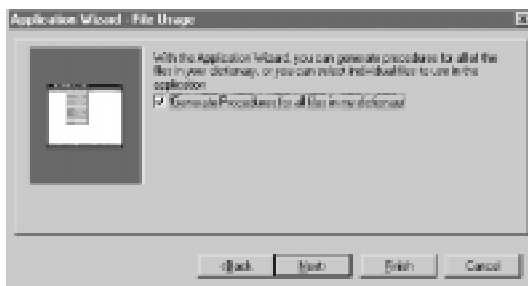
6. Нажмите кнопку (...) справа от поля ввода Dictionary File.
Появится диалог Select Dictionary.
7. Выберите C:\CW20\Examples\WebTutor\WebOrder.dct и нажмите кнопку ОК.

Запустите Application Wizard

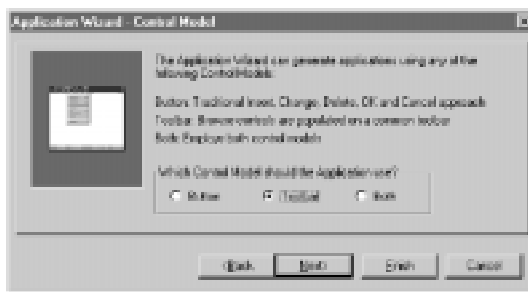
1. Отметьте признак Application Wizard, затем нажмите кнопку ОК.



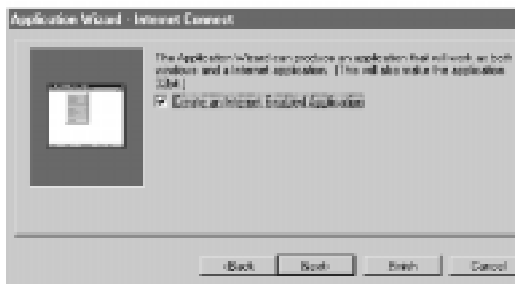
2. Нажмите кнопку Next.



3. Нажмите кнопку Next.



4. Нажмите кнопку Next.



5. Пометьте признак Create an Internet enabled application, затем нажмите кнопку Next.

Этот шаг внесет два свойства в приложение, создаваемое Application Wizard: 1) Он сделает его 32-разрядным и 2) Добавит к приложению Internet extention templates (шаблоны-расширения для Internet).



6. Нажмите кнопку Finish.
Application Wizard создаст приложение.



Создание приложения

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

Поздравления! Ваше первое web-приложение готово к размещению и запуску.

2. Нажмите кнопку OK в окне компилятора.

Размещение приложения

На последнем шаге был создан WebOrder.exe. Так как это приложение является web-пригодным, оно может работать как под Windows (как стандартный исполняемый модуль Windows), так и в Web (через Application Broker, используя browser). Теперь разместим приложение и все необходимые для его работы файлы. Заметим, что мы просто размещаем его в другую директорию на той же машине, но процесс в точности такой же, как и при размещении на машине-сервере.

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

2. Скопируйте WebOrder.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.

Запомните, что простое перетаскивание файлов в Explorer создает ярлык для исполняемых модулей. Если Вы используете метод “перетащить и опустить”, Вы должны нажать правую кнопку мыши и выбрать команду Copy из выпадающего меню.

Примечание: Мы располагаем файлы данных для примера в двух директориях. Если у Вас есть локальные файлы с данными, Вы также должны разместить их.

3. Скопируйте перечисленные ниже файлы из директории C:\CW20\BIN в директорию C:\CWICWEB\EXEC\WebTutor.

CW2RUN32.DLL
CW2TPS32.DLL
CW2ASC32.DLL
CW2DOS32.DLL

Это - DLL, поддерживающие работу Вашего приложения, включающие runtime-библиотеки и драйверы баз данных.

Этот шаг включен сюда, хотя это может и не быть необходимым при работе под Windows 95 (на Вашей машине для разработки приложений), так как пути к этим файлам содержатся в Вашей переменной PATH. Однако NT-сервер работает по-другому. У каждого пользователя есть переменная PATH и размещение вместе с приложением DLL-файлов гарантирует, что каждый пользователь получит доступ как к файлам приложения, так и к файлам поддержки. Более подробно об этом

можно посмотреть в разделе Размещение Приложений.

4. Запустите Application Broker двойным нажатием левой кнопки мыши на файле CWBrokr1.exe (или CWBroker.exe, если у Вас полная версия Application Broker) расположенном в C:\CWICWEB.

Примечание: Для этого руководства мы использовали исполняемую версию Application Broker. Версия ISAPI работает похожим образом, но с некоторыми отличиями. Они обсуждаются в главе Посредник Приложений.

5. Запустите Ваш любимый browser.

Затем проверьте работу Application Broker-а и Ваши установки TCP/IP, используя метод LocalHost loopback (заглушка):

6. В строке URL Вашего Browser-а наберите:

http://localhost/btest.htm

или

http://127.0.0.1/btest.htm,

а затем нажмите Enter.

Примечание: Если Ваш broker настроен на порт, отличный от порта 80, добавьте его номер в URL.

Например:

http://localhost:8080/btest.htm или

http://127.0.0.1:8080/btest.htm,

а затем нажмите Enter.

Если проверочная web-страница отобразится корректно, то Ваш Application Broker установлен и работает правильно. Если нет, то Вы должны вернуться к предыдущей главе и переконфигурировать настройки.



Теперь запустите приложение в браузере:

7. В строке URL Вашего Browser-а наберите:

`http://localhost/exec/webtutor/weborder.exe.0` или

`http://127.0.0.1/exec/webtutor/weborder.exe.0`

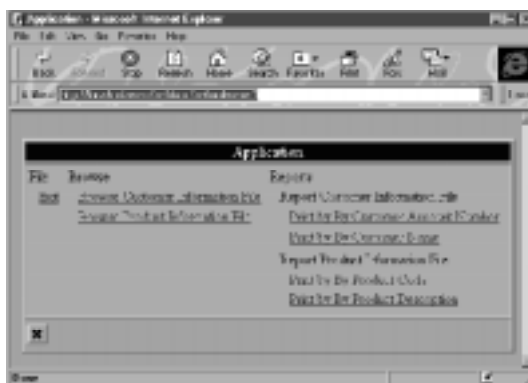
,а затем нажмите Enter.

Примечание:

Если Ваш broker настроен на порт, отличный от порта 80, добавьте его номер к доменной части URL. Например:

`http://localhost:8080/exec/webtutor/weborder.exe.0`

Поздравляем! Запущено Ваше первое web-приложение.



Сейчас Вы можете исследовать это новое приложение и сравнить его работу в web и Windows. Вы заметите, что есть несколько незначительных отличий(из-за

различий платформ), но в целом впечатления от работы приложения почти одинаковы.

8. Когда Вы закончите, нажмите на гиперсвязь Exit.

Это закроет приложение. Заметим, что browser отобразит синюю web-страницу с гиперсвязью для перезапуска приложений. Эта страница создается Application-Broker-ом автоматически, если только Вы не определили страницу для возврата из приложения в Global Internet Application Extension template (в шаблоне).

Оставьте Ваш броузер открытым с отображенной страницей перезапуска. Вы будете использовать эту страницу для перезапуска Вашего приложения.

Остаток этой главы посвящен технике оптимизации Вашего приложения для web-платформы. Это не только продемонстрирует возможности шаблона IBC, но также покажет Вам, какой мощью Вы обладаете при написании Вашего собственного кода для обеспечения некоторой “нестандартной” функциональности.

Продолжим! Пока мы только скользнули по поверхности Internet Connect.

Чем быстрее, тем лучше - оптимизация Вашего приложения

Web поднял одну проблему программирования - ограниченная пропускная способность. Очень важно использовать все доступные методы для уменьшения количества данных, передаваемых по сети. Многие пользователи подключаются к web по телефонным линиям, используя модем, что является относительно медленным сетевым подключением.

Internet Connect создан для низкой пропускной способности

Clarion Internet Connect был разработан с учетом возможной низкой пропускной способности сети. Управляющие элементы Java, которые он создает, в основном не требуют обновления всей страницы при собственном динамическом обновлении в броузере клиента. Эта форма “динамического HTML” требует передачи только небольшого количества данных. Это называется “частичным обновлением”. В этом случае обновляются только управляющие элементы, которые используются для ввода/отображения вводимых данных. Поля ввода, строковые(string) элементы Java, Java элементы изображения(Image) и Java-списки (Listbox) обычно являются

доступными для динамического обновления.

По той же причине (понижение трафика) многие управляющие элементы вызывают частичное обновление. Например, выбор новой записи в списке вызывает частичное обновление, позволяя большинству управляющих элементов отобразить текущие данные.

Частичное обновление - против полного обновления

Есть однако несколько примеров, когда частичное обновление подходит к случаю, но не включено по умолчанию. Инициирование частичного обновления вместо полного, там где это употребимо - это один из лучших способов оптимизации Ваших web - приложений.

Существует много случаев, когда подходит вариант частичного обновления, но по умолчанию установлен режим полного обновления. Так делается потому, что шаблоны (templates) не могут предвидеть всех возможных вариантов и поэтому предпочитают более безопасное полное обновление более быстрому частичному.

Например, multi-sorted (сортируемый по нескольким ключам) список, который не имеет управляющих элементов на закладках (Tabs) обрабатывается быстрее, если Вы используете Individual Control Overrides (индивидуальные настройки управляющего элемента) для определения частичного обновления для момента, когда выбирается новая закладка (Tab). При этом будут обновляться только данные в списке (listbox) вместо обновления всей страницы.

Давайте посмотрим на приложение, которое мы только что создали.

1. Переключитесь назад в Ваш browser.

2. Нажмите на гиперсвязь перезапуска.

Внутри броузера появится приложение WebOrder.

3. Нажмите на гиперсвязь **Browser Customer Information File**.

В броузере появится “окно” Browse the Customer File. Заметьте, что окно содержит список (listbox) и две закладки (Tabs). Выбор закладки изменяет порядок сортировки в списке.

4. Нажмите на каждую из закладок и отследите поведение web-страницы.

Вы должны заметить, что для обновления списка была заменена вся страница. Такое поведение присваивается по умолчанию всем листам (sheets) и закладкам

(Tabs). В следующем разделе мы изменим это поведение.

5. Нажмите на синюю клавишу X в правом конце панели инструментов для закрытия окна Browse, затем на гиперсвязь Exit для выхода из приложения.

Покиньте Ваш браузер, оставив открытой страницу перезапуска приложения. Вы будете использовать это для перезапуска приложения после того, как сделаете некоторые изменения.

Internet Procedure Extention Template (шаблон расширения для Internet)

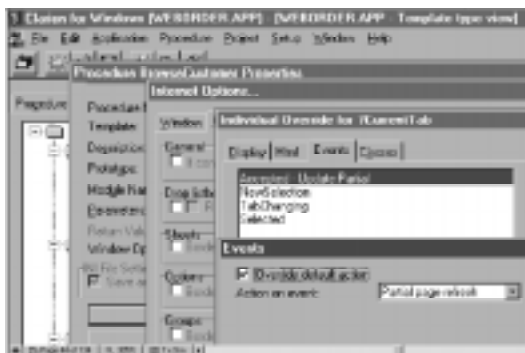
В этом разделе мы переопределим поведение управляющего элемента SHEET с целью оптимизации его производительности при работе в web.

Исходная позиция: У Вас должна быть загружена среда Clarion, и в ней открыт файл weborder.app .

1. В дереве приложения (Application Tree) выберите закладку (tab) Template. Это отсортирует процедуры по типам шаблонов (template). Заметим, что четыре процедуры используют шаблон Browse.
2. Выберите процедуру BrowseCustomer и нажмите кнопку Properties. Появится окно Procedure Properties.
3. Нажмите кнопку Internet Options.
4. Выберите закладку (tab) Controls.
5. Выберите управляющий элемент типа Sheet (?CurrentTab) в списке Individual Control Options (индивидуальные параметры элемента).
6. Нажмите кнопку Properties (свойства), затем выберите закладку (tab) с названием Events (события).
7. Выделите событие Accepted, затем нажмите кнопку Properties.

Переопределение полного обновления (по умолчанию) на частичное обновление

1. Установите признак Override default action (переопределить действие по умолчанию), затем выберите из выпадающего списка Partial page refresh (частичное обновление страницы).



2. Нажмите кнопку ОК во всех окнах, пока не вернетесь в дерево приложения (application tree) (4 раза).

3. Повторите последние девять шагов для трех других Browse процедур.

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

Ваше web-приложение готово к повторному размещению.

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

6. Скопируйте Weborder.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.

На этот раз Вы должны разместить только приложение, а файлы DLL не меняются.

Давайте запустим приложение и понаблюдаем, как изменилось его поведение после сделанных нами изменений.

Посмотрите отличие

1. Переключитесь в Ваш браузер.

2. Нажатием на гиперсвязь Restart, запустите приложение в браузере.

3. Нажмите кнопку мыши над гиперсвязью Browse Customer Information File.

4. Нажмите кнопку мыши над каждой из закладок (tab) и наблюдайте поведение web-страницы.

Вы заметите, что список (list) переотображает данные без посылки всей страницы.

5. Выйдите из приложения.

Покиньте Ваш browser, оставив открытой страницу перезапуска приложения. Вы будете использовать это для перезапуска приложения после того, как сделаете некоторые изменения

Внешний вид очень важен - Добавление графики

Web создает для пользователя приятное, полное красок пространство. Многие web - сайты (sites) создаются с целью обеспечения одовременно содержания и притягательного интерфейса. Clarion Internet Connect поддерживает наиболее часто употребляемые в web-страницах методы использования цветов и графики.

В этом разделе мы добавим фоновое (background) изображение к страницам на которых появляются окна приложения. Это обеспечит задний план приложения и позволит визуально отследить, что на экране относится к приложению, а что - нет.

Этот раздел не посвящен обучению разработки страниц или художественным методам. Он посвящен демонстрации того как использовать интерфейс шаблонов (templates) для достижения желаемого результата.

Internet Application Extention Template (шаблон расширения для Internet-приложений)

Сначала мы добавим фоновое изображение:

Исходная позиция:	У Вас должна быть загружена среда Clarion, и в ней открыт файл weborder.app .
--------------------------	--

1. В дереве приложения, нажмите кнопку Global.

Появится окно Global Properties.

2. Нажмите кнопку Extentions.

3. Появится окно Extentions and Control Templates.

4. Выделите Internet Application Extention.

Появится стандартный Windows диалог для выбора

файла.

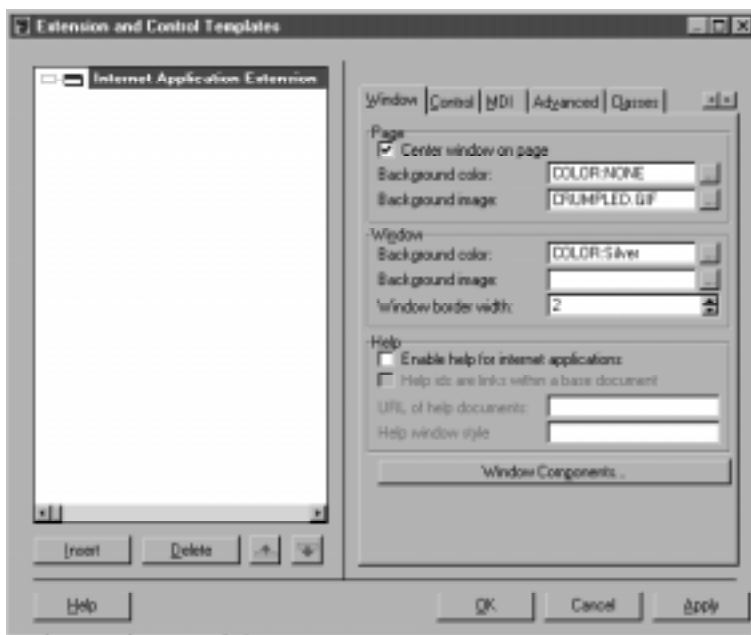
5. Выберите Crumpled.gif, затем нажмите кнопку ОК.

Добавится изображение к фону web страницы. Это - изображение скомканого листа серой бумаги. Запомните, что этот файл с изображением должен быть тоже размещен.

6. В области Window нажмите кнопку (...), расположенную за BackGround Color. Появится стандартный диалог для выбора цвета.

7. Выберите цвет Silver (серебряный), нажмите ОК.

Добавится цветовой атрибут к HTML - представлению окна приложения. Кроме добавления цвета, это будет препятствовать взгляду сквозь фоновое изображение.



8. Нажмите кнопку ОК в окнах Extentions and Control Templates и Global Properties.

Создание, размещение и запуск приложения

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

Ваше web-приложение готово к повторному размещению.

2. Откройте Windows Explorer (или Windows NT Explorer).
3. Скопируйте Weborder.exe из C:\CW20\Examples\WebTutor в C:\CWICWEB\EXEC\WebTutor.
4. Скопируйте Crumpled.gif из C:\CW20\Examples\WebTutor в C:\CWICWEB\Public..
5. Переключитесь в браузер и перезапустите приложение. Посмотрите на новый вид приложения.

В этой главе мы узнали, как создать новое приложение и сделали несколько основных изменений для оптимизации производительности работы и появления приложения при работе в Web. В следующей главе мы будем превращать уже существующее Windows приложение в web-приложение, то есть Вы научитесь публиковать в Web любые Ваши приложения.

Превращение готовых Clarion-приложений в Web-приложения

5

Перенести в web существующее приложение Clarion так же легко, как создать новое web-приложение.

В этой главе мы будем использовать WebTree.app (усеченная версия примера Tree.App, входящего в поставку Clarion). Приложение было урезано только для того, чтобы упростить руководство, обеспечить Вам быструю компиляцию и показать некоторые опции пользовательского интерфейса.

В этой главе Вы будете:

- ◆ Использовать IBC-шаблоны для переноса имеющегося Clarion-приложения в web.
- ◆ Компилировать и размещать приложение, затем выполнять его в броузере.
- ◆ Изучать использование управляющих элементов Tree (Дерево) и учиться размещать пиктограммы.
- ◆ Оптимизировать отображение Древа, применяя приемы, схожие с использовавшимися в первом руководстве.

Это должно занять около пятнадцати минут. В конце этой главы Вы будете иметь завершенное приложение для простой системы ввода заказов, с интерфейсом, отличным от использовавшегося в первом руководстве.

Давайте начнем!

Использование *Global Internet Application Extention Template* (шаблон расширения для глобальных Интернет-приложений)

Перенос приложения в web

Исходная позиция: У Вас должна быть открыта среда разработки Clarion.

В руководстве предполагается, что Вы установили Clarion в C:\CW20 и брокер приложения в C:\CWICWEB. Если Вы использовали другую директорию, Вам необходимо будет внести соответствующие изменения в команды.

Превращение Clarion-приложения в web-пригодное

1. В диалоге Pick нажмите кнопку Open...

Появится диалог Open.

2. Выберите закладку Application.

3. Выберите из списка директорий директорию C:\CW20\Examples\WebTutor, выберите WebTree.app, затем нажмите кнопку Open.

Появится Application Tree (дерево приложения).

4. В дереве приложения нажмите кнопку Global.

Появится Global Properties (глобальные свойства).

5. Нажмите кнопку Extensions.

Появится окно Extentions and Control Templates (шаблоны расширений и управляющих элементов).

6. Нажмите кнопку Insert.

7. Выделите строку Internet Application Extension, затем нажмите кнопку Select.

Тем самым добавляется шаблон Internet Application Extention, который автоматически добавляет шаблон Internet Procedure Extension к каждой процедуре приложения.

8. Нажмите кнопку ОК в окнах Extentions and Control Templates и Global Properties.

Измените на 32-бит

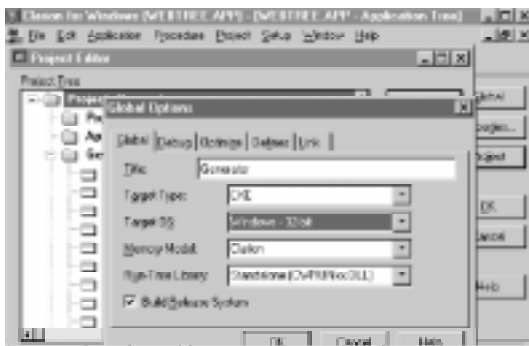
1. В дереве приложения нажмите кнопку Project.

Появится окно Project Editor.

2. Нажмите кнопку Properties.

Появится диалог Global Options.

3. В поле Target OS (операционная система) выделите Windows - 32-bit.



4. Нажмите кнопку ОК в Global Options и окнах Project Editor.

Это все, что требуется, чтобы сделать web-пригодным имеющееся приложение!

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 компиляции.

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.

5. Скопируйте все файлы пиктограмм (*.ICO) из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\Public.

Эти пиктограммы используются на кнопках панели инструментов и управляющих элементах Древа. Они должны быть помещены в директорию \Public в том порядке, в котором браузер будет их изображать. Пиктограммы в стандартной панели инструментов, которую использовало приложение из предыдущего руководства, скомпилированы в Java - классы и не нуждаются в размещении.

Запуск приложения

1. Стартуйте брокер приложения двойным щелчком на CWBrokrl.exe (или CWBroker.exe, если у Вас полная версия брокера приложения) в директории

C:\CWICWEB\.

Примечание: Как и в первом руководстве, мы будем использовать исполняемую версию брокера приложения. ISAPI-версия работает похожим образом, лишь с незначительными различиями. Они будут обсуждаться в главе Application broker.

2. Стартуйте Ваш браузер.
3. Теперь стартуйте приложение WebTree.exe в браузере.
(<http://localhost/exec/webtutor/webtree.exe.0>)

Проверка приложения

Вы должны были заметить, что это приложение по виду несколько отличается от предыдущего. Оно использует панель инструментов и не использует меню. Это обычный интерфейс для приложений web, и он продемонстрирован в примере.

1. Щелкните по кнопке Orders.

В браузере появляется “окно” Browse Customer Orders (просмотр заказов покупателей). Заметьте, что окно содержит Дерево управляющих элементов и две кнопки: для Expand All и Contract All (“Раскрыть Все” и “Сжать все”).

2. Щелкните по кнопкам Expand All и Contract All и отслеживайте, что будет происходить. Заметьте, что раскрытие и сжатие дерева обновляет всю страницу. Для оптимизации этого процесса мы будем использовать те же приемы частичного обновления, что Вы изучили в первом руководстве.

3. Выйдите из приложения (нажатием синей X).

Оставьте Ваш браузер открытым с отображенной страницей рестарта. Вы будете этим пользоваться для перезапуска Вашего приложения после внесения изменений.

Переопределение действия по умолчанию

В этом разделе мы будем переопределять действие по умолчанию управляющих кнопок для их оптимизации при выполнении через web.

Исходная позиция: В среде разработки Clarion у Вас должен быть открыт файл WebTree.app.

1. Выделите процедуру BrowseCustomer, затем нажмите кнопку Properties. Появится окно Procedure Properties.
2. Нажмите кнопку Internet Options.
3. Выберите закладку (Tab) Controls.
4. Выделите управляющую кнопку (?Expand) в списке Individual Control Options (индивидуальные параметры управляющего элемента).
5. Нажмите кнопку Properties, затем выберите закладку Events.
6. Выделите событие Accepted, затем нажмите кнопку Properties.
7. Пометьте признак Override default action (переопределить действие по умолчанию), затем выберите Partial page refresh (частичное обновление страницы) из выпадающего списка.
8. Нажмите кнопки ОК в окнах Events и Individual Overrides.
9. Выделите управляющий элемент - кнопку (?Contract) в списке Individual Control Options.
10. Нажмите кнопку Properties, затем выберите закладку Events.
11. Выделите событие Accepted, затем нажмите кнопку Properties.
12. Пометьте признак Override default action, затем выберите Partial page refresh (частичное обновление страницы) из выпадающего списка.
13. Нажимайте кнопки ОК во всех окнах до тех пор, пока не вернетесь в дерево приложения (4 раза).

Создание и размещение

1. Выберите Project > Make (или нажмите кнопку Make на панели инструментов).
Ваше web-приложение снова готово к размещению.
2. Нажмите кнопку ОК в окне компиляции.
3. Откройте Windows Explorer (Проводник) (или Windows NT Explorer).
4. Скопируйте Weborder.exe из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\EXEC\WebTutor.
5. Скопируйте все файлы пиктограмм (*.ICO) из директории C:\CW20\Examples\WebTutor в директорию C:\CWICWEB\Public.

На этот раз вам нужно только разместить приложение, пиктограммы не изменились.

Давайте выполним приложение и посмотрим, как внесенные нами изменения

повлияли на его работу.

Заметьте разницу

1. Переключитесь назад в Ваш браузер.
2. Стартуйте приложение через браузер щелчком над гиперсвязью Restart.
3. Снова щелкните по кнопке Orders.
4. Щелкните по кнопкам Expand All и Contract All и посмотрите, что происходит теперь.

Вы должны заметить, что теперь дерево переотображает данные Дерева без пересылки всей страницы.



5. Выйдите из приложения.

Поздравляем. Вы прочно стоите на пути к разработке web-приложений.

В следующей главе мы продвинемся в обсуждении средств, имеющихся в Вашем распоряжении для Internet Connect.

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.

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 Before Opening the Window (перед открытием окна).

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. 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.

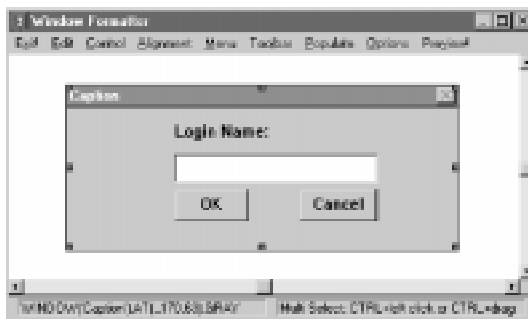
7. Выберите Populate > Control Template.

8. Выделите CloseButton и нажмите кнопку Select.

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

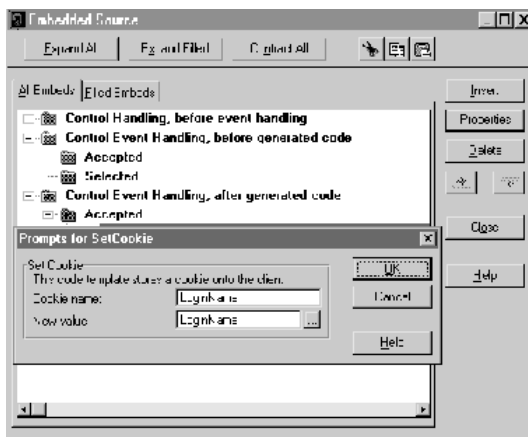
10. Измените текст на кнопке Close на ОК.

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 OK.

2. Выйдите из приложения.

3. Перезапустите в броузере 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 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\.

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.

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.

Броузер 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`.

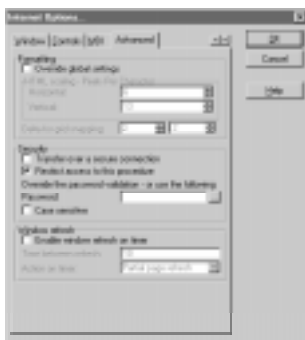
Появляется `Insert File`.

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 ОК.
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 OK 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 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\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.

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.

Шаблоны *Internet Builder Class*



Эта глава включает описание шаблонов в Internet Developer's Kit (инструментарий разработки Интернет-приложений) - Internet Builder Class (IBC) Templates(шаблоны для построения Интернет-приложений).

Шаблоны IBC состоят из одного шаблона расширения (Extention template) Global Application, процедурного шаблона (Procedure template) и нескольких кодовых шаблонов (Code templates).

Шаблон Global Internet Application Extention Template автоматически добавляет к каждой процедуре приложения шаблон расширения (Extention template). Это позволяет превратить Ваше приложение в Web - приложение за один шаг.

Комбинация глобальных и процедурных настроек обеспечивает возможность детального определения поведения системы на разных уровнях. Для определения установки для всего приложения используется глобальный параметр(Global option). Определив параметр для процедуры, Вы производите настройку только этой процедуры. Многие из глобальных и процедурных настроек одинаковы, разница только лишь в области действия настройки.

Global Internet Extention Template

Global Internet Extention (шаблон Интернет - расширение) превращает Ваше Clarion приложение в Интернет - приложение. Он прививает приложению способность генерировать динамический HTML, когда оно выполняется через Application Broker (брокер приложений).

Этот шаблон позволяет определять параметры, используемые при генерации HTML-представления Ваших окон и отчетов.

Вдобавок, он автоматически добавляет Internet Procedure Extention к каждой процедуре в приложении и к каждой впоследствии добавляемой к приложению процедуре. Это позволит переопределять для определенной процедуры многие из глобальных параметров .

Этот шаблон (Global Internet Extention) позволит Вам определить особенности появления и поведения приложения при выполнении в Web. Параметры, которые Вы здесь определите, по “природе” являются глобальными, поэтому они воздействуют на все процедуры приложения.

На процедурном уровне Вы можете переопределить большую часть этих параметров, используя настройки шаблона Internet Procedure Extention. В дополнение, некоторые параметры могут определяться на уровне обработки управляющих элементов. Комбинации этих трех уровней достаточно для обеспечения гибкости при разработке.

Примечание: Ни одна из этих настроек не работает при запуске из Windows (как исполняемый модуль).

Настройки страницы

При работе через Web текущее окно приложения отображается внутри страницы HTML (Web-страницы). Установки страницы позволяют Вам определить цвет фона и фоновое изображение для HTML-страницы.

Отцентрируйте окно на странице (Center Window on Page)

Установите этот флажок для центрирования HTML-представления Вашего окна внутри Web-страницы. Это добавляет к Web-странице признаки HTML <CENTER></CENTER>.

Цвет фона (Background color)

Вы можете определить цвет для web-страницы. Определите цвет(Color), числовой(константный) эквивалент цвета (Equate) или выберите цвет в диалоге COLORDIALOG, нажав на кнопку (...). По умолчанию цвет не установлен (эквивалент-COLOR:NONE). Это означает, что для web-страницы используется цвет, заданный по умолчанию.

Фоновое изображение (Background image)

Можно определить изображение для последующего использования в качестве фона для web-страницы. Определите имя файла с изображением или выберите его в диалоге FILEDIALOG, нажав кнопку (...). По умолчанию изображение отсутствует.

Настройки окна

При работе через Web текущее окно приложения представляется HTML <TABLE>. Это позволяет устанавливать такие параметры <TABLE>, как цвет фона и ширина обрамления. Код, сгенерированный шаблоном, вызывает для установки этих параметров метод `WebWindow.SetPageBackGround`.

Цвет фона (Background color)

Вы можете определить цвет для web-страницы. Определите цвет (Color), числовой (константный) эквивалент цвета (Equate), или выбрать цвет в диалоге `COLORDIALOG`, нажав на кнопку (...). По умолчанию, цвет не установлен (эквивалент-`COLOR:NONE`). Это означает, что для web-страницы используется цвет, заданный по умолчанию.

Совет: Вы также можете установить цвета для отдельных частей окна, таких как панель инструментов. Смотри **Параметры Компоненты Окна (Window Component Options)**.

Фоновое изображение (Background image)

Можно определить изображение для использования его в качестве фона для web-страницы. Определите имя файла с изображением или выберите его в диалоге `FILEDIALOG`, нажав кнопку (...). По умолчанию изображение отсутствует.

Совет: **Фоновое изображение повторяется (tiles) внутри элемента HTML <TABLE>, представляющего окно приложения, столько раз, сколько позволяет его размер. Для сохранения небольшого трафика используйте маленькие изображения.**

Толщина обрамления окна (Window border width)

Определите толщину обрамления окна для Вашего приложения. По умолчанию она равна 2, что создает тонкое обрамление. Для того, чтобы убрать обрамление, определите значение толщины обрамления равным 0. Сгенерированный шаблоном (template) код вызывает для установки данного свойства метод `WebWindow.SetBorderWidth`.

Помощь

Включение Помощи в Интернет-приложения (Enable Help for internet)

Установите этот флажок для разрешения связей от кнопок Help в

Вашем приложении. (Кнопка Help - кнопка с атрибутом STD:Help). Если Помощь включена, то кнопка Help будет вызывать web-страницу, базирующуюся на Help ID данного окна. Этот документ (помощь) открывается в окне браузера с названием “_HELP”, что будет вызывать открытие нового окна браузера, или если фрейм уже имеет это имя, то документ (помощь) отобразится внутри этого фрейма. Код, сгенерированный шаблоном (template), использует для установки заданных Вами параметров методы WebWindow.SetHelpDocument или WebWindow.SetHelpURL. Вы должны создать соответствующие HTML-страницы.

Смотри раздел Создание Помощи в Вашем Web-приложении.

URL документов помощи (URL of help documents)

Расположение HTML-файлов Помощи. Например, если Ваши HTML-файлы Помощи располагаются в отдельной поддиректории.

Стиль окна помощи (Help Window Style)

Вы можете определить собственный стиль для окна помощи.

Идентификаторы Help Ids являются связями внутри основного документа (Help Ids links within a base document)

Установите этот флажок, если Ваша помощь разработана как единый документ с внутривстраничными ссылками. Если признак не установлен, то Help-кнопки ссылаются на отдельные HTML-страницы.

Документ Помощь (Help Document)

Основной документ, содержащий внутри-страничные ссылки. Поле доступно только если проставлен признак Идентификаторы Help Ids являются связями внутри основного документа (Help Ids are links within a base document).

Компоненты окна (Window Components)

Нажмите кнопку для определения появления оконных компонент (таких, как TOOLBAR, MENU и области Caption (заголовок)).

Смотрите раздел Параметры Компоненты Окна (Window Component Options).

Управляющий элемент

В этой вкладке (tab) вы можете определить установки по умолчанию для генерации HTML-кода, обеспечивающего работу каждого управляющего элемента

Вашего приложения.

Совет: В добавление к настройкам, произведенным здесь, Вы можете установить параметры управляющего элемента в Интернет-параметрах процедурного шаблона. Смотрите **Индивидуальные переопределения настроек по умолчанию для управляющего элемента (Individual Overrides for a Control)**.

Основное

Если управляющий элемент отключен (If control disabled)

Параметр определяет, что отображать в браузере, когда управляющий элемент отключен. Предоставляется потому, что большинство управляющих элементов HTML не поддерживают отключения. Это устанавливает свойство `WebWindow.DisableAction`. Возможны следующие варианты:

Hide (по умолчанию)

Прячет отключенный элемент.

Спрятать, если невозможно отключить (Hide if cannot disable)

Прячет отключенный элемент, если нельзя отключить его на веб-странице. Большинство управляющих элементов HTML не могут быть отключены.

Показать (Show)

Отображает отключенный элемент. Он появляется обычным образом (т.е. доступным для обработки), но изменения, произведенные в этом поле, никак не отразятся в приложении.

Выпадающие списки (Drop listboxes)

Заменить на Java невыпадающий список

Позволяет заменить выпадающий список на загружаемый постранично список Java. Используйте эту установку, когда необходимо в выпадающем списке отображать более одной колонки.

Листы (Sheets)

Толщина обрамления (border width)

Определите толщину обрамления для управляющего элемента SHEET. По умолчанию она равна 2, что создает тонкое обрамление. Для того, чтобы убрать обрамление, определите значение толщины

обрамления равным 0. Это устанавливает свойство `WebWindow.SheetBorderWidth`.

Варианты (Options)

Толщина обрамления (border width)

Определите толщину обрамления для управляющего элемента Options. Это употребляемо только для OPTIONS с атрибутом BOXED. По умолчанию толщина равна 2, что создает тонкое обрамление. Для того, чтобы убрать обрамление, определите значение толщины обрамления равным 0. Это устанавливает свойство `WebWindow.OptionBorderWidth`.

Группы (Groups)

Толщина обрамления (border width)

Определите толщину обрамления для управляющего элемента Groups. Это употребляемо только для GROUPs с атрибутом BOXED. По умолчанию толщина равна 2, что создает тонкое обрамление. Для того, чтобы убрать обрамление, определите значение толщины обрамления равным 0. Это устанавливает свойство `WebWindow.GroupBorderWidth`.

Многодокументный интерфейс

Этот раздел определяет способ обработки меню и панелей инструментов приложения.

Совет: Для управления определенными пунктами меню или панели инструментов используйте для переопределения пункты из Интернет Параметров процедуры `Frame`.

Меню Frame

Этот раздел определяет способ обработки меню приложения. Это позволяет Вам определить, какие глобальные параметры меню отобразятся в “дочерних” окнах.

Включать в “дочерние” окна (Include on Child Windows)

Выберите вариант из выпадающего списка. Возможные варианты:

Все пункты меню (All Menu Items)

Все пункты меню появятся в “дочерних” окнах.

Нет пунктов меню (No Menu Items)

Пункты меню не появятся в “дочерних” окнах.

Игнорировать код в цикле ACCEPT процедуры Frame (Ignore code in frame's ACCEPT loop)

Установите этот флажок для того, чтобы игнорировать код для обработки пунктов меню внутри цикла ACCEPT в процедуре Frame. Если признак не установлен, то весь код выполняемый в цикле ACCEPT процедуры Frame, автоматически выполняется в “дочерней” процедуре.

Панель инструментов процедуры Frame (Frame Toolbar)

Этот раздел описывает способ обработки управляющих элементов Панели инструментов приложения. Это позволяет Вам определять, какие глобальные управляющие элементы Панели инструментов высвечиваются в “дочерних” окнах.

Include on Child Window (Включать в “дочерние” окна)

Выберите вариант из выпадающего списка. Доступны следующие значения:

All Toolbar Items

Все пункты панели инструментов появятся в “дочерних” окнах.

Standard Toolbar Only

В “дочерних” окнах появятся только “стандартные” (Standard) пункты панели инструментов. Это кнопки, добавляемые шаблоном FrameBrowseControl.

No Toolbar Items

В “дочерних” окнах не появятся пункты панели инструментов.

Ignore code in frame's ACCEPT loop (Игнорировать код в цикле ACCEPT для Frame)

Установите этот флажок для игнорирования в цикле ACCEPT(для Frame-процедуры приложения) кода обработки пунктов панели инструментов. Если признак не установлен, то код, включенный в цикл ACCEPT (для Frame), автоматически попадает в “дочернюю” процедуру.

Дополнительно

Horizontal Pixels per Char

Ширина символа, выраженная в числе точек для расчета размеров при создании апплетов Java и изображений.

Vertical Pixels per Char

Высота символа, выраженная в числе точек для расчета размеров при

создании апплетов Java и изображений.

Примечание:

Определяемые здесь значения влияют на общий вид генерируемой HTML-страницы. Например, увеличивая значение Vertical Pixels per Char, Вы увеличиваете высоту клеток HTML-таблицы.

Data for grid snapping

Число точек, рассматриваемое перед перемещением управляющего элемента. Определите значения для координат X и Y. Если управляющий элемент находится в пределах этих значений, то он не перемещается.

Page to return on exit

При желании, Вы можете определить HTML-страницу для возврата после окончания работы программы. Код, созданный шаблоном, вызовет метод `WebServer.Init` для установки свойства `WebServer.PageReturnTo`.

Time out (seconds)

Величина, определяющая максимальное количество времени бездействия (в секундах), после которого происходит закрытие приложения. По умолчанию - 600 секунд (10-секунд). Для установки свойства `WebServer.TimeOut` сгенерированный код вызывает метод `WebServer.Init`.

Subdirectory for pages

Директория, в которой приложение создает временные директории (временная директория создается для каждого активного соединения) для записи динамических HTML и графических файлов. Это также директория, в которой размещаются графические файлы. Если Вы разместите графику в этой директории, то не будет производиться ее запись во временную директорию. По умолчанию установлена директория `/PUBLIC`. Сгенерированный код вызывает для установки этого свойства метод `WebFilesManager.Init`. Не стоит устанавливать это свойство в момент работы.

Classes Local to Application Broker

Здесь определено, что файлы библиотеки поддержки Java расположены в поддиректории `/PUBLIC` основной директории брокера приложений. Если у Вас несколько серверов, то может появиться желание определить единственный источник, откуда будут задействоваться эти файлы. В этом случае Вы должны убрать этот флажок и назначить URL для файлов поддержки Java (Java Support Library). Это устанавливает свойство `WebServer.JavaClassLibrary`.

Use Long Filenames

Установите этот флажок для того, чтобы разрешить создание на Web сервере файлов с длинными именами.

Классы

Закладка *Classes* предлагает Вам определить, какие классы (объекты) используют шаблоны (*Templates*) для выполнения различных задач и модули, содержащие определения классов. Этот подход дает Вам возможность использовать столько классов, сколько Вы хотите из библиотеки IBC и столько, сколько хотите - из своих собственных классов.

Для изменения класса для определенного пункта или переназначения класса выделите его в списке и нажмите кнопку *Properties* (свойства).

Смотрите также: Переназначения класса (*Class overrides*), Глобальные параметры компоненты окна (*Global Window Component Options*).

Глобальные параметры компоненты окна

Заголовок

Это область в верхней части “окна” на HTML-странице. Это часть, представляющая заголовок окна.

Include caption

Установите этот флажок для отображения заголовка (*Caption*). Если флажок не установлен, то заголовок не используется. Это устанавливает свойство *WebWindow.CreateCaption*.

Background color

Вы можете определить цвет для использования в области заголовка. Определите цвет(*Color*), задав цветовой числовой эквивалент (*Color equate*) или выбрав цвет из *COLORIALOG*, нажав на кнопку (...). По умолчанию установлен темно-синий цвет (*Navy Blue*) (эквивалент *COLOR:Navy*). Если цвет здесь не определен, то используется тот цвет фона в окне, который Вы определяли ранее в установках параметров для окна (если Вы это делали). Если не определяли и окно приложения имеет атрибут *COLOR*, то этот цвет отображается в браузере. Для определения данного свойства сгенерированный код использует метод *WebCaption.SetBackground*.

Background Image

Вы можете определить изображение для использования его в области заголовка в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию изображение не используется. Для определения данного свойства сгенерированный код использует метод WebCaption.SetBackground.

Совет: Изображение, используемое для фона, повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Alignment

Вы можете управлять выравниванием текста в области заголовка. Варианты для выбора - это выравнивание влево, вправо и по центру. По умолчанию - по центру. Это устанавливает свойство WebCaption.Alignment.

Font Family name

Позволяет Вам определять шрифт для отображения на экране. Запомните, что браузер может отображать только те шрифты, которые установлены на компьютере клиента. Однако большинство операционных систем поддерживают замену шрифтов и будут использовать для отображения наиболее близкий шрифт. Код, сгенерированный по шаблону, использует для установки данного свойства метод WebCaption.SetFont.

Font size

При желании, Вы можете определить размер шрифта, отображаемого в области заголовка. По умолчанию не установлен (выбирается тот размер, который по умолчанию использует браузер). Для определения данного свойства сгенерированный код использует метод WebCaption.SetFont.

Font color

Вы можете определить цвет шрифта для области заголовка. Определите цвет (Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). По умолчанию не установлен и используется цвет шрифта, по умолчанию используемый браузером.

Меню

Это область меню в верхней или боковой части “окна” в HTML -странице.

Background color

Вы можете определить цвет для использования в области меню. Определите цвет(Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). Если цвет здесь не определен, то используется тот цвет фона в окне, который Вы определяли ранее в установках параметров для окна (если Вы это делали) . Если не определяли, и окно приложения имеет атрибут COLOR, то этот цвет отображается в броузере. Для определения данного свойства сгенерированный код использует метод WebMenuBar.SetBackground.

Background Image

Вы можете определить изображение для использования его в области меню в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию, изображение не используется. Для определения данного свойства сгенерированный код использует метод WebMenuBar.SetBackground.

Совет:

Изображение, используемое для фона повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Alignment

Вы можете управлять расположением меню. Возможны следующие варианты: Above Toolbar (по умолчанию), Left of Window, или Below Toolbar. Когда Вы используете Above Toolbar, меню растягивается горизонтально через верхнюю часть HTML страницы. Когда Вы используете Below the ToolBar, меню растягивается горизонтально через HTML - страницу под областью Панели инструментов (Toolbar arear). Когда Вы используете Left of Window, меню располагается вертикально слева от <TABLE>, представляющего окно приложения.

Панель инструментов

Это область панели инструментов, находящаяся в верхней части “окна” в HTML-странице (под областью заголовка).

Background color

Вы можете определить цвет для использования в области панели инструментов. Определите цвет(Color), задав цветовой числовой

эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). Если цвет здесь не определен, то используется тот цвет фона в окне, который Вы определяли ранее в установках параметров для окна (если Вы это делали) . Если не определяли, и окно приложения имеет атрибут COLOR, то этот цвет отображается в броузере. Для определения данного свойства сгенерированный код использует метод WebToolBar.SetBackground.

Background Image

Вы можете определить изображение для использования его в области панели инструментов в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию, изображение не используется. Для определения данного свойства сгенерированный код использует метод WebToolBar.SetBackground.

Create extra close button

Определяет, в каких случаях обеспечивать для окна кнопку Close. Эта кнопка - дополнение ко многим другим кнопкам окна. Она предназначена для замены системной кнопки Close, автоматически обеспечиваемой Windows, но не браузером. Если во всех Ваших окнах уже есть кнопка закрытия, у Вас нет необходимости определять еще такую же. Возможны следующие варианты выбора:

Never

Никогда не создавать дополнительных кнопок Close

If window has system menu

Создает кнопку Close тогда, когда WINDOW имеет атрибут SYSTEM и нет других кнопок.

If window has system menu

Создает кнопку Close тогда, когда WINDOW имеет атрибут SYSTEM.

Always

Всегда создает кнопку Close.

Image for close

Определяет пиктограмму отображения для кнопки Close. Определите имя файла с пиктограммой или выберите файл в диалоге FILEDIALOG, нажав кнопку (...). По умолчанию определен файл EXIT.ICO (маленькая голубая X) (поставляется с Clarion).

Клиентская область

Это область “окна” на странице HTML, представляющая клиентскую область приложения.

Background Color

Вы можете задать цвет, который будет использоваться для Вашей

клиентской области. Определите цвет (Color), цветовой эквивалент (Color equate) или выберите цвет нажатием кнопки (...) в COLORDIALOG. Если здесь цвет не был определен, то будет использоваться тот цвет, который был задан ранее в установках параметров для окна (если Вы это сделали). Если Вы не определили цвет ни там, ни здесь, и окно приложения имеет атрибут COLOR, то этот цвет и будет отображаться в броузере. Для определения данного свойства сгенерированный код использует метод WebClientArea.SetBackground.

Background image

Вы можете определить изображение, которое будет отображаться на экране как фон для клиентской области Вашего приложения. Задайте имя файла изображения или выберите файл из FILEDIALOG нажатием кнопки (...). По умолчанию, изображения нет. Для определения этого свойства сгенерированный код использует метод WebClientArea.SetBackground.

Совет:

Изображение, используемое для фона, повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Переопределение классов (Class Overrides)

Override default class (Переопределение классов по умолчанию)

Чтобы переопределить класс IBC, установите этот флажок и определите Class Name, файл Header (INC.) и файл Implementation (.CLW) в полях, описанных ниже.

Class Name (Имя класса)

Определите имя класса, который будет использоваться, или имя класса по умолчанию, если Вы хотели бы переопределить класс по умолчанию.

Header file (файл заголовка)

Определите файл заголовка (файл, содержащий объявления класса) или выберите файл из FILEDIALOG нажатием кнопки (...).

Implementation file (файл реализации)

Определите файл реализации (файл, содержащий

определения класса или исходный код) или выберите файл из FILEDIALOG нажатием кнопки (...).

Распределенный шаблон процедуры для Internet

Этот шаблон позволяет Вам доработать появление и поведение процедуры для выполнении в Web. Установки, которые Вы здесь определяете, локальны по природе, то есть они распространяются только на эту процедуру. Чтобы изменить глобальные установки (Global Settings), нажмите кнопку Global в генераторе приложений, затем - кнопку Extensions, и поменяйте установки для Internet-расширения приложения.

Чтобы поменять установки, нажмите кнопку Internet Options в окне Procedure Properties.

Примечание: ни одна из этих установок не влияет на выполнения Вашего приложения, когда оно работает локально как исполняемый модуль Windows.

Установки страницы

При работе под Web окно приложения изображается внутри страницы HTML (страницы web).

Установки страницы дают Вам возможность задать цвет фона и изображение фона для страницы HTML. Для обеспечения этих свойств сгенерированный код использует метод `WebWindow.SetPageBackground`.

Override Global Settings (переопределение глобальных установок)

Чтобы переопределить установки страницы в шаблоне Internet Application Global Extension, установите этот флажок. Установка этого флажка делает доступными для определения другие установки.

Center Window on Page (центрировать окно на странице)

Установите этот флажок, для центрирования HTML-представления Вашего окна внутри web-страницы. Это добавляет к web-странице признаки `<CENTER></CENTER>HTML`.

Background Color (цвет фона)

Вы можете задать цвет, который будет использоваться для web-страницы. Определите цвет (Color), цветовой эквивалент (Color equate), или выберите цвет нажатием кнопки (...) в COLORDIALOG. По умолчанию цвета нет (эквивалентно COLOR:NONE). Это значит, что используется цвет, установленный по умолчанию в браузере.

Background image(фоновое изображение)

Вы можете определить изображение, которое будет отображаться на экране как фон для web-страницы. Задайте имя файла изображения или выберите файл из FILEDIALOG нажатием кнопки (...). По умолчанию изображения нет.

Установки окна

При работе в Web окно приложения представлено посредством HTML <TABLE>. Поля ввода на этой закладке позволяют Вам определить появление порции “окна” страницы HTML, которая отображается при выполнении приложения в Web.

Override Global Settings (переопределение глобальных установок)

Чтобы переопределить установки страницы в шаблоне Internet Application Global Extension, установите этот флажок. Установка этого флажка делает доступными для определения другие установки.

Background Color (цвет фона)

Вы можете задать цвет, который будет использоваться для окна Вашего приложения. Определите цвет (Color), цветовой эквивалент (Color equate), или выберите цвет нажатием кнопки (...) в COLORDIALOG. По умолчанию цвета нет (эквивалентно COLOR:NONE), это значит, что используется цвет фона окна приложения. Для обеспечения этого свойства сгенерированный код использует метод WebWindow.SetBackground.

Background image

Вы можете определить изображение, которое будет отображаться на экране как фон для окна Вашего приложения. Задайте имя файла изображения или выберите файл из FILEDIALOG нажатием кнопки (...). По умолчанию изображения нет. Для установки этого свойства сгенерированный код использует метод WebWindow.SetBackground.

Совет: **Фоновое изображение повторяется (tiles) внутри элемента HTML <TABLE>, представляющего окно приложения, столько раз, сколько позволяет его размер. Для сохранения небольшого трафика, используйте маленькие изображения.**

Window border width (ширина границы окна)

Определите ширину границы для окна Вашего приложения. По умолчанию это 2, что создает тонкую границу. Чтобы границы не было, задайте границу ширины 0.

Помощь

Override Global Settings (переопределение глобальных установок)

Чтобы переопределить установки Help в шаблоне Internet Application Global Extension, установите этот флажок. Установка этого флажка делает доступными для определения другие установки.

URL of help documents

Основное место размещения файлов HTML для вашей Помощи. Например, Ваши HTML-файлы помощи размещены в отдельной поддиректории.

Help Window Style (стиль окна Помощи)

При желании Вы можете определить стиль окна Помощи.

Help Ids are links within a base document (идентификаторы Помощи являются ссылками внутри основного документа)

Установите этот флажок, если Ваша Помощь задумана как единственный документ с внутристраничными ссылками (anchors). Если он не установлен, кнопки Помощи ссылаются на индивидуальные HTML-страницы.

Help Document (документ Помощи)

Основной документ, содержащий внутристраничные ссылки. Это поле доступно, только если установлен флаг Help Ids are links within a base document.

Window Components (компоненты окна)

Нажмите эту кнопку, чтобы задать установки, определяющие появление компонентов окна (то есть, TOOLBAR, MENU и области Caption). Эти установки переопределяют любые соответствующие глобальные установки. Смотрите также Procedure Window Component Options.

Return if launched from browser (возврат, если запущен из броузера)

Закрывает процедуру, когда она выполняется в web. Эффективно делает процедуру недоступной при работе в Web.

Управляющие элементы

To Override Global settings:

Чтобы переопределить установки управляющего элемента в шаблоне Internet Application Global Extension, установите флажок слева от описания параметра. Установка этого флажка делает доступными для определения другие установки.

General

If control disabled

Определяет, что будет отображаться в броузере, когда управляющий элемент окна отключен. Такая возможность обеспечена из-за того, что большинство управляющих элементов HTML не поддерживают отключение. Это устанавливает свойство WebWindow.DisabledAction. Возможности для выбора:

Hide (no умолчанию)

Прячет все отключенные управляющие элементы.

Hide if cannot disable

Прячет любой отключенный управляющий элемент, когда его нельзя отключить на web-странице. Большинство управляющих элементов HTML не могут быть отключены.

Show

Показывает все отключенные управляющие элементы. Они нормально появляются (то есть, будут появляться как действующие), но изменения, сделанные в них, не будут действенны в разрабатываемом приложении.



Drop listboxes

Replace with Java non-drop list

Это позволяет Вам заменить выпадающий список на странично загружаемый список (Listbox) Java. Используйте этот параметр, если Ваш выпадающий список требует для отображения больше одной колонки.

Sheets

Border width

Задайте толщину рамки для управляющих элементов SHEET. По умолчанию, это 2 - для тонкой границы. Определите толщину равной 0, чтобы рамки не было. Здесь устанавливается свойство `WebWindow.SheetBorderWidth`.

Options

Border width

Задайте толщину рамки для управляющих элементов OPTION. Применяется только для элементов OPTION с атрибутом BOXED. По умолчанию это 2 - для тонкой границы. Определите толщину равной 0, чтобы рамки не было. Здесь устанавливается свойство `WebWindow.OptionBorderWidth`.

Groups

Border width

Задайте толщину рамки для управляющих элементов GROUP. Применяется только для элементов GROUP с атрибутом BOXED. По умолчанию это 2, что создает тонкую рамку. Определите толщину равной 0, чтобы рамки не было. Здесь устанавливается свойство WebWindow.GroupBorderWidth.

Individual Control Overrides

В этом месте Вы можете переопределить появление и поведение каждого управляющего элемента в окне. Выделите элемент, подлежащий изменению, и нажмите кнопку Properties. Смотрите раздел Individual Overrides for a Control.

Многодокументный интерфейс

Здесь определяется порядок обработки меню приложения(Menu) и панели инструментов(Toolbar).

Совет: для управления специфичными пунктами меню и панели инструментов установите переопределение для MDI в Интернет-параметрах процедуры Frame.

Merge Frame Menu

Установите этот флажок для подавления меню Frame при выполнении этой процедуры.

Merge Frame Toolbar

Установите этот флажок для подавления панели управления Frame при выполнении этой процедуры.

Для процедуры Frame имеются также дополнительные параметры. Смотрите Frame Procedure MDI Options.

Дополнительно

Formatting (форматирование)

Override Global Settings

Чтобы переопределить установки для форматирования в шаблоне Internet Application Global Extension, установите этот флажок. Установка этого флажка делает доступными для определения другие

установки.

Horizontal Pixels per Char

Число точек, принимаемое как ширина символа при вычислении размера для создания апплетов и изображений Java.

Vertical Pixels per Char

Число точек, принимаемое как высота символа при вычислении размера для создания апплетов и изображений Java.

Delta for grid snapping

Число точек, которое учитывается перед перемещением управляющего элемента. Задайте значение для X и значение для Y. Каждый раз при перемещении, пока элемент находится внутри этого интервала, он не перемещается.

Примечание: заданные числа влияют на общий вид созданной страницы HTML. Например, увеличение значения Vertical Pixels per Char делает ячейки таблицы HTML более высокими.

Security

Transfer over a secure connection

Если установлен этот флажок, данные пересылаются с использованием Secure Socket Layer (SSL). Это обеспечивает безопасные транзакции на уровне процедуры. Помните, что шифрование заметно замедляет выполнение программы. Вам следует обеспечивать безопасность только для процедур, которые пересылают важные данные.

Примечание: Эта возможность требует инсталляции секретной версии Брокера приложения. Смотрите главу Брокер приложений (Application Broker).

Restrict Access to this procedure

Установите этот флажок для парольной защиты процедуры и разрешения двух нижеописанных полей.

Password

Задайте пароль или выберите переменную из схемы связей файлов(file schematic) нажатием кнопки (...). Постоянный пароль обеспечивает простую защиту.

Для получения дополнительной информации смотрите Using Pass words.

Case Sensitive

Установите этот флажок для усиления проверки пароля чувствительностью на регистр символов в пароле. Если флаг не отмечен, регистр игнорируется.

Window refresh (обновление окна)

Enable Refresh on timer

Установите этот флажок для включения обновлений всей страницы или только данных на странице, базирующихся на таймере. Атрибут TIMER в WINDOW независим от этой установки. Эта установка используется в web, а атрибут TIMER - когда приложение выполняется под Windows.

Совет: это свойство следует использовать осторожно, чтобы гарантировать минимальный сетевой трафик.

Time between refresh

Определите число секунд между обновлениями.

Action on Timer

Задайте действие, которое нужно выполнить, когда наступило событие таймера.

Возможные варианты для выбора:

Partial Page refresh

Переотображает управляющие элементы Java и поля ввода HTML для обновления текущих данных.

Submit page

Посылает данные приложению на сервере и перерисовывает страницу, как предписывает серверное приложение.

Complete Page refresh

Перерисовывает всю страницу.

Индивидуальные переопределения настроек для управляющего элемента

Состав пунктов для переопределения настроек управляющих элементов основывается на типе управляющего элемента и его атрибутах. Каждое переопределение описывается здесь с пометками о возможных условных параметрах.

Override Global Settings (переопределить глобальные установки)

Установите этот флаг слева от названия для переопределения установок управляющего элемента в шаблоне Internet Application Global Extention template. После установки этого флага откроются другие параметры настройки.

Отображение

If control is disabled (если элемент отключен)

Определяет, что нужно отображать в броузере, когда управляющий элемент окна отключен. Этот параметр определяется потому, что большинство элементов HTML не поддерживают отключения. Этим устанавливается свойство IC:CurControl.DisabledAction. Возможны следующие варианты:

Hide (по умолчанию) (спрятать)

Прячет любой отключенный управляющий элемент.

Hide if cannot disable (спрятать, если невозможно отключить)

Прячет любой отключенный управляющий элемент, когда невозможно его отключение.

Show (показать)

Отображает любой отключенный элемент. Он появляется как обычно (то есть доступный для редактирования), но все изменения, сделанные в нем, не имеют эффекта в приложении.

Hide if launched from browser (спрятать, если запущен из браузера)

Установите этот флаг для того чтобы спрятать управляющий элемент при работе приложения в web. Это позволяет Вам отключать отображение некоторых данных или ограничивать функциональность для web-версии, не удаляя ее из Windows-версии.

Autospot Hyperlinks (Автораспознаваемые гиперсвязи)

Этот признак доступен для элементов List и STRING. Если он установлен, то отображаются те данные, которые содержат корректную гиперсвязь (то есть начинающиеся с http:, https:, ftp:, malto:, news:, telnet:, wais: или gopher:) для перехода по ссылке.

Allow dynamic updates (позволить динамическое обновление)

Этот параметр доступен для элементов типа STRING. Если он установлен, то строковый управляющий элемент создается на HTML-странице как строковый управляющий элемент Java, который

обновляется при каждом частичном обновлении страницы.

Примечание: Элемент типа **STRING** с переменной в качестве атрибута для **USE** автоматически становится элементом Java и не нуждается в установке этого признака. Он применим только для статических **STRING**-элементов, которые изменяются через присваивание свойств (например, `?String1{PROP:Text}='New Text'`).

Image Options (параметры изображения)

Update Image dynamically (обновлять изображение динамически)

Этот признак доступен для элементов типа **IMAGE**. Если он установлен, то управляющий элемент создается на HTML-странице как управляющий элемент Java типа **Image**, который обновляется при каждом частичном обновлении страницы.

Примечание: Элемент типа **IMAGE** с переменной в качестве атрибута для **USE** автоматически становится элементом Java и не нуждается в установке этого признака. Он применим только для статических **IMAGE**-элементов, которые изменяются через присваивание свойств (например, `?String1{PROP:Text}='New.gif'`).

Alternative text (альтернативный текст)

По желанию можно предоставить текст для отображения, пока загружается изображение. Это добавляется к признаку (tag) HTML **IMG ALT=**. Альтернативный текст отображается, пока графический файл передается в браузер (до отображения изображения), или вместо изображения, если пользователь отключил отображение изображений в настройках браузера

Border Width (толщина обрамления)

Этот параметр доступен для элементов **SHEET**, **OPTION** (если с параметром **boxed**) и **GROUP**. Определяет толщину границы управляющего элемента. По умолчанию установлено значение 2, определяющее тонкую границу. Для полного отключения границы определите значение 0.

HTML

Одной из наиболее мощных возможностей шаблонов (template) IBC является возможность встраивания HTML-кода в HTML-страницы, порождаемые web-приложениями. Эта возможность позволяет Вам добавлять HTML - код перед или

после элементов на результирующей web странице. Этот код не работает при запуске приложения через Windows.

Используя встроенный HTML, Вы можете писать любой HTML-код, поддерживаемый браузером. Вы можете вставлять Ваши собственные JavaScript, Java апплеты, элементы ActiveX, файлы Shockwave и другие объекты.

По желанию, Вы можете установить признак Remove Default HTML generation для подавления генерации кода HTML для управляющего элемента.

Смотрите также: Встраиваемый HTML

События

В этой закладке Вы можете переопределить для управляющего элемента обработку событий, заданную по умолчанию. Эта закладка применима только для тех элементов, которые порождают события.

Каждый управляющий элемент имеет по умолчанию некоторое поведение. Оно определяет, как обрабатываются его события. Например, действием по умолчанию для командной кнопки (command button) является передача страницы серверному приложению и возврат свежей web-страницы.

Способность переопределять обработку событий, заданную по умолчанию для режима выполнения приложения в браузере, позволяет Вам оптимизировать приложение для web-среды и быть спокойным за то, что весь Ваш встроенный (embedded) код выполняется в те моменты, когда Вы этого ожидаете. Например, события, порождаемые элементом “поле ввода” (entry), обрабатываются по умолчанию браузером. Это означает, что любой код, определенный для события Event:Accepted для поля ввода не будет выполнен до тех пор, пока страница не будет отправлена по действию командной кнопки или другого управляющего элемента, отправляющего страницу. Используя переопределение для управляющего элемента, Вы можете назначить частичное обновление по событию Accepted для поля ввода. и встроенный код выполнится (как при работе под Windows).

По умолчанию у большей части управляющих элементов, позволяющих ввод данных, обработка событий производится в браузере. Поэтому большая часть встроенного кода не будет выполнена в ожидаемые моменты (например, код во вставке для события Event:Accepted 1 для управляющего элемента не будет

выполнен, пока страница не будет отправлена по кнопке ОК). Этот раздел позволит Вам переопределить обработку событий браузером.

Чтобы переопределить обработку события, выделите событие и нажмите кнопку Properties.

Override Default Action

Установите признак для переопределения действия по умолчанию для управляющего элемента. Установка этого признака открывает следующие пункты:

Action on Event

Выберите вариант действия для выполнения при появлении события:

Process on Browser

Разрешает произвести обработку события локально в браузере.

Partial page refresh

Определяет, что все управляющие элементы Java и поля ввода HTML должны получить и отобразить обновленные данные.

Complete page refresh

Заменить страницу целиком.

Классы

Закладка Classes предлагает Вам определить, какие классы(объекты) использовать для решения различных задач и исходные модули, содержащие определения классов. Такой подход дает Вам возможность использовать столько классов, сколько хотите из библиотеки IBC и сколько хотите - Ваших собственных.

Чтобы изменить класс для пункта или переопределить класс, выделите его в списке и нажмите кнопку Properties.

Override default class (Переопределение классов по умолчанию)

Чтобы переопределить класс IBC, установите этот флажок и определите имя класса (Class Name), файл заголовков(Header) и файл реализации(Implementation (.CLW)) в полях, описанных ниже.

Class Name (Имя класса)

Определите имя класса, который будет использоваться или имя класса по умолчанию, если Вы хотели бы переопределить класс по умолчанию.

Если Вы выберете другой класс из IBC Library, то Вы не должны определять Header или Implementation файлы.

Header file (файл заголовка)

Определите файл заголовка (файл, содержащий объявления класса) или выберите файл из FILEDIALOG нажатием кнопки (...).

Implementation file (файл реализации)

Определите файл реализации (файл, содержащий определения класса или исходный код) или выберите файл из FILEDIALOG нажатием кнопки (...).

Процедурные параметры компоненты Окно (Global Window Component Options)

Заголовок

Это область в верхней части “окна” на HTML-странице. Это часть, представляющая заголовок окна.

Override Global Settings

Установите этот флаг для переопределения установок для Caption в шаблоне Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Include caption

Установите этот флаг для отображения заголовка (Caption). Если флаг не установлен, то заголовок не используется.

Background color

Вы можете определить цвет для использования в области заголовка. Определите цвет(Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). По умолчанию установлен темно-синий цвет (Navy Blue) (эквивалент COLOR:Navy). Если цвет не определен и окно приложения имеет атрибут COLOR, то этот цвет отображается в браузере. Для определения данного свойства сгенерированный код использует метод WebCaption.SetBackground.

Background Image

Вы можете определить изображение для использования его в области заголовка в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию изображение не используется. Для определения данного свойства сгенерированный код использует метод WebCaption.SetBackground.

Совет: Изображение, используемое для фона, повторяется столько раз, сколько позволяет его размер внутри области HTML `<TABLE>`, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Alignment

Вы можете управлять выравниванием текста в области заголовка. Варианты для выбора - это выравнивание влево, вправо и по центру. По умолчанию - по центру.

Font Family name

Это позволяет Вам определять шрифт для отображения на экране. Запомните, что браузер может отображать только те шрифты, которые установлены на компьютере клиента.

Font size

При желании, Вы можете определить размер шрифта, отображаемого в области заголовка. По умолчанию не установлен (выбирается тот размер, который по умолчанию использует браузер).

Font color

Вы можете определить цвет шрифта для области заголовка. Определите цвет(Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...).

Меню

Это область меню в верхней или боковой части “окна” в HTML -странице.

Override Global Settings

Установите этот флаг для переопределения установок для Menu в Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Background color

Вы можете определить цвет для использования в области меню. Определите цвет(Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). Для определения данного свойства сгенерированный код использует метод `WebMenuBar.SetBackground`.

Background Image

Вы можете определить изображение для использования его в области меню в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию изображение не используется. Для определения данного свойства сгенерированный код использует метод

WebMenuBar.SetBackground.

Совет: Изображение, используемое для фона повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Alignment

Вы можете управлять расположением меню. Возможны следующие варианты: Above Toolbar (по умолчанию), Left of Window.

Панель инструментов

Это область панели инструментов, находящаяся в верхней части “окна” в HTML-странице (под областью заголовка).

Override Global Settings

Установите этот флаг для переопределения установок для Menu в Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Override Global Settings

Установите этот флаг для переопределения установок для Toolbar в Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Background color

Вы можете определить цвет для использования в области панели инструментов. Определите цвет (Color), задав цветовой числовой эквивалент (Color equate) или выбрав цвет из COLORDIALOG, нажав на кнопку (...). Для определения данного свойства сгенерированный код использует метод WebToolBar.SetBackground.

Background Image

Вы можете определить изображение для использования его в области панели инструментов в качестве фона. Определите имя файла с изображением или выберите его, используя диалог FILEDIALOG, нажав кнопку (...). По умолчанию изображение не используется. Для определения данного свойства сгенерированный код использует метод WebToolBar.SetBackground.

Совет: Изображение, используемое для фона, повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

Close button**Override Global Settings**

Установите этот флаг для переопределения установок для кнопки Close в Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Create extra close button

Определяет, когда создавать кнопку Close для окна.

Image for close

Определяет пиктограмму для отображения для кнопки Close. Определите имя файла с пиктограммой, или выберите файл в диалоге FILEDIALOG, нажав кнопку (...). По умолчанию определен файл EXITICO (маленькая голубая X) (поставляется с Clarion).

Клиентская область

Это область “окна” на странице HTML, представляющая клиентскую область приложения.

Override Global Settings

Установите этот флаг для переопределения установок для кнопки Close в Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Background Color

Вы можете задать цвет, который будет использоваться для Вашей клиентской области.

Определите цвет (Color), цветовой эквивалент (Color equate), или выберите цвет нажатием кнопки (...) в COLORDIALOG. Для определения данного свойства сгенерированный код использует метод WebClientArea.SetBackground.

Background image

Вы можете определить изображение, которое будет отображаться на экране как фон для клиентской области Вашего приложения. Задайте имя файла изображения или выберите файл из FILEDIALOG нажатием кнопки (...). По умолчанию изображения нет. Для определения этого свойства сгенерированный код использует метод WebClientArea.SetBackground.

Совет:

Изображение, используемое для фона, повторяется столько раз, сколько позволяет его размер внутри области HTML <TABLE>, представляющей область заголовка окна. Обеспечьте небольшой размер изображения для того, чтобы сохранить трафик.

MDI-параметры процедуры Frame

Меню приложения

Override Global Settings

Установите этот флаг для переопределения установок для панели управления MDI в Internet Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Include on Child Windows

Выберите значение параметра из выпадающего списка. Возможны следующие варианты:

Global Setting

Пункты меню будут отображаться в дочерних окнах так, как определено в глобальных параметрах (Global).

All Menu Items

Все пункты меню отобразятся в дочерних окнах.

No Menu Items

В дочерних окнах не будут отображаться пункты меню.

Selected Menu Items

Позволяет Вам выбрать определенные пункты меню из списка.

Ignore code in frame's ACCEPT loop

Установите этот флажок, чтобы код для пунктов меню, расположенный в цикле ACCEPT игнорировался.

Панель инструментов приложения

Этот раздел определяет стиль обработки управляющих элементов панели приложения. Он дает Вам возможность определить, какие из глобальных управляющих элементов панели отображаются в “дочерних” окнах.

Override Global Settings

Установите этот флаг для переопределения установок для панели управления MDI в Internet Application Global Extention template. Установка этого признака открывает для определения другие пункты.

Include on Child Windows

Выберите значение параметра из выпадающего списка. Возможны следующие варианты:

Global Setting

Пункты панели управления будут отображаться в дочерних окнах так,

как определено в глобальных параметрах (Global).

All Toolbar Items

Все пункты панели управления отобразятся в дочерних окнах.

Standard Toolbar Only

Только стандартные пункты панели управления отобразятся в дочерних окнах.

No Toolbar Items

В дочерних окнах не будут отображаться пункты панели управления.

Selected Toolbar Items

Позволяет Вам выбрать определенные пункты панели управления из списка.

Ignore code in frame's ACCEPT loop

Установите этот флажок, чтобы код для пунктов панели управления, расположенный в цикле ACCEPT, игнорировался.

Dynamic Html Code Template

Этот код шаблона позволяет Вам вставить динамический HTML-код в любую из точек встройки Internet. Этот шаблон годится только для тех точек вставки кода, которые могут “на ходу” выдавать данные на передаваемую страницу HTML.

Вы можете определить в поле ввода любое корректное выражение Clarion. Все переменные, использованные в приложении, будут иметь текущее значение на момент времени, когда пишется HTML код.

Примечание:

При создании выражения для записи кода HTML Вы должны помнить о корректной обработке специальных символов, таких как <, используя в этих случаях два символа подряд.

Этот шаблон использует метод Target.WriteLine для записи значения выражения на передающуюся страницу HTML.

Смотрите также раздел: Embedding HTML.

Static Html Code Template

Этот шаблон позволяет Вам вставить статический HTML-код в любую из точек вставки кода для Internet. Этот шаблон годится только для тех точек вставки кода, которые могут “на ходу” выдавать данные на передаваемую страницу HTML.

Вы можете задать любой значимый код HTML в поле ввода.

Этот шаблон использует метод Target.WriteLine для записи значения выражения в передающуюся страницу HTML.

Примечание: Если Вы используете Static HTML Code Template, обработка специальных символов поддерживается автоматически.

Смотрите также раздел: Embedding HTML.

GetCookie Code Template

Этот шаблон позволяет Вам восстановить cookie (переменную, используемую для координации в межзадачном пространстве) с клиентской машины.

Cookie Name	Обеспечьте имя для cookie. Это имя используется в шаблоне SetCookie Code для записи cookie. Если cookie не существует, устанавливаемой переменной присваивается нулевое значение.
Variable to Set	Выберите переменную из схемы связей файлов нажатием кнопки (...). Значение cookie присваивается переменной.

Смотрите также раздел: GetCookie Code Template, Cookies (Persistent Client Data).

SetCookie Code Template

Этот шаблон позволяет Вам установить cookie на клиентской машине для восстановления в дальнейшем.

Cookie Name	Обеспечьте имя для cookie. Это имя для использования в шаблоне GetCookie Code при восстановлении cookie. Если существует переменная cookie с тем же именем, то она переписывается..
New Value	Задайте значение или выберите переменную из схемы связи

файлов(file schematic) нажатием кнопки (...). Это значение присваивается cookie.

Смотрите также раздел: GetCookie Code Template, Cookies (Persistent Client Data).

Cookies (Persistent Client Data)

Cookies являются методом для web-серверов как для хранения, так и для восстановления информации на клиентской стороне соединения. Это позволяет серверу хранить данные на машине клиента и затем их восстанавливать.

Сервер может послать клиенту (броузеру) кусок данных, который клиент хранит локально.

Это известно как cookie (источник названия неизвестен). Cookies содержат диапазон URL, для которых это допустимо. Впоследствии, когда клиент возвращается по URL внутри этого диапазона, сервер может опросить cookie и использовать эти данные. Сервер не может восстанавливать информацию с других серверов (то есть, сервер не может опрашивать cookie, которые находятся вне его области диапазонов).

Этот механизм напоминает парадигму хранения и восстановления INI-файлов в Windows (GETINI и PUTINI) и обеспечивает метод для идентификации предпочтений пользователя и других данных. Например, приложение, которое требует от пользователя обеспечить его имя до входа может использовать cookie, чтобы избежать повторной регистрации после первого посещения.

Примечание:

Cookies являются машинно - зависимыми, так что клиенту, который осуществляет доступ к сайту более чем с одной машины, необходимо будет обеспечить cookie для каждой машины, поэтому cookie хранится на машине. К тому же, cookies зависят от броузера: клиент, пользующийся более чем одним броузером, должен будет устанавливать и получать cookies для каждого броузера.

Ваши web-пригодные приложения могут использовать cookies для хранения предпочтений пользователя, таких как город или штат по умолчанию для заведения новых записей. Эти установки могут быть восстановлены каждый раз, когда

пользователь выполняет приложение через web.
Смотрите также раздел: GetCookie Code Template, SetCookie Code Template

AddServerProperty Code Template

Этот шаблон позволяет Вам устанавливать значение заданного выходного элемента данных http в заголовке HTTP.

Property Name

Обеспечьте имя для cookie. Это имя, использующееся в SetCookie Code Template для записи cookie. Если cookie не существует, устанавливаемой переменной присваивается нулевое значение.

Property Value

Выберите переменную из схемы связи файлов(file schematic) нажатием кнопки (...). Значение cookie присваивается свойству.

Смотрите также раздел: GetServerProperty Code Template

GetServerProperty Code Template

Этот шаблон позволяет Вам получить значение заданного элемента данных http в заголовке HTTP.

Property Name

Обеспечьте имя для свойства HTTP. Это имя используется в методе SetServerProperty для установки значения поля HTTP. Если поле HTTP не существует, устанавливаемой переменной присваивается нулевое значение.

Variable to Set

Выберите переменную из схемы связей файлов(file schematic) нажатием кнопки (...). Значение свойства присваивается переменной.

Смотрите также раздел: SetServerProperty Code Template

Дизайн Web - приложений: соображения



Большинство правил разработки приложений применимы к разработке Web-приложения. В обоих случаях (web, Windows) одинаково важно обеспечить согласующийся, понятный интерфейс для обеих платформ.

Помните, что “платформа” Web - это не Windows. Ваш интерфейс должен быть интуитивно понятным для пользователей на всех платформах. Управляющие элементы в библиотеке поддержки Java интуитивно понятны, но для облегчения их использования Вы можете пожелать лаконичного объяснения того, как они работают в Вашем приложении.

Соображения пропускной способности

Web вводит одну дополнительную программистскую проблему - сохранение трафика. Важно сохранять Ваши окна простыми и использовать все методы, способные сократить сетевой трафик. Этот раздел дает некоторые пояснения, но отнюдь не все. Он предназначен для того, чтобы дать Вам пищу для размышлений во время разработки приложений.

Используйте частичное обновление, где возможно

Использование частичного обновления, где только возможно, - лучший путь оптимизировать Ваши Web-приложения.

Есть много случаев, когда подходит частичное обновление, но по умолчанию обновление - полное. Это необходимо по той причине, что шаблоны не могут предусмотреть все возможности. Например, список с вариантами сортировки, у которого нет управляющих элементов, размещенных на закладках, работает лучше, если Вы используете индивидуальные переопределения для управляющих элементов для задания частичного обновления при выборе закладки. При этом вместо полной замены страницы будут изменяться только данные в списке.

Чтобы переопределить поведение SHEET в предыдущем примере, произведите следующие действия:

1. Из окна Procedure Properties нажмите кнопку Internet Option.
2. Выберите закладку Controls.

3. Выделите управляющий элемент Sheet в списке Individual Control Options (мастер при генерации SHEET обычно называет его ?CurrentTab).
4. Нажмите кнопку Properties, затем выберите закладку Events.
5. Выделите событие Accepted, потом нажмите кнопку Properties.
6. Установите флажок Override default action (переопределить действие по умолчанию), затем выберите из выпадающего списка Partial page refresh (частичное обновление страницы).
7. Нажмите кнопки ОК во всех окнах для сохранения и выхода.

Будьте экономны с управляющими элементами

Разместите в окне необходимый минимум управляющих элементов. Это хорошая практика в разработке приложений Windows и еще более важно в реализации browser/server.

При использовании списков разместите в списке настолько мало управляющих элементов, насколько возможно для уникальной идентификации записи пользователем. Это сокращает количество данных, пересылаемых для наполнения списка. Если Вы хотите отобразить больше данных для каждой записи, Вы можете разместить “горячие поля” после списка, и они будут обновляться по мере скроллинга списка пользователем.

Используйте графику умеренно

Это общее правило для разработки web. Вам следовало бы ограничить количество графики на экране, чтобы ускорить загрузку страницы. В добавление, Вы должны бы сократить размер файла настолько, насколько возможно, для сохранения трафика. Многие графические утилиты обладают инструментом для приспособления графических файлов к использованию в web.

Соображения о косметическом дизайне

Использование групп

Когда Вы размещаете GROUP или WINDOW, предложения, относящиеся к описанию управляющего элемента, не обязательно оканчиваются внутри структуры GROUP. Это может вызвать ситуацию, когда HTML-представление окна отличается от оригинала. Убедитесь, что все управляющие элементы, которые Вы желаете разместить внутри GROUP действительно находятся внутри GROUP-структуры.

В первом примере из описанных ниже (BadWind) предложения, описывающие управляющие элементы, все находятся вне GROUP-структуры. Это окно прекрасно отображается в Windows, так как значения атрибута AT управляют положением и размером GROUP box. При запуске через Web, GROUP box является клеткой HTML<TABLE> и управляется ее содержимым.

```
BadWind WINDOW('Caption'), AT( , , 260 , 120 ), GRAY
GROUP('Customer Info'), AT( 5 , 9 , 205 , 102), USE(?Group1), BOXED
END
PROMPT('Customer:'), AT(11 , 28), USE(?CUST:Name:Prompt)
ENTRY(@s30), AT(61, 26), USE(CUST:Name), LEFT, REQ
PROMPT('Address:'), AT(15 , 47), USE(?CUST:Address:Prompt)
ENTRY(@s30), AT(61, 65), USE(CUST:Address), LEFT
PROMPT('City:'), AT(29 , 69), USE(?CUST:City:Prompt)
ENTRY(@s30), AT(61, 65), USE(CUST:City), INS
PROMPT('State:'), AT(25 , 88), USE(?CUST:State:Prompt)
ENTRY(@s2), AT(61, 86), USE(CUST:State), LEFT , UPR
END
```

Во втором примере (GoodWind) описания управляющих элементов находятся внутри GROUP-структуры (т. е. между предложениями GROUP и END) и будут отображаться так, как предполагалось при выполнении через Web.

```
BadWind WINDOW('Caption'), AT( , , 260 , 120 ), GRAY
GROUP('Customer Info'), AT( 5 , 9 , 205 , 102), USE(?Group1), BOXED
PROMPT('Customer:'), AT(11 , 28), USE(?CUST:Name:Prompt)
ENTRY(@s30), AT(61, 26), USE(CUST:Name), LEFT, REQ
PROMPT('Address:'), AT(15 , 47), USE(?CUST:Address:Prompt)
ENTRY(@s30), AT(61, 65), USE(CUST:Address), LEFT
PROMPT('City:'), AT(29 , 69), USE(?CUST:City:Prompt)
ENTRY(@s30), AT(61, 65), USE(CUST:City), INS
PROMPT('State:'), AT(25 , 88), USE(?CUST:State:Prompt)
ENTRY(@s2), AT(61, 86), USE(CUST:State), LEFT , UPR
END
END
```

Использование изображений

Управляющие элементы IMAGE с переменной в качестве атрибура предложения USE автоматически становятся управляющими элементами Java. Эти управляющие элементы автоматически обновляются, когда значение переменной изменяется (т. е. производится частичное обновление страницы). Если Вы хотите использовать эту возможность для статического IMAGE, который изменяется через присвоение

значения свойству (например, ?Image{PROP:TEXT}='New.gif'), используйте Individual Controls Overrides (индивидуальные переопределения для управляющего элемента) и установите признак для динамического обновления.

Графические файлы, используемые управляющими элементами IMAGE, раскрываются во временную директорию для сеанса соединения в случае, если они не найдены в директории /PUBLIC. Динамическая библиотека (runtime) будет раскрывать файлы различных типов, но большинство браузеров поддерживают только форматы GIF и JPG. Поэтому при работе в Web Вам следует ограничиться только двумя этими форматами при выборе типа графического файла для управляющего элемента IMAGE. Вы также должны устанавливать режим hide(спрятать) для IMAGE, формат которого не поддерживается браузером, используя Individual Controls overrides. Если IMAGE использует файл, который не включен в исполняемый модуль (not linked), Вы должны разместить этот файл в директорию приложения.

Вы должны обеспечить альтернативный текст для изображения (в Individual Control Overrides). Он добавляется к признаку HTML . Альтернативный текст отображается, пока графический файл передается в браузер (до отображения изображения), или вместо изображения, если пользователь отключил в настройках браузера отображение изображений.

Пиктограммы, используемые в управляющих элементах LIST или на BUTTONs, автоматически не раскрываются и должны быть размещены в директории /PUBLIC.

Если Вы ссылаетесь на изображение в HTML -коде, Вы должны указать расположение графического файла. Если Вы производите размещение для EXE-версии Application Broker, Вы можете поставить перед именем файла знак '/' и разместить изображение в директории /PUBLIC. Например, . Если Вы используете ISAPI DLL версию Application Broker, Вы должны использовать метод SELF.FILES.GETAlias для определения виртуального пути к файлу.

Например:

```
Target.WriteLine('<<IMG SRC=" ' & SELF.FILES.GETAlias('mygif.gif') & ' ">')
```

найдет файл mygif.gif в любой директории, предоставленной серверному приложению.

Разработка интерфейса пользователя

MDI window access

Windows-приложения часто используют Multiple Document Interface (MDI). Это позволяет открыть несколько экземпляров “дочерних” MDI-окон. Каждое из этих окон доступно и в него можно перейти, используя несколько навигационных методов (например, меню Window). Это очень удобно, но вносит некоторую путаницу при переносе приложения на web-платформу. Web-страница в браузере представляет собой один документ, однако основное серверное приложение может быть MDI-приложением и содержать множество окон. В приложении на сервере может быть открыто много окон, но браузер отображает только текущее. Вы должны помнить это при разработке приложения.

В Web-пригодном приложении Вы можете разрешить видимость в “дочерних” окнах всех пунктов меню и панели инструментов. Это может быть полезно, когда мы хотим позволить пользователю входить в разные части приложения без закрытия “дочернего” окна, пользуясь пунктами главного меню или панели инструментов. В этом сокрыта потенциальная ловушка, заключающаяся в возможности вызова пользователем нескольких экземпляров процедуры. Хотя только одно окно может быть видимо в определенный момент времени, открыто при этом может быть несколько. Если открыто два или более одинаковых окна, пользователю может показаться, что окно не закрывается при нажатии кнопки Close. По этой причине Вы должны запретить доступ к пунктам Global menu/Toolbar или определить ограничение на единственность экземпляра для каждой MDI-процедуры, используя код для потокового ограничения (Thread limiting code). Один из подходов к ограничению нитей продемонстрирован в одном из стандартных примеров Clarion - EventMgr.APP.

Неподдерживаемые стандартные Windows-диалоги

Есть определенные стандартные Windows-диалоги, которые непригодны для приложений, работающих в Web. Вызов любого из них приводит к появлению на экране сообщения Not Supported:

HALT
COLORDIALOG
FILEDIALOG
FONTDIALOG
PRINTERDIALOG
STOP

Если Вы вызываете любой из этих диалогов нажатием кнопки, то используйте Individual Control Options для кнопки для того, чтобы спрятать ее при работе через браузер (“Hide if launched from Browser”)(Internet Options Controls).

Если Вы вызываете функцию из исходного кода, заключите вызов функции в условную структуру.

Например:

```
IF NOT WebServer.Active      !Работаем через web?
  retval=COLORDIALOG()      !Если нет, вызываем COLORDIALOG
END
```

Использование параметров командной строки

Если Ваше приложение должно получать параметры из командной строки, Вы можете передать их, используя командную строку браузера, или гиперссылку.

В поле ввода, содержащем адрес расположения браузера (URL), определите URL, за ним следует имя исполняемого модуля, затем точка и нуль (.0) и завершает строку параметр с лидирующим вопросительным знаком.

Например,

```
HTTP://mydomain.com/myapp.exe.0?MyParametr
```

Для обработки параметра в приложении, Вы должны опросить свойство WebServer.CommandLine. Если Вы создаете гибридное приложение и хотите получать параметры командной строки в Windows и Web, используйте код, похожий на приведенный в примере:

```
IF WebServer.Active      ! Работаем через web?
  PRE:MyField = WebServer.CommandLine
ELSE
  PRE:MyField = COMMAND("")
END
```

Примечание: Если Вы передаете несколько параметров, то необходимо произвести разбор строки для доступа к индивидуальным параметрам.

Изменение класса для индивидуального параметра

В некоторых случаях у нас может появиться желание заменить класс, заданный по умолчанию для управляющего элемента. Например, управляющий элемент STRING, являющийся переменной, трансформируется в управляющий элемент Java

String, а Вам хотелось бы , чтобы он стал обычным текстом HTML. Вы можете изменять элемент за элементом, редактируя закладку Individual Control Overrides Classes Tab. В этом примере Вы не переопределяете класс, а просто определяете другой класс для использования при обработке данного управляющего элемента.

1. В окне Procedure Properties нажмите кнопку Internet Options.
2. Выберите закладку Controls.
3. Выделите управляющий элемент в списке Individual Control Overrides, затем нажмите кнопку Properties.
4. Установите флаг Override Default Class.
5. Выберите класс из выпадающего списка (в этом примере это класс WebHtmlStringClass). Вы не должны обеспечивать файл заголовка (Header File) и реализации (Implementation).

Вы можете использовать тот же подход для замены JavaImageControl на элемент HTML .

Вызовы API

Вызовы API в Windows связаны с машиной, на которой выполняется приложение. В действительности, Web-пригодные приложения выполняются на сервере, а представление посылается клиенту в форме страниц HTML. Таким образом, любые вызовы API в Вашем приложении выполняются на сервере.

Во многих случаях это не годится. Например, издающий звук файл на сервере - вообще не лучшая идея, и пользователь, работающий в приложении, не будет его слышать. В других случаях, Вам следует запретить вызов, когда приложение работает через web.

Если Вы производите вызов с помощью управляющего элемента BUTTON, используйте Individual Control Options, чтобы “Спрятать, если запущено из Броузера”-“Hide if launched from Browser”(управляющие элементы параметров Internet).

Если Вы делаете вызов в исходном тексте, поместите функцию вызова внутрь условной структуры. Например:

```
IF NOT WebServer.Active           ! Работаем в Web?
    SoundFile='fanfare.wav'
    sndPlaySound(SoundFile,1)
END
```

В других случаях будет вполне подходяще сделать вызов на сервере. Например, процедура, которая использует МАРІ, чтобы послать email с сервера, основана на событии. В других случаях Вам следует удостовериться, что вызов работает правильно на сервере. Стоит придерживаться того же правила при выполнении через web.

Похожим образом отчеты без разрешенного просмотра перед печатью (Print Preview) будут печататься на сервере. В некоторых случаях это допустимо, но важно при этом понимать, что происходит.

Соображения секретности

Есть несколько способов обеспечения секретности в Ваших web-приложениях.

- ◆ Обеспечение секретности внутри самого приложения.
- ◆ Ограничение доступа (парольная защита) к процедуре, когда она стартует через Web.
- ◆ Пересылка через секретное соединение.

Первый метод - обеспечение безопасности внутри самого приложения - не требует дополнительного рассмотрения в Вашем Web-приложении. Усиление секретности в версии Windows должно так же работать в Вашем Web-приложении.

Второй метод - ограниченный доступ при работе в Web - использует встроенное в Броузер распознавание.

Третий метод - пересылка через секретное соединение - преследует другую цель. Он не предназначен для ограничения доступа пользователю. Он предполагает запрещение перехвата данных во время передачи. Эта мера безопасности может быть использована отдельно или вместе с любой из двух других.

Использование паролей

Парольная защита шаблона расширения процедур для Internet использует встроенную в HTTP поддержку аутентификации. При вызове защищенной паролем процедуры отображается окно аутентификации броузера. Вам не нужно создавать окно для сбора информации для регистрации в системе. Парольная защита основана на области, имени пользователя и пароле. Областью является защищенная процедура.

Когда браузер обращается к защищенной паролем области, он получает обратно ответ, содержащий имя пользователя и пароль для доступа к области. По умолчанию имя области создается из заголовка окна и имени процедуры. Все это сохраняется в свойстве `WebWindow.AuthorizeArea`. Браузер предлагает пользователю ввести имя и пароль. Это отправляется затем для проверки в программу. Если программа воспринимает пароль (т.е. она возвращает `TRUE` из метода `WebWindow.ValidatePassword`), отображается новая страница, иначе браузер предлагает ввести все сначала. После трех попыток браузер отображает сообщение, информирующее пользователя, что доступ запрещен. Эта страница автоматически возвращает пользователя в последнее активное место в программе.

Примечание: Если страница уже посещалась во время текущего соединения, браузер подставляет пользовательское имя и пароль без необходимости ввода. Эта возможность встроена в большинство браузеров.

В процедурный шаблон (procedure template) встроены два уровня парольной защиты. Простейшим методом является выбор ограниченного доступа и определение единственного или переменного пароля. В этом случае происходит автоматическая проверка шаблоном и игнорирование имени пользователя. Если Вы используете переменную, шаблон сверяет введенный пароль с текущим значением переменной.

Более мощным методом является переопределение метода `WebWindow.ValidatePassword` вставкой кода в точку вставки кода `Internet-Password Validation Code Section`. Эта точка вставки находится внутри метода с двумя параметрами: `UserName` и `Password`, получаемыми из браузера. Вы должны возвращать `TRUE`, если пароль верен, и `FALSE` - если нет. Это позволяет Вам выбирать информацию из файла или использовать другой метод для проверки пароля.

Пример:

```
USE:UserID=UserName
Get(Userlist,USE:UserIDKEY)
IF ERRORCODE() THEN RETURN(False).
IF USE:UserPassword=Password
  RETURN(True)
Else
  RETURN(False)
END
```

При желании, Вы можете изменить сообщение, отображаемое в диалоге пароля браузера, присваиванием значения `WebWindow.AuthorizeArea` в точке встройки

Internet-After Initializing the window object.

Использование SSL(Secure Socket Layer)

Эта мера секретности требует, чтобы Вы выполняли ISAPI-версию брокера приложения под ISAPI-согласованным Web-сервером, и имели инсталлированный Digital Sertificate. Смотрите главу The Application Broker. SSL-поддержка шаблона Internet-расширения процедур использует ISAPI SSL-шифрование во время исполнения процедуры. Когда вызвана процедура с включенным SSL, брокер приложения переключается в режим SSL. Когда процедура заканчивается, восстанавливается нормальный доступ. Это дает возможность секретных транзакций на уровне процедуры. Помните, что шифрование заметно сказывается на производительности. Вам нужно обеспечивать секретность только для процедур, пересылающих важные данные. Разрешение шифровать только такие процедуры улучшает выполнение как на стороне клиента, так и на стороне сервера.

Для включения SSL для процедуры:

1. В окне Procedure Properties нажмите кнопку Internet Options.
2. Выберите закладку Advanced.
3. Установите флаг Transfer over a secure connection.

Использование встроенного HTML

Одной из наиболее мощных возможностей IBC- шаблонов (IBC Templates) является способность встраивания HTML-кода в HTML-страницы, являющиеся выходными для web-пригодного приложения, работающего через Application Broker. Когда Вы встраиваете HTML-код (используя специальные точки встраивания, добавляемые шаблонами), он вставляется в определенные места в HTML-файле, возвращаемом в браузер, который выполняет приложение.

Существуют два метода встраивания HTML:

1. Individual Control Overrides (индивидуальные переопределения для управляющего элемента) в Internet Procedure Extention Template(шаблон Интернет-расширение для процедур). Этот пункт предоставляет два текстовых поля для ввода HTML-кода.

2. Использование Dynamic HTML Code Template (шаблон для динамического HTML-кода), или Static HTML Code Template (шаблон для статического HTML-кода) в одной из точек вставки для Интернет. Эти шаблоны для вывода “на ходу” в

передаваемые HTML-файлы используют виртуальный метод Target.WriteLine.

Static HTML Code Template позволяет Вам встраивать код в том виде, как он написан. Dynamic HTML Code Template позволяет Вам соединять HTML-код с переменными из Вашего приложения.

При желании, Вы сами можете использовать метод Target.WriteLine в любом пригодном для вставки этого кода месте.

Эти пункты вставки кода в начале описания содержат слово INTERNET. Использование метода Target.WriteLine в одном из таких пунктов позволит Вам добавлять HTML-код в различные места HTML-документа в момент передачи пользователю. Этот код не влияет на работу приложения при выполнении его под Windows.

Например, если Вы хотите, чтобы в конце страницы, передаваемой Брокером (Application Broker) в процедуру приложения, появился блок текста, вставьте Static HTML Code Template в пункте Internet, before the closing </BODY> tag в генераторе приложения (Application Generator) и определите HTML-код. Этот HTML-код добавляется к результирующей HTML-странице, пересылаемой клиенту.

Вы можете использовать виртуальный метод Target.WriteLine в любых пунктах вставки кода, где применимы Dynamic HTML Code Template и Static HTML Code Template.

Пример:

Добавьте этот код в пункт вставки кода Internet, before the closing </BODY> tag:

```
Target.WriteLine('<<h>Copyright 1997 , TopSpeed&trade; Corporation, All Righths Reserved.<</ >')
```

Примечание: При написании исходного текста на Clarion для записи HTML-кода помните об особой обработке спецсимволов, таких как < , используя их повторение. При использовании Static HTML Code Template эта ситуация отрабатывается автоматически.

Одним из преимуществ использования Clarion-кода в этих точках вставки является возможность управлять HTML-кодом. Пример, приведенный ниже, показывает простой способ случайного отображения гиперссылки:

```
EXECUTE RANDOM(1,5)
Target.WriteLine('<<A HREF="http://www.topspeed.com">Visit TopSpeed<</A>')
Target.WriteLine('<<A HREF="http://www.clariononline.com">Visit ClarionOnLine<</A>')
Target.WriteLine('<<A HREF="http://www.isetips.com">Visit IceTips<</A>')
Target.WriteLine('<<A HREF="http://www.finansist.com">Visit Finansist<</A>')
Target.WriteLine('<<A HREF="http://www.topspeed.com/tsnews.htm">TopSpeed News<</A>')
```

Использование ссылок на файлы во встроенном HTML-коде

При использовании ссылок на файлы во встроенном HTML-коде помните, что каждое соединение имеет свою временную директорию. Поэтому, /PUBLIC никогда не является текущей директорией для передаваемой web-страницы. Это означает, что Вы должны указывать расположение файлов. Это можно сделать двумя способами.

Если Вы ссылаетесь из HTML на изображение, необходимо указать расположение файла с изображением. Если Вы разворачиваете приложение, используя EXE-версию Application Broker, то должны поставить перед именем файла символ / и разместить файл в директории /PUBLIC. Например: .

При использовании версии ISAPI DLL брокера приложений, Вы должны использовать метод SELF.FILES.GetAlias() для определения виртуального пути к файлу.

Например:

```
Target.WriteLine('<<IMG SRC="" & SELF.FILES.GetAlias('mygif.gif') & "">')
найдет файл mygif.gif в любой директории, открытой серверному приложению.
```

Примечание: Предпочтительным является использование метода **SELF.FILES.GetAlias()**, потому что он работает в обеих версиях **Application Broker (EXE и ISAPI DLL)**.

Для использования собственных файлов с классами Java Applet, используйте CODEBASE=признак, как показано ниже.

Если Вы размещаетесь под EXE-версией брокера приложения, Вы можете вместо лидирующего символа / и разместить файл .CLASS в директорию /PUBLIC. Если Вы используете версию ISAPI DLL брокера приложения, Вы должны опросить свойство WebServer.JavaClassPath для определения виртуального пути <CODEBASE>.

Пример вставки HTML:

```
<IMG SRC="/mypic.gif">
<applet codebase="/" code="Ticker Tape.class" width="500" height="32">
</applet>
Embedded Source Examples (in any Internet Embed Point)
Target.WriteLine('<<IMG SRC="' & SELF.FILES.GETAlias('mygif.gif')& ">')
Target.WriteLine('<<applet ')
Target.WriteLine('Codebase = "' & WebServer.JavaClassPath & ' "')
Target.WriteLine('code="TickerTape.class">')
Target.WriteLine('<</applet> ')
```

Примечание: В APPLET HTML tag атрибут CODEBASE должен предшествовать атрибуту кода. Это распечатано в ошибочном порядке в некоторых ссылках HTML. Код HTML с атрибутами, стоящими в неправильном порядке, может вызвать ошибку апплета (с сообщением "Not found").

Реализация помощи в Ваших Web-приложениях

Ссылки на HTML-странице производятся, основываясь на Help ID текущего окна. Это строится одним из двух способов: используя базовый документ с внутристраничными ссылками, или используя индивидуальные документы помощи. Это определено в Global Application Extension Template или в Procedure Extension template's Internet Options.

Использование базового документа с внутристраничными ссылками

Этот метод использует единичную Web-страницу с внутристраничными закладками или ссылками. Обращение к странице конструируется добавлением Help ID к имени базовой страницы с символом # между ними (то есть HELP.HTM#IDNAME). Нажатие на кнопку Help вызывает открытие страницы и пролистывание до подходящей ссылки. В примере ниже первое окно имеет Help ID ~FirstWindowID. Это значит, что кнопка Help вызовет HelpFile.HTM#FirstWindowID.

Пример:

```
<html>
<head>
<title> Example Help Document</title>
```

```
</head>
<body background="bgrnd.gif" bgcolor="#FFFFFF">
<h1 align="center">Program Help </h1>
Вводный текст.....
Вводный текст.....
Вводный текст.....
Вводный текст.....
<h2 align="center"><a name="FirstWindowID">Help For First Window</a></h2>
Объяснение как процедура работает
Объяснение как процедура работает
Объяснение как процедура работает
Объяснение как процедура работает
<h2 align="center"><a name="SecondWindowID">Help For Second Window</a></h2>
Объяснение как процедура работает
Объяснение как процедура работает
Объяснение как процедура работает
Объяснение как процедура работает
</body>
</html>
```

Использование индивидуальных документов помощи

Этот метод использует единственную web-страницу для каждого окна. Запрос к странице строится на добавлении к Help ID расширения .HTM. Нажатие на кнопку Help вызывает открытие страницы.

Оба метода открывают страницу в новом окне броузера под названием "_HELP". Если Вы открываете Ваше приложение внутри множества фреймов, один из которых называется "_HELP", страница помощи открывается в этом фрейме.

Web-пригодное приложение, выполняемое брокером приложения, создает HTML-файлы в директории /Public. Эти страницы посылаются броузеру, который запускает приложение, и обновляются и перепосылаются, когда клиент взаимодействует с приложением.

Управляющие элементы Windows и их HTML-эквиваленты

Web-пригодное приложение, выполняемое Application Broker, передает HTML броузеру, который запускает приложение и обновляет и перепосылает, когда клиент взаимодействует с Web-страницей, представляющей приложение.

Некоторые управляющие элементы легко транслируются в HTML, другие же создаются как JAVA-классы, используя Clarion Java Support Library. Некоторые оконные управляющие элементы еще полностью не воплощены в этой версии.

В списке, приведенном ниже, перечислены стандартные оконные управляющие элементы, поддерживаемые Clarion и их эквиваленты, создаваемые web-пригодным приложением Internet Connect.

SSTRING(переменная строка)

Отображает Java String Control (строка-управляющий элемент JAVA), обновляемый динамически.

STRING

По умолчанию, отображается как текст. Через индивидуальные настройки (Individual Control Overrides) можно определить, чтобы отображался как Java String Control (строка-управляющий элемент JAVA), обновляемый динамически. Если Вы меняете STRING через присвоение свойств, необходимо указать, что строка обновляется динамически.

IMAGE

Статическое изображение отображается как HTML-изображение , где в качестве источника определяется графический файл. Через индивидуальные настройки (Individual Control Overrides) можно определить, чтобы отображался как Java Image Control (изображение -управляющий элемент JAVA), обновляемый динамически. Для получения дополнительной информации смотрите Images (изображения).

REGION

Частичная поддержка. REGION, который включает элемент IMAGE и функциональность которого реализована в его EVENT:Accepted, создает HTML-изображение как таблицу изображения (image map) (USEMAP=) с функциональностью области, ассоциируемой с этой частью изображения.

LINE

Не поддерживается—используйте встроенный HTML для изображения <HR> или .

BOX

Не поддерживается—используйте встроенный HTML для изображения .

ELLIPSE

Не поддерживается—используйте встроенный HTML для изображения .

ENTRY

Создается как HTML-поле ввода `<INPUT TYPE=TEXT VALUE=value in field>`. Шаблоны ввода не поддерживаются.

BUTTON

Создается как `<INPUT TYPE=SUBMIT>`, если только не имеет `ICON`, затем создается кнопка `JAVA`, на которой отображается пиктограмма. Пиктограммы, отображаемые на кнопках, должны быть размещены в директории `/PUBLIC`.

PROMPT

Отображается как текст.

OPTION

Создается как HTML `<OPTION>`. Если `OPTION` имеет атрибут `BOXED`, он реализуется в HTML как `<TABLE>` с границей, определенной в параметрах `Global` или `Procedure` для `OPTIONS`.

CHECK

Создается как HTML checkbox `<INPUT TYPE=CHECKBOX VALUE=value in field>`.

GROUP

Если `GROUP` имеет атрибут `BOXED`, он реализуется в HTML как `<TABLE>` с границей, определенной в параметрах `Global` или `Procedure` для `GROUPS`.

Tree

Создает `Java Tree List Box`. Поддерживает все атрибуты, включая условные цвета и пиктограммы. Пиктограммы должны быть размещены в директории `/PUBLIC`.

LIST

Создает `Java List Box`, который поддерживает большинство атрибутов `LIST`, включая условные цвета и пиктограммы. Пиктограммы должны быть размещены в директории `/PUBLIC`. Когда `Java List Box` является активным элементом в броузере, поддерживаются навигационные нажатия (стрелка вверх, страница вверх и т.д.). Если у `LIST` есть локатор, `Java List Box` поддерживает его, когда он является активным. Обработка двойного нажатия мыши тоже поддерживается. “Взять и перетащить”, “редактирование по месту” и контекстное меню по нажатию правой кнопки мыши не поддерживаются.

COMBO

Создается как HTML-поле ввода `<INPUT TYPE=TEXT VALUE=value in field>`.

SPIN

Создается как HTML-поле ввода `<INPUT TYPE=TEXT VALUE=value in field>`.

TEXT

Создается поле HTML Text <TEXTAREA>.

CUSTOM(.VBX)

Не поддерживается.

MENU

Создает список гиперссылок, которые отображаются в верхней части HTML-страницы или в левой части окна, что определяется в Global Internet Options.

ITEM

Смотри меню.

RADIO

Создает кнопку HTML Radio.

APPLICATION

HTML <Table> внутри HTML-страницы.

WINDOW

HTML <Table> внутри HTML-страницы.

REPORT

Если доступен Print Preview, это создает серию HTML-страниц с кнопками навигации Java (Next page, Previous page и т.д.). Если Preview не включен, то отчет будет печататься на сервере.

HEADER, FOOTER, BREAK, FORM, DETAIL

Смотри REPORT.

OLE

Не поддерживается (кроме возможности вставки ActiveX в встроенный HTML).

FileDropCombo

Создается как HTML drop-down (структура <SELECT>) со значениями из файла, указанного в Options. Здесь не поддерживаются множественные колонки. При желании Вы можете создать список Java Non-drop, который поддерживает множественные колонки.

droplist

Создается как HTML drop-down (структура <SELECT>). Здесь не поддерживаются множественные колонки. При желании Вы можете создать список Java Non-drop, который поддерживает множественные колонки.

PROGRESS

Не поддерживается.

SHEET

Создается как управляющий элемент JAVA Tab.

TAB

Создается как управляющий элемент JAVA Tab.

PANEL

Не поддерживается. Вы можете использовать GROUP с подходящей

толщиной границы для обеспечения схожего внешнего вида.

TOOLBAR

Создается как ряд в HTML <TABLE>. Управляющие элементы на панели инструментов размещаются, как это определено в глобальных или процедурных Internet-параметрах.

Программирование приложений вручную

Шаблоны Internet Connect порождают код, необходимый для подготовки к работе в Web Clarion-приложений. Однако Вы не должны использовать шаблоны Internet Connect для подготовки к Web Ваших программ.

Это означает, что Вы можете использовать для подготовки к Web Ваших вручную написанных программ IBC Library. Эта глава представляет небольшую, написанную вручную программу “Hello Web”, использующую IBC Library. В этой главе обсуждаются требования проектной системы для IBC Library.

Наиболее легкий путь для изучения использования IBC Library в программах, написанных вручную, - это подготовка приложения к Web с использованием Internet Connect Templates с последующим изучением сгенерированного кода.

HelloWeb - пример программы

Следующая гибридная Web/Windows -программа отображает одно окно или Web-страницу с сообщением “Hello Web” и кнопкой “Goodbye Web” для завершения программы.

```

HelloWeb PROGRAM
LinkBaseClasses      EQUATE(1)      !Разрешить LINK в описаниях CLASS
                                   !так компоновщик (linker) может найти файлы
                                   !реализации (.clw)
BaseClassDllMode      EQUATE(0)      !Активизировать DLL в описаниях
                                   ! CLASS
                                   ! для требующегося 32-разр.
                                   ! разименования (dereference )
                                   ! Описать BrokerClass
                                   ! Описать WebWindowClass
                                   ! Описать IC стандартные EQUATES
INCLUDE('ICBROKER.INC')
INCLUDE('ICWINDOW.INC')
INCLUDE('ICSTD.EQU')
MAP
```



```

Hello                                ! Прототип процедуры Hello
WebControlFactory(SIGNED),*WebControlClass ! Прототип WebControlFactory
MODULE(' ')
    SetWebActiveFrame(<*WebFrameClass>)    ! Прототип SetWebActiveFrame
END
END
Broker            BrokerClass            ! Описать объект Broker
HtmlManager       HtmlClass              ! Описать объект HtmlManager
JavaEvents        JslEventsClass         ! Описать объект JavaEvents
WebServer         WebServerClass         ! Описать объект WebServer
WebFilesManager   WebFilesClass          ! Описать объект WebFilesManager
ShutDownManager   CLASS(ShutDownClass)
Close             PROCEDURE,VIRTUAL
END
ICServerWin       Window,AT(-1,-1,0,0)    ! Описать "невидимое" окно на
                                           ! сервере
END
CODE
SetWebActiveFrame()                    ! Сообщить объектам IBC
                                           ! (WebWindow), что нет активного
                                           ! фрейма приложения
WebFilesManager.Init(1, ' ')           !Инициализировать WebFilesManager
JavaEvents.Init                          !Инициализировать JavaEvents
Broker.Init('HelloWeb', WebFilesManager) !Инициализировать Broker
HtmlManager.Init(WebFilesManager)       !Инициализировать HtmlManager
WebServer.Init(Broker, ' ',600,' ',WebFilesManager) !Инициализировать WebServer
IF (WebServer.GetInternetEnabled())      ! Если запущено брокером приложения
    OPEN(ICServerWin)                   ! открыть "невидимое" окно на сервере
    ACCEPT
        IF (EVENT() = EVENT: OpenWindow)
            WebServer.Connect           ! Установить канал для брокера приложения
            Hello
            BREAK
        END
    END
ELSE                                     ! Если запущено не брокером приложения
    Hello                               ! Вызов Hello (режим Web)
END
WebServer.Kill                          ! Закрыть объект WebServer
HtmlManager.Kill                        ! Закрыть объект HtmlManager
Broker.Kill()                           ! Закрыть объект Broker
JavaEvents.Kill                         ! Закрыть объект JavaEvents
WebFilesManager.Kill                    ! Закрыть объект WebFilesManager
Hello    PROCEDURE

Window WINDOW,AT(, , 139,59),GRAY,DOUBLE ! описать окно
    STRING('Hello Web!'),AT(51,14),USE(?Hello) ! со строкой Hello Web

```

```

        BUTTON('Goodbye Web!'),AT(39,31),USE(?Bye) ! и кнопкой Goodbye Web
    END
WebWindow WebWindowClass                                ! Описать объект WebWindow

CODE
OPEN(window)                                             ! Открыть окно
WebWindow.Init(WebServer,HtmlManager)                  ! Инициализировать объект
                                                         ! сбором информации об
                                                         ! окне и его управляющих
                                                         ! элементах

ACCEPT
    IF WebWindow.TakeEvent/() THEN BREAK.               ! Обработка события Web:
                                                         ! обрабатывает все события,
                                                         ! необходимые для ответов на
                                                         ! запрос клиента, например,
                                                         ! создает новую страницу
                                                         ! HTML

    IF EVENT() = EVENT:Accepted                          ! Обычная обработка события Windows
        POST(Event:CloseWindow)                        ! Закрыть окно по кнопке ?Bye
    END
END
CLOSE(window)                                           ! Закрыть окно
WebWindow.Kill                                         ! Закрыть объект WebWindow
RETURN

WebControl Factory PROCEDURE(SIGNED Type) !Создать экземпляры объектов WebControl
NewControl      & WebControlClass                  !запрошенные объектом WebWindow
CODE
CASE (Type)
OF CREATE:ClientArea
    NewControl &= NEW WebClieAreaClass
OF CREATE:String
    NewControl &= NEW WebHtmlStringClass
OF CREATE:TextButton
    NewControl &= NEW WebHtmlButtonClass
END
IF (~NewControl &= NULL)
    NewControl.IsDynamic = TRUE
END
RETURN NewControl
ShutDownManager.Close  PROCEDURE
CODE
WebServer.Kill
HtmlManager.Kill
Broker.Kill()
JavaEvents.Kill
WebFilesManager.Kill

```

Обсуждение проекта, закодированного вручную

Библиотека IBC требует несколько компонент для успешной компиляции и связи объектных модулей. Укажите следующие компоненты в диалоге Project Editor. Для получения более полной информации смотрите The Project System в Руководстве пользователя.

ICSTD.CLW

ICSTD.CLW содержит множество процедур, разделяемых несколькими различными объектами IBC. Прототипы для этих процедур находятся в ICSTD.INC. Эти процедуры не являются классами или методами и, следовательно, не могут быть определены для загрузчика через атрибут LINK как методы IBC. Для того, чтобы указать на них загрузчику, необходимо добавить файл ICSTD.CLW в дерево проекта, в пункт с названием External source files. Файл ICSTD.CLW по умолчанию устанавливается в Clarion-директорию LIBSRC\.

Драйвер DOS-файлов

Объекты библиотеки IBC используют драйвер DOS-файлов для записи HTML-кода и данных JSL, запрашиваемых браузером клиента. Вы должны добавить драйвер DOS-файлов CLW в пункт дерева проекта с названием Database drive libraries для разрешения ссылок IBC на процедуры драйвера DOS-файлов.

Драйвер ASCII-файлов

Объекты библиотеки IBC используют драйвер ASCII-файлов для выполнения отчета. Вы должны добавить драйвер ASCII-файлов CLW в пункт дерева проекта с названием Database drive libraries для разрешения ссылок IBC на процедуры драйвера ASCII-файлов.

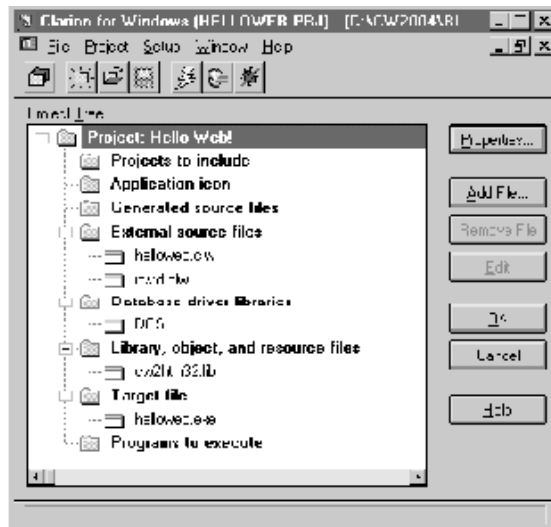
CW2HTM32.LIB

CW2HTM32.LIB содержит множество откомпилированных объектов, разделяемых несколькими различными объектами IBC. Прототипы для этих исполняемых объектов находятся в ICSTD.INC. Для того, чтобы указать на них загрузчику, необходимо добавить файл CW2HTM32.LIB в дерево проекта, в пункт с названием Library, object, and resource files.

Web-пригодные программы должны быть 32-х разрядными

По всем практическим соображениям, Web-пригодные программы должны быть 32-х разрядными. В Интернет-среде несколько клиентов могут одновременно

обращаться к приложению, следовательно, программа должна поддерживать существование нескольких экземпляров на Web-сервере. 32-х разрядные программы позволяют это (в противоположность 16-ти разрядным).



9

Краткое руководство по IBC Library

Для обеспечения работы гибридных Web/Windows приложений, шаблоны Internet Connect Templates основываются на библиотеке Internet Builder Class (IBC) Library. В этой главе кратко документированы методы и свойства, используемые шаблонами Internet Connect Templates, а так же другие методы и свойства из IBC Library, которые Вы можете использовать при создании гибридных Web/Windows приложений.

Для получения полной документации обратитесь к IBC Library Reference. Все методы и свойства IBC Library полностью документированы в IBC Library Reference. IBC Library Reference находится на инсталляционном диске Internet Connect в формате .PDF.

Классы и их шаблонно-сгенерированные объекты

Шаблоны Internet Connect создают экземпляры объектов из IBC Library. Имена объектов обычно похожи на соответствующие имена классов, но не совпадают полностью. В результате, сгенерированный код Вашего Web-приложения выглядит примерно так:

```
Broker.Init  
MainFrame.TakeEvent  
IC:CurFrame.CopyControlsToWindow  
WebWindow.OptionBorderWidth = 2  
IC:CurControl.Init  
IC:CurControl.DisabledAction = DISABLE:Show  
WebMenuBar.SetBackground(16711680, "")  
HtmlPreview.Init(WebServer, HtmlManager, PrintPreviewQueue)
```

Различные классы IBC и их экземпляры в шаблонах перечислены ниже таким образом, чтобы Вы легче могли определить объекты IBC в сгенерированном коде приложения. В разделе Quick Reference перечислены объекты для шаблонов вслед за наименованиями классов.

<u>Internet Builder Class</u>	<u>Template Generated Object</u>
BrokerClass	Broker
HtmlClass	HtmlManager
JslEventsClass	JavaEvents

TextOutputClass	Target
HttpClass	BrokerHttp
WebFilesClass	WebFilesManager, Broker.Files, HtmlManager.Files, Broker.Http.Files, JavaEvents.Files, WebServer.Files, WebWindow.Files и Target.Files
WebServerClass	WebServer
WebClientManagerClass	Broker .CurClient
WebFrameClass	MainFrame и IC:CurFrame
WebWindowClass	WebWindow
WebControlClass	IC:CurControl
WebCaptionClass	WebCaption
WebClientAreaClass	WebClientArea
WebMenuBarClass	WebMenuBar
WebToolBarClass	WebToolBar
WebReportClass	HtmlPreview

Краткое руководство

BrokerClass(Broker)

Init (инициализирует объект BrokerClass)
Kill (закрывает объект BrokerClass)
ServerName (имя сервера)

WebClientManagerClass(Broker.CurClient)

IP (IP-адрес клиента)

HtmlClass (HTMLManager)

Init (инициализирует объект HTMLClass)
Kill (закрывает объект HTMLClass)

JslEventsClass (JavaEvents)

Init (инициализирует объект JslEventsClass)
Kill (закрывает объект JslEventsClass)

TextWriterClass (HtmlManager или Target)

WriteLine (пишет одну строчку текста)

HttpClass (Broker.Http)

GetCookie (получить cookie от клиента)

SetCookie (установить cookie для клиента)

SetProcName (установить имя защищенной области)

SetProgName (установить имя сервера)

WebFilesClass (WebFileManager или Files)

GetAlias (возвращает HTML alias для файла)

Init (инициализирует объект WebServerClass)

Kill (закрывает объект WebFilesClass)

SelectTarget (устанавливает общедоступный или секретный канал)

WebServerClass (WebServer)

Active(Web или Windows - режим)

CommandLine(параметры командной строки)

Connect(открывает канал соединения с Брокером)

Init(инициирует объект WebServerClass)

PageToReturn(возвращает URL)

ProgramName(путь на сервере)

Quit(завершение серверной программы)

SetSendWholePage(вызов полного обновления страницы)

SetNewPageDisable(подавление выходящих web-страниц)

TimeOut(период отсутствия активности, после которого происходит завершение задачи)

WebFrameClass(MainFrame или IC:CurFrame)

CopyControlsToWindow(помещение глобальных управляющих элементов в локальное окно)

FrameWindow(ссылка на приложение)

TakeEvent(обработать события Броузера и цикла ACCEPT)

WebWindowBaseClass(WebWindow)

AllowJava(генерировать или подавлять генерацию JavaScript)

BorderWidth(толщина обрамления Web-страницы)

CloseImage(изображение на кнопке Close)

CreateCaption(включает заголовок на web-страницу)
CreateClose(включает кнопку Close на Web-страницу)
DisabledAction(HTML по умолчанию для отключенных управляющих элементов)
FormatBorderWidth(толщина обрамления клетки HTML таблицы)
GroupBorderWidth(толщина обрамления группы элементов(group box))
Menubar(расположение меню)
OptionBorderWidth(толщина обрамления выбора из вариантов(option box))
SheetBorderWidth(толщина обрамления листа(sheet))

WebWindowClass(WebWindow)

AuthorizeArea(имя защищенной паролем Web-страницы)
HelpDocument(документ помощи (HTML))
HelpEnabled(флаг разрешения помощи(HTML))
HelpRelative(удаленный или локальный документ помощи)
IsSecure(общедоступный или секретный канал)
AddControl(добавление управляющего элемента)
CreateHtmlPage(генерация HTML для окна)
GetControlInfo(возвращает ссылку на управляющий элемент)
GetToolBarMode(возвращает Toolbar entity)
Init(инициализирует объект WebWindowClass)
Kill(закрывает объект WebWindowClass)
MenubarType(расположение меню)
SetBackground(установить фон Web-страницы)
SetPassword(требовать пароль)
SetSplash(сделать это окно splash-окном)
SetTimer(установить для Web-страницы таймер и действие)
SupressControl(не включать управляющий элемент на Web-страницу)
TakeEvent(обработать события Броузера и цикла АСCEPT)
ValidatePassword(проверить пароль)

WebControlClass(IC:CurControl)

DisableAction(HTML для отключенных управляющих элементов)
CreateHtml(записать HTML для управляющего элемента и его атрибутов)
Freq(номер управляющего элемента)
ParentFreq(номер родительского управляющего элемента)
Init(инициирует объект WebWindowClass)
SetBorderWidth(установить толщину обрамления)

WebJavaStringClass(IC:CurControl)

SetAutoSpokLink(установить “живые” гипертекстовые ссылки)

WebHtmlImageClass(IC:CurControl)

SetDescription(установить альтернативный текст для Web-изображения)

WebJavaListClass(IC:CurControl)

ResetFromQueue(записать изменения в очередь для LIST на сервере)

SetAutoSpotLink(установить “живые” гипертекстовые ссылки)

SetEventAcion(установить очередь - источник данных)

WebCaptionClass(WebCaption)

Alignment(выравнивание текста)

SetBackground(установить фон заголовка Web-страницы)

SetFont(установить шрифт заголовка Web-страницы)

WebClientAreaClass(WebClientArea)

SetBackground(установить фон области клиента Web-страницы)

WebMenubarClass(WebMenubar)

SetBackground(установить фон области меню Web-страницы)

WebToolbarClass(WebToolbar)

SetBackground(установить фон области панели инструментов Web-страницы)

WebReportClass(HtmlPreview)

Init(инициировать объект WebReportClass)

Kill(закрыть объект WebReportClass)

Preview(генерировать HTML для представления отчета)

Глоссарий

Все определения являются общими понятиями, кроме специально помеченных. Понятия, помеченные (Clarion) являются специфичными для языка Clarion и для Среды разработки Clarion.

applet

Небольшое, целевое приложение. Необязательно представляют собой отдельные исполняемые модули. Маленькие программы, написанные на Java, обычно называют апплетами. В HTML, <APPLET> означает Java applet.

Application Broker

(Clarion) Application Broker необходим для исполнения гибридных Web/Windows приложений Clarion. Application Broker запускает гибридное Web/Windows приложение на Internet - сервере и обновляет Clarion Java Support Library (JSL) в Броузере. Затем Application Broker организует трафик сообщений в сессии удаленного соединения, направляя события, порождаемые Java Support Library в гибридное Web/Windows приложение, а HTML - скрипты, порождаемые приложением - в Бrowsers.

Broker

(Clarion) Смотри Application Broker.

Client

(Clarion) Бrowsers клиента, запускающий посредством Application Broker гибридное Web/Windows приложение.

cookie

Информация, хранимая на машине клиента по запросу сервера.

default button

Командная кнопка, которая по умолчанию активизируется, когда пользователь нажимает кнопку ввода.

disabled

Окно, меню или управляющий элемент, видимый, но не способный получить управление.

encryption

Представление данных в зашифрованном виде, чтобы неавторизованный пользователь не смог получить доступ к данным в понятном представлении.

font

Основная часть названия файлов, связанных одним типом начертания. Например, "Times New Roman" - это имя шрифта, а "Times New Roman plain", "Times New Roman Italic", "Times New Roman Bold" - это стили, содержащиеся в отдельных файлах.

font style

Форматирование символа, применяемое к начертанию шрифта, например bold, italic или bold italic.

Gif image

Формат Graphics Interchange File (GIF). Формат изображений, популяризуемый CompuServe. В основном, признанный за способность эффективного сжатия изображения, содержащего 256 или менее цветов. Внимание: везде, где Вы используете слово “GIF”, Вы должны добавлять упоминание торговой марки : “GIF (Graphics Interchange File) является торговой маркой CompuServe Information Services”.

global toolbar

Горизонтально или вертикально упорядоченная группа командных кнопок и/или других управляющих элементов, в основном предполагающая доступ к ней в продолжении всего времени выполнения программы.

hide

Предотвращает вывод на экран управляющего элемента или окна. Управляющий элемент существует, но не виден на экране конечному пользователю.

HTML

Hyper-Text Markup Language-язык, используемый Интернет-броузерами для форматирования и отображения Web-страниц.

Hybrid Web/Windows Application

Гибридные Web/Windows приложения выглядят как стандартные Windows приложения при работе в Windows, но работают как Интернет сервера, когда запускаются из Application Broker. Гибридные Web/Windows приложения могут управляться из любого Java-совместимого броузера (например, Microsoft Internet Explorer или Netscape Navigator).

icon(пиктограмма)

Графическое представление определенного физического объекта, присутствующего в системе (например, принтера). Пиктограмма также может обозначать действие, понятие программы в случаях, когда появляется на командных кнопках. Обычно, файл с пиктограммой имеет расширение .ICO. Одним из основных свойств пиктограммы является встроенная поддержка прозрачности. Это позволяет отображать маленькие изображения без уничтожения фона.

include file

Внешний файл с исходным кодом, считываемый и препроцессируемый во время компиляции. В Clarion, Equates и других файлах в директории LIBSRC по умолчанию являются include

Internet Developer's Kit

файлами.

(Clarion) Internet Developer's Kit является дополнительным продуктом, который может использоваться с Clarion Standard, Professional или Enterprise Edition для разработки новых гибридных Web/Windows приложений, или для приспособления к Web уже существующих приложений. Версия Application Broker, поддерживающая одно соединение, включена в Internet Developer's Kit.

Java Support Library

(Clarion) Java Support Library (JSL) - это небольшое множество Java-классов (меньше 200k), которое обеспечивает в Интернет-браузере широкое множество Windows-подобных управляющих элементов. JSL порождает события из Интернет-браузера и обрабатывает сообщения от Интернет-сервера.

JPG image

Графический формат, поддерживающий 24-х разрядные цвета. Обычно обеспечивает большое сжатие при потере разрешающей способности.

JSL data

Протокол и данные, которые гибридное Web/Windows приложение посылает в Интернет-браузер для обработки библиотекой Java Support Library (JSL). Гибридное Web/Windows приложение посылает JSL-данные в Интернет-браузер для достижения очень быстрого частичного обновления Web-страницы.

Remote Computing Session

(Clarion) Clarion Application Broker организует события, порождаемые Java Support Library (JSL) и HTML-страницы, порождаемые гибридными Web/Windows приложениями, в удаленную сессию, поддерживая состояние диалога между браузером и сервером.

Reusable Client

(Clarion) Java Support Library (JSL) - это небольшое множество Java-классов (менее 200 K), генерирующее события из Интернет-браузера и обрабатывающее сообщения от Интернет-сервера. Этот "тонкий" клиент может использоваться повторно любым гибридным Web/Windows приложением, минимизируя, таким образом, время соединения и требования к ресурсам локального браузера (RAM и дисковое пространство).

Server

(Clarion) Гибридное Web/Windows приложение, запускаемое Application Broker-ом по запросу Интернет-броузера.

Session Router

(Clarion) Session Router обеспечивает выполнение удаленной сессии в Интернет на нескольких Application Broker-ах в случаях, когда это требуется (например, в случаях высокой популярности приложения или других случаях необходимости развертывания дополнительных серверов Интернет). Вы можете отдельно приобрести Session Router.

timer

Ресурс Windows, который может автоматически посылать сообщение приложению через предопределенные интервалы.

Ultra-thin Reusable Client

(Clarion) Java Support Library (JSL) - это небольшое множество Java-классов(менее 200 К), генерирующее события из Интернет-броузера и обрабатывающее сообщения от Интернет-сервера. Этот “тонкий” клиент может использоваться повторно любым гибридным Web/Windows приложением, минимизируя, таким образом, время соединения и требования к ресурсам локального броузера (RAM и дисковое пространство).

