

1

Введение

Добро пожаловать!

Благодарим Вас за приобретение Clarion!

Вы держите в руках самый мощный из имеющихся в продаже инструмент по разработке прикладных программ для Windows! Теперь Вы можете создавать сложные прикладные программы для Windows быстрее, чем могли даже представить.

Уровни пользователя / разработчика

Clarion управляет несколькими уровнями и каждый уровень поддерживает, будь то уровень пользователя или разработчика.

Если вам необходимо несложное приложение, обслуживающее базу, то используя Clarion, вы можете выполнить эту работу за считанные минуты. Ключом тут будет словарь базы данных. Если Генератор Приложений знает, какие файлы и таблицы вам нужны в вашем приложении и как они связаны, он без проблем построит приложение. И потому ваша задача - представить один или несколько файлов и определить тип отношений (если файлов два или более). Небольшое руководство **Quick Start**, представленное этой книгой, покажет вам, насколько прост этот процесс.

Мастер приложения может создать полнофункциональное приложение, используя по умолчанию метафору Clarion приложения. Мы называем эту метафору browse-form и она “проистекает” прямо из структуры базы данных. Приложение создается примерно так: (1) пользователь манипулирует базой, - всей или только частью, одним файлом или множеством связанных файлов, - прокручивая их список, в котором каждый файл представлен одной записью. Окно, в котором это происходит, называется окном просмотра (browse). (2) Пользователь выбирает определенную запись в списке для некоторого действия, например, редактирования данных. Обычно это происходит в отдельном окне, в котором поля базы появляются в особых полях для редактирования. Эти поля называются формой для обновления. Кроме того, форма может использоваться для ввода новых данных. Заметим также, что в процессе ввода пользователь может посмотреть значения в связанных таблицах;

просмотр открывается в отдельном окне. За один шаг пользователь может выбрать пункт, закрыть новое окно, разместить значение в поле редактирования формы. Все это называется “lookup”.

Это самое общее описание. Помимо прочего, каждое окно просмотра, управляемое из панели, открывает отдельные пути со своими собственными буферами записей, обеспечивающими безопасность управления данными. Вызвав закладку, вы можете выбрать последовательность ключей. Двойной щелчок на записи в списке открывает форму обновления с автоматическим согласованным контролем (есть поддержка для многопользовательского режима) и необязательной поддержкой ссылочной целостности (поддерживается связями вашей базы данных).

Это может сделать каждый! Для того, чтобы начать - выберите файл данных из списка.

Среда визуальной разработки

Управление в окне Clarion дает вам гораздо больше, чем все другие инструменты Быстрой Разработки Приложений, которые обычно предлагают вам управление пользовательским интерфейсом, но при этом вы должны написать код, обеспечивающий согласованную функциональность. В Clarion вы добавляете шаблон, который содержит в себе управление и все, что необходимо для данных и исполнимого кода.

Это значит, что вы не должны писать код, - один щелчок дает законченное решение: управление интерфейсом пользователя и код, позволяющий выполнить эту работу. Более того, каждый шаблон имеет собственный пользовательский интерфейс. Если вы посмотрите свойства шаблона, вы увидите закладку “Actions” (действие). Выбирая поле падающего списка или заполняя поле в списке редактирования, вы настраиваете поведение шаблона так, что он точно отображает ваши требования. Вы будете задавать значение поля “actions” (действие) для шаблонов во многих местах в процессе изучения этой книги.

Руководство “**Генератор Приложений**” в книге “Изучая Clarion” познакомит вас со всеми RAD - инструментами Clarion.

Когда вы используете интерфейс шаблонов для уточнения их свойств, Генератор Приложений пишет код (на языке Clarion), который обеспечивает

функционирование шаблона по вашему требованию. Этот код объектно-ориентированный, создан с применением Библиотеки ABC (*Application Builder Class*), и потому очень компактный и эффективный. Используя шаблоны, вы можете сделать очень много без ручного кодирования.

Возможности языка Clarion поначалу кажутся невероятными. Руководство по языку насчитывает около 1000 страниц. Одним из важных преимуществ владения комплексом (система шаблонов плюс возможность писать код руками) в том, что вы можете писать на языке так неспешно или так быстро, так много или так мало, как вы пожелаете. Если же вам не нравится решение частных проблем с помощью шаблонов, вы можете использовать их только поначалу. Далее вы можете использовать интерфейс шаблонов, чтобы модифицировать чуть больше, по вашему вкусу. В конце концов, когда вы узнаете язык Clarion, вы напишите решение самостоятельно и будете иметь полную свободу делать это своим способом. (Отметьте, что можно просто купить шаблоны третьих фирм и решать проблему таким образом).

Законченный язык программирования

На самом общем уровне вы можете, используя Clarion, написать приложение руками. Clarion имеет высокий уровень абстракции, компактную грамматику, поэтому вы можете легко обходиться с записью или множеством записей. Вы не должны знать, как Clarion создает приложение... но если вы будете знать, как он работает, это поможет вам создавать лучшие приложения. Руководство **Генератор Приложений** в книге “Изучая Clarion” позволит вам написать меньше в сравнении с вашим собственным объектно-ориентированным кодом; руководство “**Язык Clarion**” (также из книги “Изучая Clarion”) познакомит вас с языком Clarion для полностью ручного кодирования.

Профессиональные разработчики по достоинству оценят язык Clarion. Он был создан как основа для бизнес-программирования. Несмотря на свою компактность, TopSpeed'овский компилятор элегантно преобразует весь ваш код - создали его вы или Генератор Приложений - в высокооптимизированный машинный код.

Не имеет значения, какой уровень в работе подразумеваете вы, Вы будете работать более продуктивно, если изучите все руководства до конца.

Что Вы найдете в этой книге

Эта книга является первым знакомством со средой Clarion. Ниже перечислены разделы этой книги и дано краткое их описание.

Введение	Глава 1: эта глава содержит первые подробные инструкции по установке Clarion на Ваш жесткий диск.
Обучение работе с Quick Start	Глава 2: Несколько быстрых шагов с Мастером позволят вам создать законченное приложение за пять минут. Потратьте меньше часа, и эта глава покажет вам, как использовать Мастер Приложений и Мастер Процедур, чтобы создать простое приложение. Распространите их функциональность на изящную программу, управляющую двумя связанными файлами. Вы завершите все это, не написав ни единой строки кода.
Средства разработки	Глава 3: Обзор концепции Clarion разработки приложений. Здесь вы найдете описания всех инструментов среды разработки.

Где еще вы можете найти информацию

Первое место, куда следует обратиться за дополнительной информацией - раздел "How do I..." в файле помощи. Тут освещаются многие важные вопросы, которые обычно возникают у новичков. Чтобы добраться до этого раздела, нажмите кнопку "How do I...?" в оглавлении файла помощи (Help).

Система помощи в Clarion обширна. Гипертекстовый файл помощи появляется, когда вы нажимаете кнопку F1 или выбираете одну из команд Help меню. Help оформлен диалогом, обеспечивающим точную контекстную помощь, когда это необходимо.

Документация Clarion представлена несколькими книгами:

Getting Started

это книга, которую Вы сейчас читаете.

Изучая Clarion

Книга содержит два пошаговых руководства. Первое руководство - Генератор Приложений — дает широкое представление обо всех инструментах Среды Разработки. Второе - представляет программирование на языке Clarion.

Справочник приложений

Справочник инструментов Clarion вы будете использовать на всем протяжении разработки приложения (для поставки Standard Edition он предлагается вам на компакт-диске в формате .PDF). Вы найдете здесь описания шаблонов, поставляемых с продуктом, полную документацию по Библиотеке ABC - Библиотеке Классов, используемой в шаблонно-ориентированном кодировании и бездну информации по драйверам.

Описание языка (в стандартном комплекте поставки содержится на компакт диске в формате .PDF) является законченным руководством по языку Clarion. Оно содержит описание и примеры всех утверждений и функций. Содержание этой книги разбито по темам. Полный текст *Описания языка* содержится также во встроенной системе справки. При работе в текстовом редакторе для получения подсказки достаточно разместить курсор на любом предложении или функции языка Clarion, а затем нажать клавишу F1.

Руководство пользователя Содержание этой книги разбито по темам, обусловленным основными компонентами среды (в стандартном комплекте поставки содержится на компакт диске в формате .PDF).

Руководство программиста является сборником серьезных статей на различные темы программирования на языке Clarion и настройки среды разработки (содержится только в профессиональной или полной поставке). Эта книга содержит также полный обзор языка шаблонов Clarion, включая описание всех конструкций и функций с примерами, помогающими создавать свои собственные шаблоны.

Руководство пользователя по Report Writer содержит описание Report Writer for Windows™ - самостоятельного инструмента для создания отчетов конечного пользователя (содержится только в поставке Professional или поставке Enterprise Editions - на компакт диске в формате .PDF; доступно как отдельный продукт).

Инструментарий для корпоративной разработки Документация по всему инструментарию для корпоративных разработок и клиент / серверных разработок доступна вам в поставке Clarion Enterprise Edition (включается только в Enterprise

Edition; некоторые компоненты, входящие в эту поставку, доступны как отдельные продукты).

Руководство пользователя Deployment Kit

Документы по *Deployment Kit* от TopSpeed - руководства конечного пользователя по установкам многих инструментов (включается только в Enterprise Edition; доступно как отдельный продукт).

Business Math Library

Документация по бизнес / статистическим функциям (включается только в Enterprise Edition; доступна как отдельный продукт).

Master Index

Полный список по всей печатной документации и документации в .PDF - формате.

Все книги на CD в .PDF формате доступны программе Adobe Acrobat Reader, которую вы можете инсталлировать с компакт-диска Clarion. Файлы формата .PDF полнотекстовые, с доступным поиском, могут быть распечатаны на вашем принтере, если вы того захотите.

Это важно: если содержание некоторого раздела электронной справочной системы противоречит содержанию соответствующего раздела печатной документации, руководствуйтесь электронным справочником.

Фирма TopSpeed Corp. прилагает все возможные усилия для обеспечения корректности печатной документации. Однако, для изготовления печатной копии требуется время, приводящее к задержке в выходе документации. И если мы можем обновить справочный файл, поставляемый непосредственно с Вашей версией продукта, печатная документация будет обновлена позже.

Соглашения, принятые в этом руководстве

Соглашения о шрифтах:

Курсив	Этим шрифтом выделяется последовательность символов, которую требуется ввести с клавиатуры, например, <i>введите это</i> .
МАЛЕНЬКИЕ ПРОПИСНЫЕ БУКВЫ	Так обозначаются клавиши клавиатуры, например, ENTER или ESCAPE.
Утолщенный	Этим шрифтом выделяются команды и параметры раскрывающихся меню или текст в окнах диалога.
ГОТИЧЕСКИЕ БУКВЫ	Этим шрифтом используется для выделения примеров, для схем и распечаток текстов программ, а также для примеров использования утверждений.

Соглашения об обозначениях клавиш клавиатуры

F1	Означает нажатие на клавишу. Нажмите и отпустите клавишу F1.
ALT+X	Означает комбинацию нажатий на клавиши. Нажмите клавишу ALT, и, удерживая ее, нажмите клавишу X. Затем отпустите обе клавиши.

Информация о продукте

Регистрация продукта

Перед началом использования Clariion заполните и отошлите регистрационную карточку, входящую в комплект поставки. Наличие регистрационной карты Reply Card дает Вам право на получение важных дополнительных услуг. После регистрации Вы сможете воспользоваться службой технической поддержки TopSpeed, Вы будете автоматически получать аннотации новых продуктов и уведомления об изменениях и исправлениях.

Техническая поддержка

Вам предоставляется неограниченная бесплатная техническая поддержка в рамках информационной службы CompuServe и сети Internet. Сотрудники TopSpeed, входящие в группу поддержки (известную как *Team TopSpeed*), будут своевременно отвечать на Ваши вопросы. Здесь Вы сможете также получить ответы и советы от других пользователей Clarion. Мы очень рекомендуем всем нашим покупателям воспользоваться этим видом сервиса.

- При работе с CompuServe введите GO TOPSPEED для доступа к TopSpeed'овскому форуму поддержки.
- При работе с Internet обратитесь к конференции *comp.lang.clarion* с помощью сервера USENET.
- В Internet посетите страничку TopSpeed'овского сайта <http://www.topspeed.com>, чтобы узнать о перспективах Clarion.

В корпорации TopSpeed возможна также и платная техническая поддержка. Чтобы воспользоваться ею, Вам достаточно позвонить по телефону (900) 884 - 0444. Мы предоставляем несколько программ платной поддержки. Дополнительную информацию по этому вопросу Вы можете получить, позвонив в службу работы с покупателями корпорации TopSpeed по телефонам (800) 354-5444 или (954) 785-4555.

Система распространения документов по факсу

Фирма TopSpeed Corp. позволяет обладателям факсимильных аппаратов получать пользующиеся наибольшим спросом технические и торговые документы. Среди этих документов описания продуктов, техническая документация, перепечатки статей, прайс-листы и информация о самых последних изменениях в продуктах TopSpeed.

Чтобы получить документ, позвоните по телефону (954) 785 - 4555, нажмите 53 и следуйте инструкциям системы. Система имеет дружественные интерактивные меню. Если Вы позвонили в первый раз, Вы можете заказать список доступных документов, чтобы иметь возможность выбрать нужный. Вы можете просто ввести цифровой код документа и документ немедленно будет направлен Вам. Позвонить в службу рассылки факсов можно как с факсимильного аппарата, так и с любого телефона с тоновым набором.

Программа начальной установки

Программа начальной установки Setup разархивирует и копирует файлы Clarion на жесткий диск.

- При установке Clarion возможен выбор устанавливаемых компонент (например, файлов примеров).
- Для всех версий Windows в установке запрашивается разрешение на включение в предложение PATH файла AUTOEXEC.BAT каталога, содержащего директорию Clarion.
- При работе в Windows 3.x: запрашивается установка пиктограммы Диспетчера программ (Program Manager) для Clarion - среды разработки, Отладчика, файлов помощи.
- При работе в Windows 95 и Windows NT запрашивается разрешение на включение всех иконок файлов Clarion среды разработки в программы Start Menu.

Системные требования

Вы можете сформировать Clarion среду разработки на любой системе, удовлетворяющей минимальным системным требованиям Microsoft Windows 3.x, Windows 95, или Windows NT. Рекомендуемый минимум приводится ниже:

- Windows 3.x, 8 МБ RAM.
- Windows 95, 16 МБ RAM.
- Windows NT, 32 МБ RAM.
- В зависимости от того, какие опции установки вы предпочтете, вам потребуется от 12 до 35 МБ свободного пространства на жестком диске.

Ваши Clarion-приложения будут удобны в использовании на компьютерах, удовлетворяющих минимальным требованиям операционной системы.

Программа начальной установки

Чтобы запустить программу начальной установки Clarion при работе в Windows 3.x:

1. Вставьте в привод CD-ROM компакт-диск с установочной программой Clarion.
2. Из Program Manager (Диспетчера программ), File Manager (Диспетчера файлов) или другой программы оболочки, способной запустить программу, выберите команду File Run (Файл Выполнить).
3. В окне диалога Run (Выполнить) введите D:\SETUP (где D: - буква диска, присвоенная приводу CD-ROM) и нажмите кнопку ОК.

Программа начальной установки Setup обеспечивает появление вступительного экрана и другой текстовой информации.

Чтобы запустить программу начальной установки Clarion при работе в Windows 95:

1 Вставьте в привод CD-ROM компакт-диск с установочной программой Clarion.

2 Из меню Start выберите Settings > Control Panel (Настройки - Панель управления).

3 Выберите Add/Remove Programs (Установка/удаление программ), а затем нажмите кнопку Install (Установить).

В этом случае процесс установки пройдет под руководством Мастера Windows 95.

Параметры начальной установки

После запуска программы Setup на экране появится окно, отображающее опции установки.

1. Выберите требуемые параметры, предлагаемые экранами мастера каталог назначения), затем нажмите кнопку NEXT.

Основные компоненты среды разработчика (IDE) будут установлены в подкаталог BIN ниже каталога назначения, заданного Вами в окне диалога.

Программа начальной установки Clarion устанавливает все файлы в каталог назначения и в его подкаталоги. Ни один файл не устанавливается ни в какой другой каталог.

Во время установки на экране будут отображаться шкалы, показывающие количество скопированных файлов.

2. Ответьте Yes или No на запрос об изменении строки PATH в файле AUTOEXEC.BAT.

Для работы среды разработчика (IDE) Clarion необходимо, чтобы в строке PATH файла AUTOEXEC.BAT был указан подкаталог BIN. Если Вы ответили NO, Вам придется вручную отредактировать файл AUTOEXEC.BAT.

Впоследствии Clarion еще только один раз изменит Ваши системные файлы. При первом запуске он добавит свой раздел в файл WIN.INI (один из файлов настроек Windows).

3. Ответьте Yes или No на запрос о помещении в пиктограмм группы Clarion в меню Start.

4. Ответьте Yes или No на запрос об отображении файла README.

Если Вы не хотите сейчас читать этот файл, Вы сможете позже найти его пиктограмму в группе, отображаемой в Program Manager (Диспетчера программ) (или меню Start), созданной программой начальной установки. Мы рекомендуем Вам прочитать этот файл сразу после того, как будут скопированы все файлы.

5. По окончании начальной установки нажмите ОК.

Запуск Clarion

Для запуска Clarion найдите его пиктограмму в программной группе Clarion, укажите на нее курсором и дважды щелкните правую кнопку мыши. Если пиктограмма Clarion находится в меню Start, достаточно щелкнуть один раз.

На экране появится среда разработчика (IDE) Clarion, готовая к работе.

- В третьей главе Вы найдете оперативный путеводитель по частям IDE, диаграмму пиктограмм строки управления.
- А сейчас перейдите ко второй главе этого руководства, для того, чтобы менее чем за час создать две эффектные программы от словаря данных до окон просмотра и окон форм.

2

Обучение работе с Quick Start

В Clarion Вы можете создать словарь данных и рабочую программу, не введя ни единого символа исходного текста. Все, что вам необходимо сделать - это определить поля в ваших файлах данных, и Quick Start создаст законченную прикладную программу под Windows. Если же Вы умеете быстро вводить символы с клавиатуры, Вы сможете это сделать всего за 5 минут! Созданная прикладная программа содержит как процедуру формы для обновления файлов, так и множество окон просмотра и отчетов по ключевым полям в файле данных.

Достаточно определить поля одного файла. Для каждого поля Вам необходимо только задать имя, формат отображения и ключевую информацию. Эти начальные значения образуют словарь данных. А Quick Start, взяв за основу этот словарь, создаст прикладную программу!

В этой главе:

- С помощью Quick Start Вы создадите файл CUSTOMER, содержащий информацию о покупателях, и программу, поддерживающую этот файл. Затем откомпилируете и выполните программу.
- Воспользуетесь Quick Start, как одним из самых быстрых способов Clarion, для добавления файла к словарю данных.
- Поработаете с редактором словаря Dictionary Editor, соотнося два файла.
- Воспользуетесь генератором программ Application Generator для увеличения функциональности прикладной программы с помощью одного из Мастеров процедур Clarion.
- Наконец, с помощью Мастера программ (Application Wizard) Вы создадите вторую полностью реляционную программу, основанную на том же словаре данных.

Все эти операции займут у Вас около тридцати минут и не потребуют “программирования вручную”. По окончании главы Вы будете владельцем двух законченных прикладных программ, работающих с базой данных, состоящей из двух связанных файлов.

Приступим!

Мастер Quick Start

Создание словаря и приложения

Стартовая точка:

Среда Clarion разработки должна быть открыта.

Для начала Вам необходимо создать подкаталог C:\CLARION\TUTORIAL, в котором и будет размещаться созданная Вами программа. В этом случае подразумевается, что Clarion установлен в каталог C:\CLARION. Если же Вы задали другой каталог для размещения Clarion, соответствующим образом модифицируйте далее, выполняя наши инструкции.

Создайте рабочую директорию

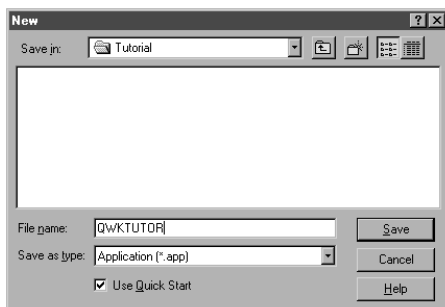
1. Задайте переключение для соответствующего инструмента вашей системы (Менеджер файлов, Проводник и т.д.) и создайте поддиректорию с именем *TUTORIAL* - в поддиректории *Clarion4*.

2. Вернемся к Clarion.

Создание вашего первого Clarion приложения

1. В окне диалога Pick (Выбрать) нажмите кнопку New... (Создать...).

На экране появится окно диалога New (Создать). Это стандартное окно диалога Open File, позволяющее изменить текущий каталог и имя файла.



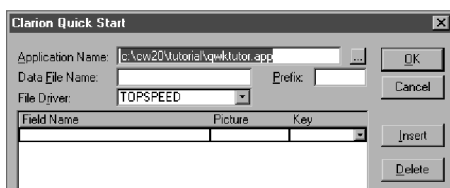
2. Выберите Application (*.app) в выпадающем списке для уточнения типа сохраняемого файла.

3. В списке каталогов Directories выберите \CLARION4\TUTORIAL.

4. В поле File Name (Имя файла) введите QWKTUTOR.

Мастер Quick Start использует заданное Вами имя QWKTUTOR как для прикладной программы, так и для словаря данных. Она автоматически создает QWKTUTOR.DCT (файл словаря данных) и QWKTUTOR.APP (файл прикладной программы). Длинные имена вы можете использовать, но не в этом курсе (для того, чтобы пользователи Windows 3.1 тоже насладились успехом).

5. Установите флажок Use Quick Start, (Использовать Мастер Quick Start), а затем нажмите кнопку Save (Сохранить). На экране появится окно диалога Wizard Quick Start.



Определение словаря данных

1. В поле Data File Name (Имя файла данных) введите Customer и нажмите клавишу TAB.

Это имя будет присвоено файлу словаря данных. Обратите внимание, что три первых символа данного имени (CUS) автоматически помещаются в поле Prefix (Префикс). Префикс, базирующийся на имени файла, однозначно определяет имена полей при работе с несколькими файлами, в которых могут встретиться поля с одинаковыми названиями.

2. Нажмите клавишу TAB для выбора CUS в качестве префикса.

Теперь Вы перешли к полю File Driver (Драйвер файла). В этом поле по умолчанию задан драйвер TOPSPEED (один из двух собственных форматов Clarion).

3. Нажмите клавишу TAB для набора драйвера TOPSPEED.

Теперь Вы перешли в окно списка определений полей.

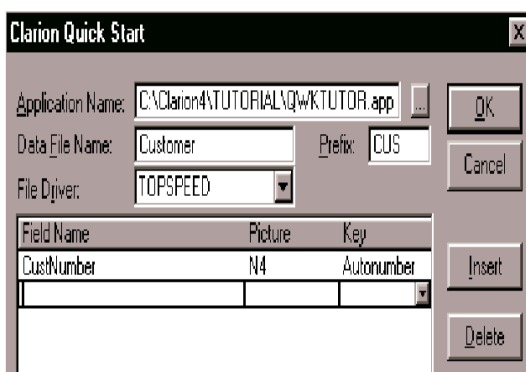
Внимание: Под Windows 3.1 следует загрузить программу SHARE.EXE или VSHARE.386.

4. В первой строке столбца Field (Поле) введите CustNumber и нажмите клавишу TAB.

Таким образом, Вы создали поле с именем CustNumber. Quick Start также использует это имя по умолчанию в качестве названия поля и заголовка столбца. Название поля отображается при размещении поля в окне. Заголовок столбца используется в отчетах.

5. В столбце Picture (Вид) введите N4 и нажмите клавишу TAB.

В этом столбце задается вид по умолчанию для элементов управления окна и отчета, что неявно определяют тип данных для Мастера Quick Start.



6. В столбце Key (Ключ) нажмите кнопку раскрытия списка. На экране появится список возможных значений. Выделите цветом *Autonumber*, а затем нажмите клавишу TAB.

Это значит, Вы задали создание уникального ключа в качестве компонента данного поля. Если задано значение *Autonumber*, Ваша прикладная программа будет присваивать каждому покупателю (customer) свой собственный, отличающийся от других, номер, который будет автоматически возрастать при добавлении новой записи. Quick Start создает процедуры просмотра (browse) и отчеты, основанные на каждом заданном ключе, который прокручивают записи, отсортированные по полю CustNumber.

Курсор переместился на строку ниже, разрешая тем самым определить следующее поле.

7. В столбце Field (Поле) введите Company и нажмите клавишу TAB.

Таким образом, Вы создали поле с именем Company, в котором будет храниться название предприятия покупателя.

8. В столбце Picture (Вид) введите S20 и нажмите клавишу TAB.

В этом столбце задается внешний вид по умолчанию для элементов управления окна и отчета. В данном случае Вы задали строковое (STRING) поле из 20 символов.

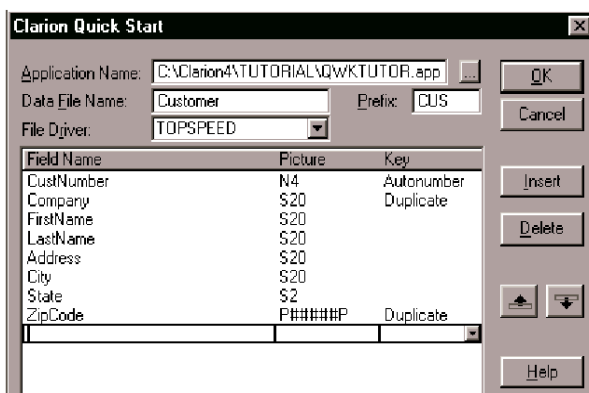
9. В столбце Key (Ключ) нажмите кнопку раскрытия списка. На экране появится список возможных значений. Выделите цветом Duplicate, а затем нажмите клавишу TAB.

Таким образом, Вы задали создание ключа, позволяющего дублировать вводимые данные. Этот ключ позволяет ввести двух покупателей из одного и того же предприятия (с одинаковым значением в поле Company).

10. Завершите определение файла, определив оставшиеся поля в соответствии с указанной ниже таблицей:

<u>FIELD (ПОЛЕ)</u>	<u>PICTURE (ВИД)</u>	<u>KEY (КЛЮЧ)</u>
FirstName (Имя)	S20	(no key)
LastName (Фамилия)	S20	(no key)
Address (Адрес)	S20	(no key)
City (Город)	S20	(no key)
State (Страна)	S2	(no key)
ZipCode (Почтовый индекс)	P#####P	Duplicate

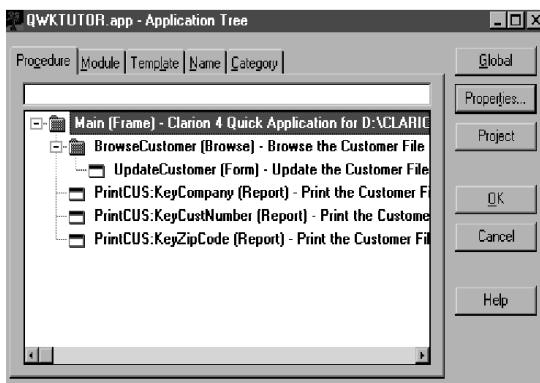
Обратите внимание на задание вида поля ZipCode (Почтовый индекс) - (P#####P). Данный шаблон определяет пятизначное цифровое поле.



Окончание работы с Мастером Quick Start и создание прикладной программы.

1. После того как Вы определите все поля, нажмите кнопку ОК.

Quick Start создаст Вашу программу, а на экране появится окно диалога Application Tree (Структура прикладной программы).



Обратите внимание на представленную в этом окне структуру. Вверху структурного дерева находится процедура MAIN (Frame), являющаяся каркасом программы, поддерживающим работу с несколькими документами (Multiple Document Interface (MDI)). Эта процедура содержит меню, из которого вызываются все остальные процедуры

Ниже процедуры MAIN располагается процедура просмотра BrowseCustomer (browse), к которой прикреплена процедур формы UpdateCustomer (form). Процедура просмотра отображает данные в порядке, указанном тремя ключами, заданными при определении полей. Также Вы видите три отчета, основанные на этих ключах.

Создание этой прикладной программы завершилось, не потребовав то Вас больше никакой информации.



2. Выберите команду Project Run (Проект Выполнить) или нажмите на панели управления кнопку RUN.

Генератор программ создал исходный текст Вашей прикладной программы, откомпилировал, скомпоновал и выполнил.

Примите наши поздравления! Ваша программа работает.

Исследование приложения

Добавьте несколько записей. Выберите Browse > Browse файла Customer.

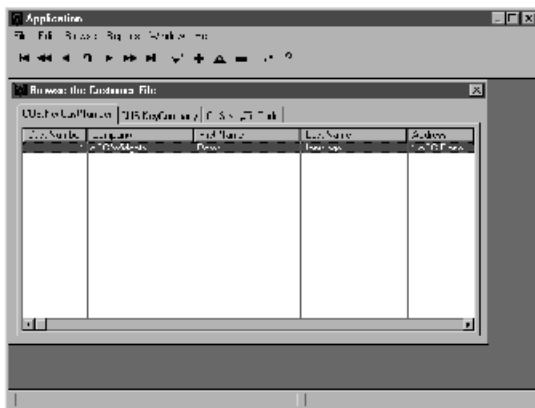
1. Нажмите кнопку Insert (Вставить).



3. После того, как вы ввели данные первого покупателя, нажмите кнопку ОК, чтобы закрыть окно формы и сохранить запись в файле данных.

Это наглядно показывает Clariion-метафору работы “по умолчанию” Browse-Form. Список Browse отображает записи файла данных, процедура Form используется для обновления (добавления, изменения, удаления) отдельных записей файла. Поддерживается метафора *Scrolling Form*, когда записи данных редактируются непосредственно в списке (мы вернемся к этому позже).

Добавим некоторое количество записей (можете также использовать щелчок в списке просмотра и выбрать Insert из падающего меню, чтобы вставить записи). Заметьте, что при добавлении новой записи в файл значение поля CustNumber автоматически увеличивается на единицу относительно максимального номера, добавленного ранее. Обратите внимание на различный порядок сортировки в процедуре просмотра, затем переопределите размер отображаемых колонок и используйте горизонтальную линейку прокрутки для того, чтобы исследовать данные в поле, которые первоначально вы видеть не могли. Вы можете даже напечатать отчет, если желаете.



4. Выберите File>Exit, чтобы закрыть ваше приложение. Вы сможете также попробовать запустить его из Диспетчера файлов File Manager или Explorer. Оно действительно работает в Windows, не требуя среды Clarion.

Сохраните Вашу программу и выйдете из Генератора программ (Application Generator). Тогда мы сможем изменить (модифицировать) словарь данных, который только что создали. Далее мы добавим второй связанный файл в словарь данных.

5. Для закрытия Генератора программ Application Generator выберите команду File > Close (Файл Закрыть) или нажмите кнопку ОК.

Мастер Quick Load

Вы можете быстро добавить файл в словарь данных, воспользовавшись для этого режимом Quick Load редактора словаря Dictionary Editor. Работа в этом режиме осуществляется так же, как и в Мастере Quick Start - определите поля (имена, вид, ключи), и определение полей автоматически добавляется в словарь. Этой опцией можно воспользоваться в любой момент при добавлении в словарь нового файла.

Добавление файла

В этом разделе мы добавим новый файл для хранения телефонных номеров покупателей. В этом файле каждому покупателю может соответствовать несколько телефонных номеров (соотношение один-ко-многим).

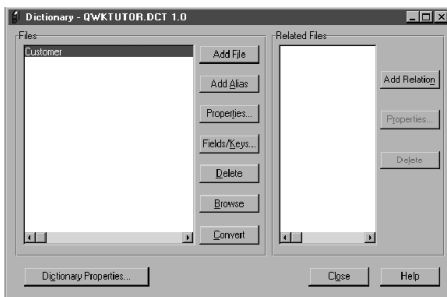
Использование Quick Load в Редакторе Словаря

1. Выберите команду File > Open (Файл Открыть) или нажмите на панели управления кнопку Open (Открыть).

Появится диалог Open.

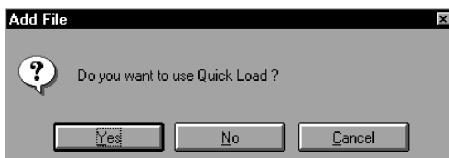
2. Выберите Dictionary (Словарь) из меню Files(Файл) выпадающего списка, выберите файл QWKTUTOR.DCT, а затем нажмите кнопку Open (Открыть).

На экране появится окно диалога Dictionary Editor (Редактор словаря).



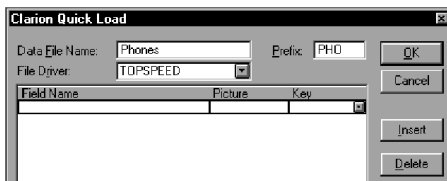
3. Нажмите кнопку Add File (Добавить файл).

На экране появится окно диалога, содержащее вопрос “Do you want use Quick Load?” (“Будете использовать Quick Load?”).



4. Нажмите кнопку Yes (Да).

На экране появится окно диалога Quick Load. Обратите внимание, что это окно очень похоже на окно диалога Мастер Quick Start.



5. В поле Data File Name (Имя файла данных) введите Phones и нажмите клавишу TAB.

6. Нажмите клавишу TAB для набора РНО в качестве расширения (Prefix). Теперь Вы перешли к полю File Driver (Драйвер файла).

7. Нажмите клавишу TAB для выбора драйвера TOPSPEED. Теперь Вы перешли в окно списка определений полей.

8. В первой строке столбца Field (Поле) введите CustNumber и нажмите клавишу TAB.

Таким образом, Вы создали поле с именем CustNumber. Это поле связано с полем CustNumber файла Customer. Использование одинаковых имен полей упрощает связывание двух полей в отношении. В созданном коде Clarion добавляет расширение (префикс) или имя файла (отделяется двоеточием) к меткам поля, создавая уникальные имена для полей, содержащихся в разных файлах, но имеющих одинаковые имена.

9. В столбце Picture (1) введите N4 и нажмите клавишу TAB.

10. В столбце Key (Ключ) нажмите КНОПКУ РАСКРЫТИЯ СПИСКА. На экране появится список возможных значений. Выделите цветом Duplicate, а затем нажмите клавишу TAB.

Таким образом, Вы задали создание ключа, позволяющего дублировать вводимые данные, разрешая тем самым вводить более одной записи на пользователя. Это позволяет нам задать для двух файлов отношение One-to-Many (Один-ко-многим).

Курсор переместился на строку ниже, разрешая тем самым определить следующее поле.

11. В столбце Field (Поле) введите Area (Область) и нажмите клавишу TAB.

12. В столбце Picture (Вид) введите P(###)P, затем дважды нажмите клавишу TAB (не определяйте ключ) для перехода к следующему полю.

Определенный Вами формат задает поле в виде трех цифр, заключенных в круглые скобки. (Стандартная нотация для телефонных кодов в Северной Америке).

13. Завершите создание файла, определив оставшиеся поля в соответствии с указанной ниже таблицей:

<u>FIELD (ПОЛЕ)</u>	<u>PICTURE (ВИД)</u>	<u>KEY (КЛЮЧ)</u>
Phone (Телефон)	P#####P	(no key)
Description (Описание)	S20	(no key)

14. По окончании определения всех полей нажмите кнопку ОК. Затем Quick Load спросит вас, завершили ли вы определение полей.

15. Нажмите кнопку ОК.

Quick Load создаст определение файла и добавит его в словарь.

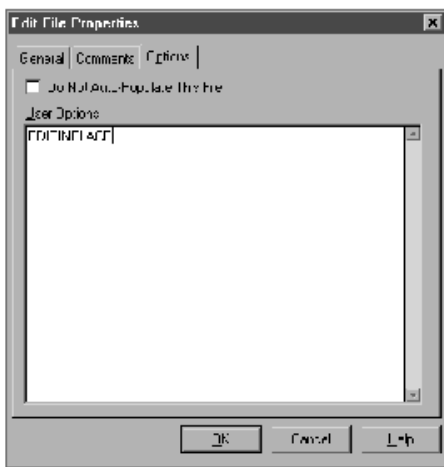
Установка(задание) нового файла для использования метафоры Scrolling Form

Так как в файле только четыре поля, он является хорошим примером задания файла, для которого редактирование данных справа в прокручиваемом списке - наиболее подходящий вариант.

1. Подсветите поле *Phones* в списке файла и нажмите кнопку Properties.

2. Выберите кнопку Options.

3. В пользовательских управляющих опциях введите *EDITINPLACE*. Это позволит шаблонам Clariion генерировать код, позволяющий вам непосредственно редактировать данные файла, как только они появляются в поле списка просмотра.



4. Нажмите кнопку ОК.

Добавление отношений

Очевидно, мы хотим, чтобы файл Phones (Телефоны) содержал номера телефонов покупателей. Это означает, что между двумя этими файлами должна существовать связь. В нашем случае, когда один покупатель может иметь несколько

телефонных номеров, однозначно подходит отношение один-ко-многим. Для задания этого отношения мы должны связать (link) их вместе в словаре данных. Таким образом, генератору программ (Application Generator) предоставляется информация, необходимая для доступа к связанным записям.

Задание отношения двух файлов

1. В окне списка Files (Файлы) выделите цветом строку Customer, а затем нажмите кнопку Add Relation (Добавить связь).

На экране появится окно диалога New Relationship Properties (Новые характеристики отношения). В этом окне Вы можете определить отношение.



2. Убедитесь, что в поле Type (Тип) содержится значение 1:MANY.

3. В поле Primary Key (Основной ключ) нажмите кнопку раскрытия списка. На экране появится список возможных значений. Выделите цветом строку KEY_CustNumber, а затем нажмите клавишу TAB.

Это ключ, определенный в файле Customer (сторона “Один” в отношении один-ко-многим), который мы и используем для связи двух файлов.

4. В поле Related File (Связываемый файл) нажмите кнопку раскрытия списка. На экране появится список возможных значений. Выделите цветом строку Phones, а затем нажмите клавишу TAB.

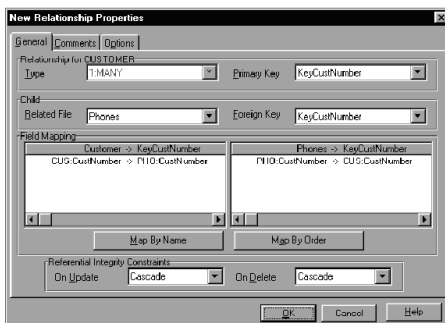
5. В поле Foreign Key (Внешний ключ) нажмите кнопку раскрытия списка. На экране появится список возможных значений. Выделите цветом строку KEY_CustNumber, а затем нажмите клавишу TAB.

Это ключ, определенный в файле Phones (сторона “Многие” в отношении один-ко-многим), который мы и используем для связи двух файлов.

Теперь связывающие поля должны быть назначены (map), чтобы генератор программ Application Generator точно знал, какие именно поля этих двух файлов являются родственными (связующими). Поскольку мы используем одинаковые имена полей, это будет просто.

6. Нажмите кнопку Map By Name (Назначить по имени).

Связи отобразятся в двух окнах списка Field Mapping (Назначение поля).



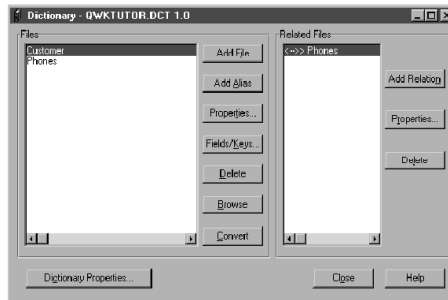
Задание мер по поддержанию целостности (Referential Integrity Constraints)

1. В панели Referential Integrity Constraints (Требования по поддержанию целостности) в выпадающем списке On_Update (На обновление) выберите строку Cascade (1).

2. В панели Referential Integrity Constraints (Требования по поддержанию целостности) в выпадающем списке On_Delete (На удаление) выберите строку Restrict (Ограничить).

Созданный исходный текст программы будет автоматически поддерживать внутреннюю целостность реляционной базы данных. Руководство “Изучение Clarion” (*Learning Clarion*) объясняет это свойство. Краткое обсуждение теории реляционных баз данных Вы найдете в Руководстве пользователя в разделе “Использование редактора словаря Dictionary Editor”.

3. Чтобы закрыть окно диалога New Relationship Properties (Новые характеристики отношения), нажмите кнопку ОК. Словарь при этом будет выглядеть так:



4. Для сохранения словаря выберите команду File Save (Файл Сохранить) или нажмите на панели управления кнопку Save (Сохранить).

5. Для закрытия словаря выберите команду File Close (Файл Закрыть) или в окне диалога Dictionary (Словарь) нажмите кнопку Close (Закрыть).

Мастера процедур

В дополнение к Мастеру Quick Start, Clarion содержит также Мастера процедур. Эти мастера создают процедуры просмотра (Browse), формы (Form) и отчета (Report) также быстро, как Мастер Quick Start создает целый словарь данных и прикладную программу. Мастера процедуры просмотра (Browse), формы (Form) и отчета (Report) доступны в любой момент при создании процедуры в уже существующей прикладной программе.

Сейчас мы воспользуемся Мастером просмотра (Browse) для создания процедуры просмотра файла Phones. При этом также создастся процедура обновления, если только для файла в этот момент не производится редактирование по месту.

Загрузите генератор программ Application Generator

1. Выберите команду File Pick (Файл Выбрать) или нажмите на панели управления кнопку Pick (Выбрать).

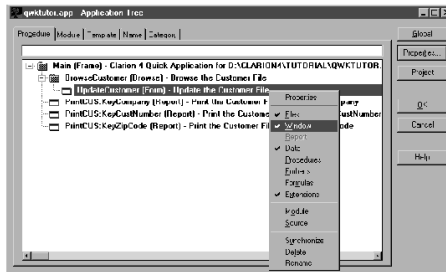
2. Выберите закладку Application (Прикладная программа), выделите цветом строку C:\CLARION4\TUTORIAL\QWKTUTOR.APP, а затем нажмите кнопку Select (Выбрать).

На экране появится окно диалога Application Tree (Структура прикладной программы).

Редактирование процедуры форм (Form)

1. В окне диалога Application Tree (Структура прикладной программы) выделите строку UpdateCustomer (Form), а затем нажатием правой кнопки мыши вызовите выпадающее меню.

Это меню позволяет воспользоваться инструментами генератора программ, не открывая окна диалога Properties (Характеристики процедур).



В меню, появившемся на экране, флажками отмечены инструменты, содержащие данные.

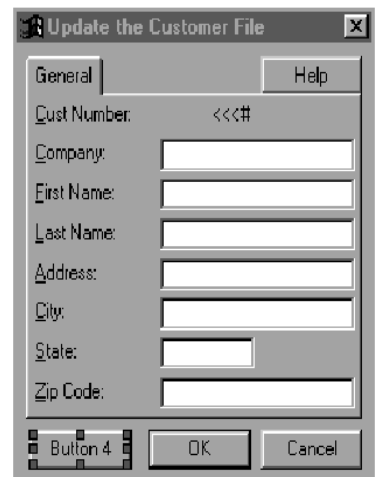
2. Нажмите кнопку Window (Окно).

На экране появится форматер окна Window Formatter. С его помощью Вы сможете легко отредактировать окно и его элементы управления.

3. Выберите команду Control > Push Button (Элемент управления > Кнопка) или выберите в панели инструментов элемент “Кнопка” (его пиктограмма выглядит, как кнопка ОК).

4. Разместите выбранный элемент управления ниже закладок, щелкнув для этого мышью в нижнем левом углу окна.

5. Выделите только что помещенную кнопку и щелкните правой кнопкой мыши. На экране появится выпадающее меню, из которого следует вызвать окно диалога Button Properties (Характеристики кнопки), выбрав для этого команду Properties (Характеристики).



6. В поле Text (Текст) введите Phones.

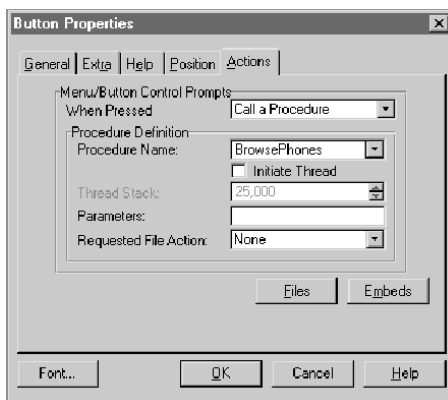
Таким образом, Вы изменили надпись на кнопке.

7. Выберите закладку Actions (Действия).

В окне с этой закладкой определяются действия, производимые элементами управления. В нашем примере мы хотим, чтобы по нажатию кнопки выполнялась процедура просмотра (Browse), отображающая на экране записи файла Phones, связанные с выделенной в данный момент цветом записью файла Customer.

8. Из раскрывающегося списка When Pressed (По нажатию) выберите значение Call a Procedure (Вызов процедуры).

В окне с закладкой Actions (Действия) откроется группа Procedure Definition (Определение процедуры).



9. В поле Procedure Name (Имя процедуры) введите BrowsePhones, а затем нажмите кнопку ОК.

Таким образом, Вы определили процедуру, которая будет вызываться по нажатию данной кнопки.

10. Выбрав в меню команду Exit!, вернитесь в окно диалога Application Tree (Структура прикладной программы), а затем на запрос о сохранении проделанных изменений, нажмите кнопку Yes.

К дереву процедур добавится процедура BrowsePhones с пометкой (To Do).

Использование Мастера просмотра (Browse)

Мастера процедур создают законченные процедуры, основываясь на минимальной информации, которую Вы вводите в предлагаемой Вам серии окон диалога.

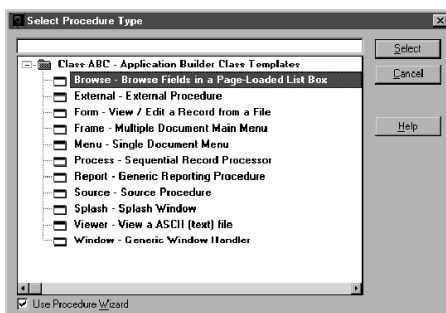
Вызов мастера просмотра

1. Выберите курсором процедуру BrowsePhones (ToDo) и дважды щелкните левой кнопкой мыши.

На экране появится окно диалога Select Procedure Type (Выбор типа процедуры). В этом окне Вы можете выбрать шаблон процедуры. Этим шаблоном воспользуется генератор программ Application Generator для создания исходного текста программы.

2. Выделите цветом шаблон процедуры просмотра (Browse), установите флажок Use Procedure Wizard (Использовать мастер процедуры), а затем нажмите кнопку Select (Выбрать).

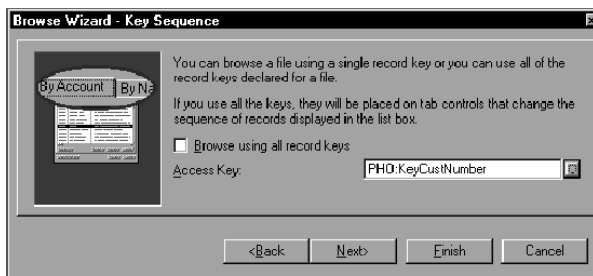
Установка флажка Use Procedure Wizard (Использовать мастер процедуры) включает использование мастеров процедур.



3. После того, как Вы прочтете всю информацию, содержащуюся в первом окне диалога Wizard (Мастер), нажмите кнопку Next (Дальше) и приступайте к созданию процедуры.



4. На вопрос “Which file do you want to browse?” (“Выберите файл для просмотра данных”) в поле ввода введите Phones, а затем нажмите кнопку Next (Дальше).

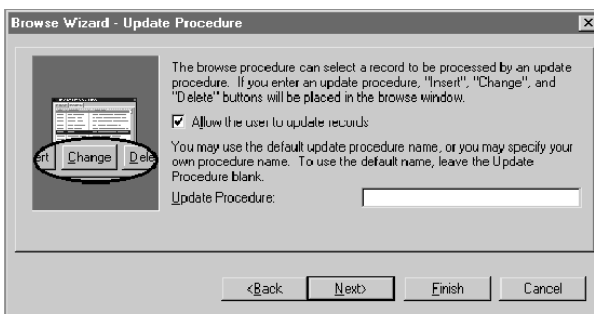


5. Снимите флажок просмотра с использование всех ключевых полей (Browse using all record keys), нажмите кнопку с многоточием (...), выберите в списке пункт PHO:KeyCustNumber, а затем нажмите кнопку Next (Дальше).

Таким образом, Вы задали ключ, который будет использоваться при сортировке записей для вывода их в окне списка.

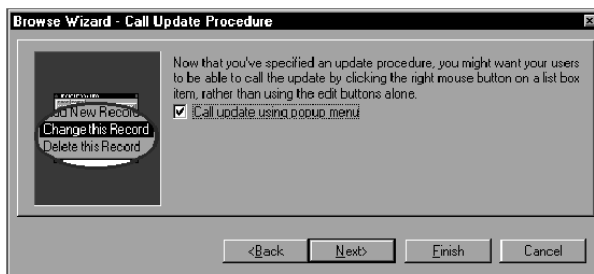
6. В поле ввода Update Procedure (Процедура обновления) введите UpdatePhones, а затем нажмите кнопку Next (Дальше).

Таким образом Вы указали процедуру, которая будет создана мастером просмотра (Browse Wizard) для обновления записей файла телефонов (Phones). В этом случае это не является необходимостью, так как Шаблон будет создавать код, чтобы редактировать этот файл по месту, отбрасывая потребность в процедуре Update. Однако, если мы не специфицировали EDITINPLACE в Опциях Пользователя, мастер просмотра действительно вызовет мастера форм, чтобы создать обновление процедуры форм для нас. Указание имени процедуры обновления на данном этапе избавляет от вызова мастера формы вручную.



7. Установите флажок Call update using popup menu (Обновление из всплывающего меню), а затем нажмите кнопку Next (Дальше).

Таким образом, Вы дадите пользователям своей программы возможность вызывать процедуру обновления из всплывающего меню (которое будет появляться, как только пользователь поместит курсор мыши в окно списка и щелкнет правой кнопкой мыши).



8. Снимите флажок Provide buttons for child files (Обеспечить кнопки для дочерних файлов), а затем нажмите кнопку Next (Дальше).



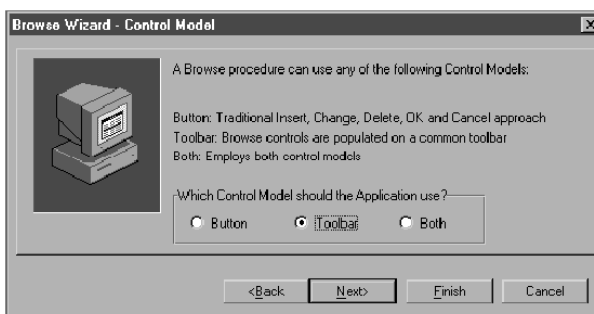
9. Выберите элемент Assume that the parent record is active (Проверка активности родительской записи), а затем нажмите кнопку Next (Дальше).

Таким образом, Вы обеспечили работу процедуры только с отображаемыми на экране записями файла Phones, связанными с записью файла Customer, в момент вызова процедуры, находившейся в памяти.



10. Снимите флажок Provide a “Select” button (Обеспечить кнопку выбора “Select”), а затем нажмите кнопку Next (Дальше).

Поскольку данная процедура не будет автоматически подставлять данные в поле ввода, необходимость в этой кнопке отсутствует.

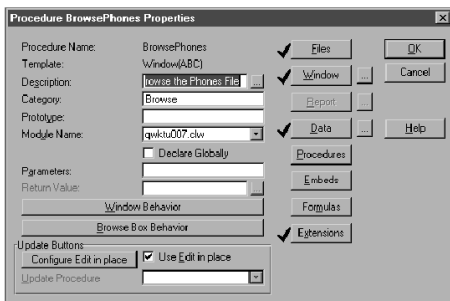


11. Задайте вариант использования *Toolbar* для Which Control Model should the Application Use (Какую модель управления будет использовать Приложение?) и нажмите кнопку Next.

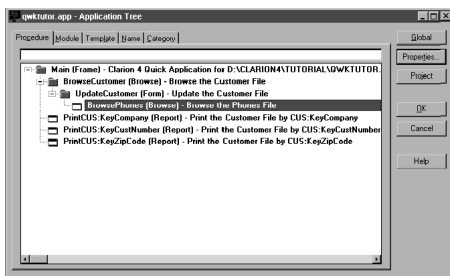
Это даст процедуре *BrowsePhones* возможность постоянного просмотра и согласованность с процедурой *BrowseCustomer*. При этом для вставки, обновления и удаления будут использоваться кнопки панели управления *Toolbar*.

12. Нажмите кнопку Finish (Готово), приняв значение флажка Overwrite existing procedures (Переписать существующие процедуры), установленные по умолчанию.

Теперь Мастер просмотра (Browse Wizard) начнет создание процедуры просмотра (Browse) данных файла и связанной с ней процедуры обновления (Form) записей файла Phones. Если EDITINPLACE не специфицировано в опциях пользователя файла Phone, шаблон просмотра будет создавать связанную процедуру обновления для сохранения записей файла Phones. По окончании создания процедур работа перейдет в окно Procedure Properties (Характеристики процедуры), относящееся к только что созданной процедуре просмотра.



13. Нажмите кнопку Ok и вернитесь в окно Application Tree (Структура программы).



Обратите внимание, что к структуре программы добавилась только что созданная процедура.

14. Выберите команду Project > Run (Проект - Выполнить) или нажмите на панели кнопку Run (Выполнить).

Теперь у Вас есть законченная прикладная программа реляционной базы данных. Эта программа обеспечивает многооконный просмотр данных главного файла, содержащего данные о счетах, ограничение числа просматриваемых данных подчиненного файла только записями, связанными с одной (текущей) записью родительского файла. Также эта программа содержит код, обеспечивающий

внутреннюю непротиворечивость базы (Referential Integrity), защищающий Вашу базу данных от подчиненных записей, “оставшихся без родителей”.

Нажмите новую кнопку Phones и добавьте записи файла Phones для каждого из ваших покупателей.

15. По окончании проверки работы Вашей новой программы выберите команду File > Exit (Файл - Выйти) и вернитесь в Clarion.

16. Закройте генератор программ Application Generator выбрав команду File > Close или нажав кнопку ОК.

И, наконец, Мастер прикладной программы (Application Wizard)!

И Мастер Quick Start, и Мастера процедуры просмотра (Browse), формы (Form), отчета (Report) являются очень мощными инструментами. Однако, Clarion содержит в своем арсенале сверхпродуктивных инструментов еще более мощное оружие.

Мастер прикладной программы (Application Wizard) на основе существующего словаря данных создает законченную прикладную программу. Используемый при этом словарь данных может содержать любое желаемое количество файлов. Эти файлы должны быть полностью определены, в том числе должны быть заданы отношения между файлами, меры по поддержанию внутренней целостности (Referential Integrity), правила целостности данных (Data Integrity), определяемые в редакторе словаря Dictionary Editor.

Для облегчения использования Мастера программы (Application Wizard), редактор словаря (Dictionary Editor) содержит параметры характеристик каждого файла, поля и ключа. Взаимодействие редактора словаря Dictionary Editor с Мастером программ (Application Wizard), а также со всеми остальными Мастерами процедур, позволяет Вам настраивать создаваемую программу.

Использование редактора словаря (Dictionary Editor) для задания параметров Мастера.

1. Выберите команду File > Pick (Файл - Выбрать) или нажмите на панели инструментов кнопку Pick (Выбрать).

2. Выберите закладку Dictionary (Словарь), выделите цветом строку C:\CLARION4\TUTORIAL\QWKTUTOR.DCP, а затем нажмите кнопку Select (Выбрать).

На экране появится окно диалога Dictionary Editor (Редактор Словаря).

3. Выделите цветом файл Customers, а затем нажмите кнопку Fields / Keys... (Поля/Ключи...).

4. Выделите закладку Keys (Ключи), щелчком мыши выделите строчку KeyCustNumber, а затем нажмите кнопку Properties (Характеристики).

На экране появится окно диалога Edit Key Properties (Редактирование характеристик ключа).

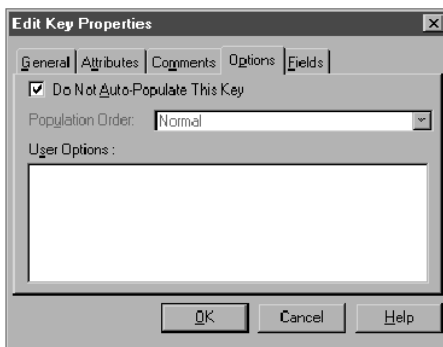
5. Выберите закладку Options (Параметры).

Установки, находящиеся на этой закладке, определяют поведение Мастеров. Параметры, расположенные в верхней половине окна, используются для стандартного набора шаблонов, поставляемых вместе с Clarion. Окно ввода User Options (Параметры пользователя) предназначено для использования шаблонов, поставляемых третьими фирмами и позволяет настраивать любые параметры этих Мастеров, не предусмотренные в стандартных шаблонах.

6. Поставьте флажок Do Not Auto-Populate (Не встраивать этот ключ автоматически).

Установка этого флажка указывает Мастеру прикладной программы (Application Wizard) не использовать сортировку по этому ключу при создании процедуры просмотра (Browse) данных файла. Данная операция уничтожает закладку CUS:KeyCustNumber в процедуре просмотра (Browse) файла Customer, содержащего информацию о покупателях.

Кроме перечисленных выше, редактор словаря обладает огромным числом возможностей по настройке выходных форм, создаваемых всеми мастерами. По окончании урока дополнительную информацию по этому важному вопросу можно получить в *Application Handbook* в разделе *Optimizing the Wizards*.



7. Нажмите кнопку ОК.

8. В окне диалога Field / Key Definition (Определение поля/ключа), а затем нажмите кнопку Close (Заккрыть).

9. В окне диалога Dictionary (Словарь) нажмите кнопку Close (Заккрыть). На экране появится запрос на сохранение файла словаря, нажмите кнопку Yes (Да).

Вот и все. Далее, с помощью Мастера прикладной программы (Application Wizard) мы создадим законченную программу.

Использование Мастера прикладной программы (Application Wizard)

Мастер прикладных программ создает законченные приложения, используя минимальный набор сведений, вводимых в последовательности окон диалога.

Создание новой прикладной программы

1. Выберите команду File > New (Файл - Новый).

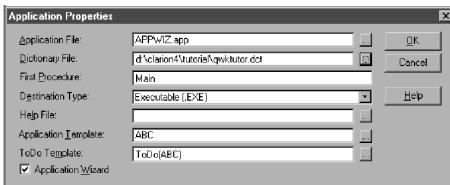
На экране появится окно диалога New (Новый).

2. Из списка каталогов Directories выберите каталог C:\CLARION4\TUTORIAL.

3. В поле File Name (Имя файла) введите APPWIZ.APP.

4. Снимите флажок Use Quick Start Wizard (Использование Мастера Quick Start), а затем нажмите кнопку Create (Создать).

На экране появится окно диалога Application Properties (Характеристики прикладной программы).

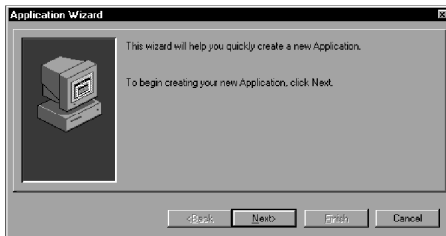


5. Нажмите кнопку с многоточием (...), расположенную справа в поле ввода Dictionary File (Файл словаря).

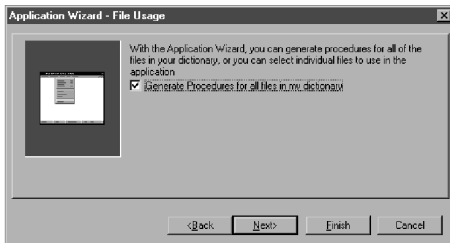
На экране появится окно диалога Select Dictionary (Выбор словаря).

6. Выделите цветом файл c:\clarion4\tutorial\qwkttutor.dct, а затем нажмите кнопку Open.

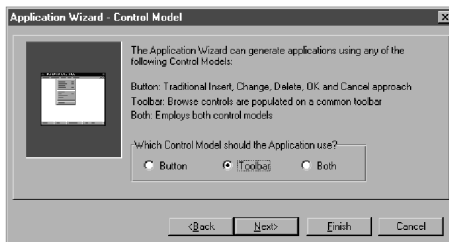
7. Установите флажок Application Wizard (Мастер прикладной программы), а затем нажмите кнопку OK.



8. Нажмите кнопку Next (Дальше).



9. Нажмите кнопку Next (Дальше).

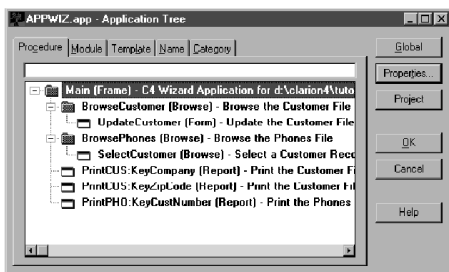


10. Нажмите кнопку Next (Дальше).



11. Нажмите кнопку Finish (Готово).

Мастер прикладной программы (Application Wizard) создаст приложение.



12. Выберите команду Project > Run (Проект - Выполнить) или нажмите на панели инструментов кнопку Run (Выполнить).

Примите наши поздравления! Выполняется Ваша вторая, полностью созданная прикладная программа.

Теперь Вы можете проверить ее работу и сравнить с уже созданной перед этим программой. Вы заметите, что созданные программы несколько различаются. Различия не сложные, так как мы проектировали первое приложение поэтапно (по шагам), вместо того, чтобы использовать Application Wizard (Мастер приложений).

Что дальше

Если вы можете сделать так много, используя только ярлычки Clarion, чего же можно ожидать от согласованной работы всех других инструментов, обеспеченных Clarion?

Наравне с Мастерами, Clarion предоставляет множество других эффективных инструментов, которые можно использовать в генераторе программ (Application Generator). И первые из этих инструментов - шаблоны элементов управления (Control), расширений (Extension) и кода (Code), способные добавить функциональность любой процедуре. С помощью этих инструментов процедуры изменяются и настраиваются также легко, как и с помощью Мастеров - и ни один из них не потребует программирования вручную.

Следующие главы этой книги отражают важность философии разработки приложений, которую выражает Clarion. Пожалуйста, читайте внимательно. Это важная основа материала, потому что как вы могли слышать, Clarion отличается от других средств разработки. Понимание философии Clarion облегчит восхождение к вершинам.

Руководство “Генератор приложений” (Application Generator) размещено в книге “Изучение Clarion” (*Learning Clarion*). Это настольное руководство проведет вас через создание системы ввода без использования механизма шаблонов. Продемонстрирует работу всех механизмов Clarion, покажет вам, какую мощь вы имеете, когда в конце вы самостоятельно напишите некоторый код, обеспечивающий “нестандартную функциональность”.

Продолжайте! Вы только скользнули по поверхности, а есть еще много всего!

3

Эволюция: обзор

Дао Clarion

В Китае понятие "дао" означает "Путь". Clarion предоставляет совершенно новый путь создания прикладных программ под Windows - это Вы увидите по программе, которую Вы создадите приблизительно за тридцать минут! Однако Clarion - это намного больше, чем Мастера, и поэтому мы уделим немного времени на объяснение основных концепций, положивших основу набору инструментов Clarion.

Clarion - язык программирования!

В основании Clarion(инструмента разработки) лежит язык программирования Clarion - бизнес-ориентированный язык программирования четвертого поколения для общего применения. Язык Clarion в ваших руках является настоящим инструментом. Но Clarion - это больше, чем просто язык Clarion, это также полный набор инструментов для создания и поддержки приложений коммерческих баз данных.

Верхним слоем языка Clarion является Среда разработчика - полностью интегрированный набор инструментов, предназначенных для быстрой разработки прикладной программы (RAD) и быстрой поддержки приложений (RAM). Все эти инструменты используются для создания кода языка Clarion и избавляют от необходимости написания текста программы вручную. Вот почему Clarion for Windows является таким мощным средством - оставляя возможность написания кода вручную, набор инструментов создаст большую часть "стандартного" текста, позволяя сконцентрироваться на собственных задачах приложения.

Уровни абстрагирования

Чем дальше программист отходит от "голого металла" компьютера (двоичной системы), тем более продуктивным становится его труд.

Немного истории

Первые программы писались в двоичных кодах, а труд программиста был очень тяжел. На смену двоичному коду пришел язык ассемблер, призванный облегчить труд программистов, поскольку код, создаваемый на этом языке, был более понятен при чтении. Любой конструкции языка ассемблера соответствовала только одна

машинная конструкция.

Программисты на ассемблере заметили, что для реализации стандартных программ им постоянно приходится писать одни и те же последовательности директив. Вследствие этого были созданы языки программирования третьего поколения (такие, как фортран, Си, PL/1 и т.д.), где одной конструкции языка могло соответствовать несколько машинных кодов, реализующих стандартные задачи. Программирование стало более продуктивным и перешло на более высокий уровень абстрагирования, чем “голый металл” компьютера. Был только один недостаток - создавались “стандартные” последовательности машинных кодов, и таким образом, утратилась гибкость, возможная при работе непосредственно с языком ассемблер.

Современность- языки четвертого поколения

Преимущества языков четвертого поколения, таких, как Clarion, были достигнуты сходным образом. Например, язык программирования Clarion, автоматически обрабатывает все стандартные функции Windows (такие, как перерисовка экрана). При работе с языками третьего поколения программистам требуется писать обработку этих функций самостоятельно. В языке Clarion программа “Здравствуй, Мир!” состоит из 9 строчек исходного текста, а в языке C/C++ для создания этой же программы пришлось бы написать более 80 строчек!

Обобщая вышесказанное: чем выше абстрагированность, тем более продуктивным может быть программирование. С каждым следующим уровнем программисты становятся более продуктивными, заставляя компьютер делать за них все больше работы - создавать стандартный код, реализующий стандартные задачи. Этот фундаментальный принцип лежит в основе набора инструментов Clarion.

Секрет продуктивности

При работе с более высокими уровнями абстрагированности возникает «проблема» потери гибкости достижения задуманного результата. Эта проблема решена в Clarion.

Секрет настоящей продуктивности программиста, достигаемой без потери гибкости, - работа на наиболее высоком уровне абстрагированности, позволяющем достичь желаемого результата. Набор инструментов Clarion содержит несколько уровней абстрагированности, что позволяет выбрать уровень, наиболее подходящий для достижения желаемого результата. При этом Вы «не приклеены» к высшему

уровню и избавлены от необходимости «возиться» на нижнем уровне.

Набор инструментов, который создает объектно-ориентированный код для вас, использует библиотеку классов Clarion (*Application Builder Class (ABC) Library*) - это верхний уровень - и когда вам необходимо, вы можете добавить собственный код - нижний уровень, - чтобы повысить уровень функциональности.

Другими словами, Секрет Продуктивности (о котором вы читали чуть раньше), заключается в том, чтобы работать изящно, без напряжения. И что уж совершенно верно, так это то, что Clarion реализован так, чтобы позволить вам сделать это - он создает за вас стандартный код, так как вы не хотите делать это; этот код достаточно гибок для того, чтобы позволить вам добавить “звоночки и свисточки” (Это такая шутка.)

«Уровни абстрагированности» в Clarion

К моменту первого знакомства с Clarion Вы уже поработали с наивысшими уровнями абстрагированности Clarion: с Мастерами. Далее Вам предстоит познакомиться с большей частью перечисленных ниже мастеров.

Редактор словаря

Ваш Словарь Данных - простой репозиторий для всех определений файлов данных.

Мастер программ

Основываясь на определениях, содержащихся в файле данных и настройках словаря данных, создает за несколько минут законченную «стандартную» прикладную программу.

Мастера процедур

Основываясь на определениях, содержащихся в файле данных и настройках словаря данных, создают «стандартные» процедуры (например, список окна просмотра или форму ввода данных).

Шаблоны процедур

Создают для вас «стандартные» процедуры (например, рамку основного меню программы или отчет). При этом используются конструкции окна или отчета, созданные в форматере окна или отчета, и сведения, введенные вами на запросы о

том, как процедура должна работать.

Шаблоны элементов управления

Добавляют в конструкции окон или отчетов «стандартные» элементы управления (например, элемент управления «Дерево» для связанных файлов) и создают все коды для обеспечения работы элементов.

Шаблоны расширений

Создают коды процедуры, обеспечивающей свойства, не связанные с какими-либо элементами управления окна (например, экран Дата/Время в строке состояния окна программы).

Шаблоны кода

Создают код процедур, как правило, выполняющих одну задачу, связанную с конкретным элементом управления окна (например, проверку ввода данных).

Встраиваемый код

Помещает некоторый Ваш собственный) код на языке Clarion в любую из многочисленных точек встраивания, расположенных в любой из процедур. Используется для настройки и/или изменения поведения кода, созданного по шаблону.

Шаблон источника

Шаблон, предназначенный для написания Ваших собственных процедур или функций на языке Clarion, в процессе создания программы с помощью генератора программ.

Интерфейс Windows и стандартная библиотека Си

При необходимости, из любого написанного кода на языке Clarion (целых процедур или точек встраивания) можно непосредственно вызвать функции интерфейса Windows (Windows API) или стандартной библиотеки Си. Прототипы этих функций уже созданы и входят в Professional и Enterprise поставки Clarion.

Код Си/Си++ или Модула-2

Можно написать любую необходимую функцию TopSpeed C/C++ или Modula-2 (компилятор включен в Professional и Enterprise поставки и продается отдельно)

и напрямую связать ее с программой, написанной на Clarion.

Как видно, уровни абстрагированности Clarion позволяют соединить любые уровни от мастера программ до написания кода на языке Си (или Ассемблере для желающих помучиться).

Ключ Clarion к максимальной продуктивности

Всегда работайте на наиболее высоком из подходящих уровне абстрагированности:

- Используйте Мастер программ для создания «начальной точки» программы.
- Используйте Мастера процедур для добавления новых процедур.
- Необходимые изменения в любом шаблоне процедур осуществляется одним из двух интерактивных инструментов Clarion (форматером окна или отчета).
- При необходимости добавьте шаблоны элементов управления, расширений и кодового шаблона.
- Настройки осуществляются с помощью встраиваемого кода.
- Если требуется что-то, выходящее за возможности имеющихся шаблонов, используйте шаблон источника и напишите код на языке Clarion (или возьмите подходящий шаблон, поставляемый третьими фирмами).
- Опускайтесь до уровней интерфейса Windows и языка Си/Си++ только в случае крайней необходимости.

Это и есть философия Clarion, позволяющая работать эффективнее, особо не затрудняясь, используя для «грязной работы» набор инструментов. Следуйте этим советам, и Вы обнаружите, что продуктивность, по сравнению с любыми другими ранее использовавшимися инструментами, возросла.

Программирование, осуществляемое с помощью шаблонов

Генератор программ (Application Generator) работает с шаблонами. Т.е. он является инструментом, изменения которого определяются требованиями шаблона, используемого Вами для создания программы. Шаблоны процедур, элементов управления, кодов и другие возможные шаблоны обеспечивают огромный рост продуктивности при создании программы. Они дают возможность использования множества полезных свойств объектно-ориентированного программирования, в особенности многократное использование одних и тех же модулей. Шаблоны незаметно создают эффективный объектно-ориентированный код для вас, используя библиотеку *Application Builder Class (ABC)*. Шаблоны, поистине, ключ Clarion к

быстрому созданию приложений.

Что такое шаблон?

В Clarion шаблон не является неизменным, а полностью интерактивным инструментом, запрашивающим необходимую для создания исходного кода прикладной программы информацию. Если Вы измените информацию, при следующей компиляции создастся другой код, отличный от предыдущего.

Шаблоны Clarion позволяют вам вставить собственный код в любую из многих точек вставки, доступных вам в каждом шаблоне. У вас нет необходимости сохранять весь сгенерированный код, чтобы достичь успеха. Гарантировано, что ваши настройки не будут затерты последующими генерациями - шаблон просто породит ваш код вместо стандартного.

Реестр шаблонов (REGISTRY.TRF) сохраняет информацию обо всех уже написанных выполняемых файлах, кодах и структурах данных, которые могут быть настроены по-новому и заново использованы. Вы можете использовать реестр шаблонов для изменения конструкции по умолчанию любого окна или отчета шаблона, придав им вид, который они будут затем принимать каждый раз при первом создании процедуры.

Вы можете изменить стандартный шаблон языка Clarion и сохранить эти изменения в реестре. Также Вы можете добавить шаблоны других поставщиков, написать свои собственные и *пользоваться ими наравне со стандартными* (в случае Professional или Enterprise поставки Clarion). Это делает генератор программ бесконечно расширяемым инструментом.

Как пользоваться шаблонами?

При создании новой процедуры с использованием шаблона Вы выбираете шаблон процедуры, создающий код, наиболее подходящий для осуществления поставленных задач, а затем настраиваете его с помощью инструментов разработки. Шаблоны процедур включают в себя такие элементы, как “окна просмотра” для просмотра групп записей и “окна форм” для единовременного редактирования одной записи. Если процедура представляет собой окно с меню, команды (действия) меню автоматически добавляются к структурному дереву процедур программы и отмечаются параметром “ToDo”, как любая другая процедура.

Обычно для настройки процедуры вызывают один из форматов - Window Formatter (Форматер окна) или Report Formatter (Форматер отчета) - и добавляют, например, новое окно или элемент управления. Форматеры являются визуальными инструментами конструирования, позволяющими выбрать для конструирования окно или отчет методом “выбери и щелкни” (“point and click”). Вам достаточно выбрать в панели инструментов требуемый элемент управления, разместить его, щелкнув мышью под желаемой позицией, а затем, щелкнув правой кнопкой мыши, начать редактировать его свойства.

С помощью шаблонов элементов управления, кода и расширений Вы можете увеличить функциональность уже созданной процедуры. Шаблоны элементов управления содержат элементы управления и весь стандартный выполняемый код для управления ими. Все предварительно встроенные элементы управления, появляющиеся в окнах, созданных по умолчанию шаблонами процедур, в действительности являются шаблонами, которые можно настроить для всех функций процедуры.

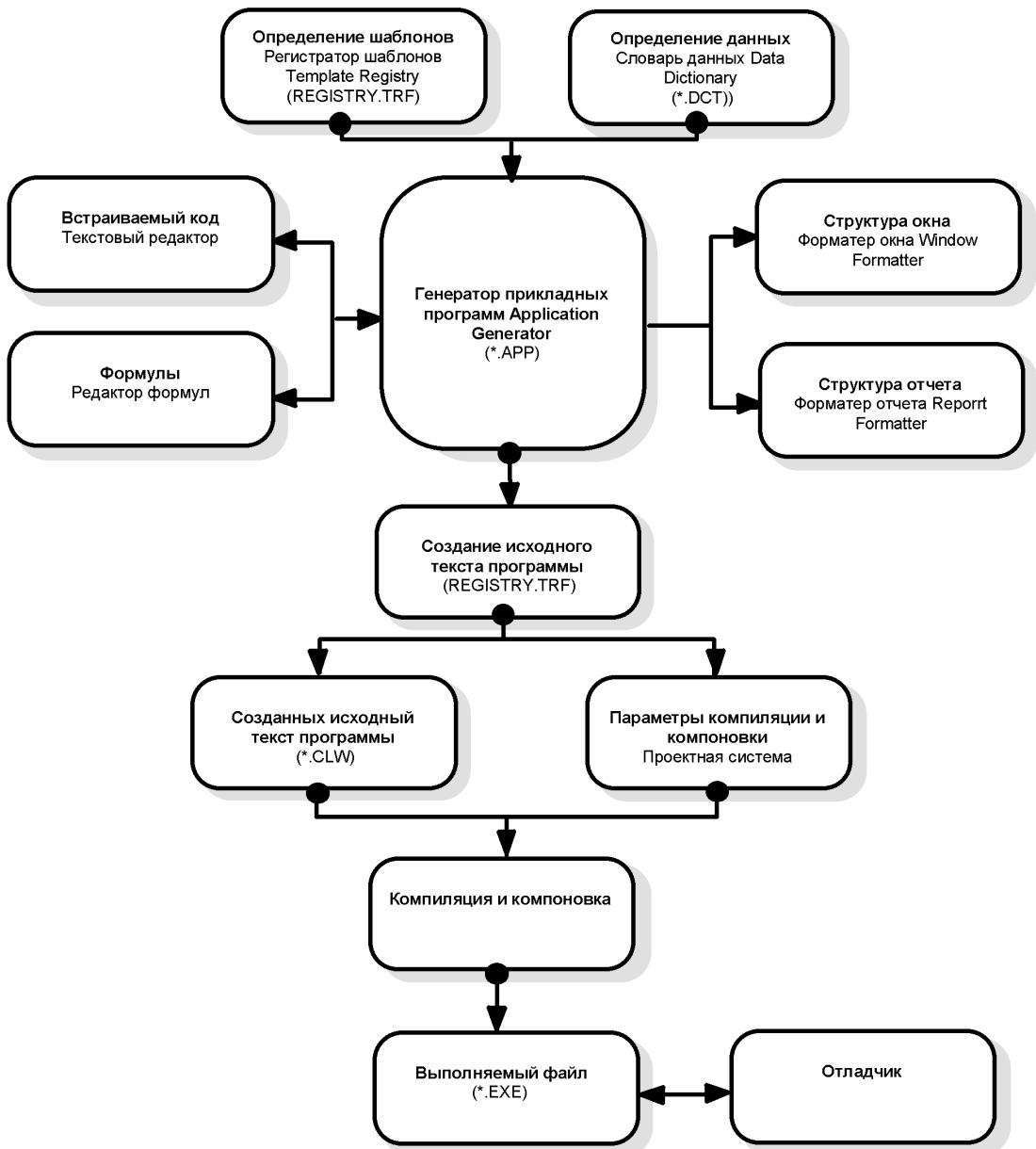
Шаблоны расширений добавляют выполняемый код, повышающий функциональность процедуры.

Также Вы можете настроить процедуру и другим способом - добавив встроенный (embedded) исходный код. Генератор программ (Application Generator) отображает структурное дерево, содержащее все имеющиеся “точки вставки”: находящиеся до, после и в самой процедуре, а также все “точки вставки”, относящиеся ко всем событиям окна или элементов управления, созданных процедурой. Для создания желаемого кода Вы можете выбрать точное место для выполняемого кода, а затем, при необходимости, изменить его вручную, или использовать готовые “шаблоны кода”. Генератор программ Application Generator создает исходный код программы на основе шаблонов и всех ваших настроек (включая встроенный исходный код).

Clarion предоставляет богатый ассортимент уже написанных шаблонов процедур, с помощью которых Вы можете быстро создать прикладную программу типа базы данных. В главе “Обучение работе с Quick Start” данного руководства Вы познакомились с Мастерами. Руководство по генератору приложений в книге

“Изучение Clarion” познакомит вас с использованием шаблонов Clarion для создания приложений под Windows, необходимых вам.

Среда разработки Clarion



Среда разработки Clarion состоит из семи функциональных частей. К каждой части среды можно обратиться из любой другой ее части. При работе с генератором программ Application Generator, нажатием на кнопки, появляющиеся в различных окнах диалога, можно перейти в остальные части среды.

Диаграмма разработки прикладной программы, представленная в начале этой главы, отображает взаимодействие частей среды друг с другом и реестром шаблонов. В центре всего процесса создания находится генератор программ Application Generator.

В этом разделе описываются все части среды. Они представлены в том порядке, в каком обычно могут использоваться программистом, работающим с генератором программ Application Generator. Описание каждой части содержит пример окон диалога, которые программист должен заполнить, “описывая” генератору программ Application Generator свойства своей программы.

Программирование в Clarion, в большинстве своих проявлений, является просто “прогулкой” по серии окон диалога. Хотя некоторые файлы следует создавать раньше других, строго определенной последовательности заполнения этих окон не существует.

Редактор словаря Dictionary Editor

Словарь данных Data Dictionary (файл с расширением .DCT), управляемый редактором словаря Dictionary Editor содержит описание базы данных, включая ее файлы, поля, связи, правила проверки полей, меры по поддержанию целостности (referential integrity constraints) и многое другое. *При конструировании прикладной программы этот файл создается первым.*

Определения файла можно как создать с нуля, так и импортировать из существующих файлов данных. Если у вас поставка Enterprise Edition, вы можете, воспользовавшись Синхронизатором, синхронизировать словарь данных Clarion со словарем данных конечного пользователя. Это позволяет вам получать все описания файлов сразу из SQL баз конечного пользователя (невзирая на гигантскую работу); это позволит сохранить словарь данных синхронизированным с SQL RDBMS все время, не смотря на изменения, которые может сделать Администратор .

Остальные части среды разработки используют информацию, заданную Вами в словаре, позволяя, например, просто разместить поля данных в конструируемом окне диалога или отчете. Основываясь на определениях, заданных Вами в словаре

данных Data Dictionary, генератор программ Application Generator создает исходный текст всех конструкций, обращающихся к файлу данных. Мастер программ может создать полностью функциональную программу, основываясь только на словаре данных!

Генератор программ

Генератор программ Application Generator создает исходный текст Вашей программы, по одной процедуре за раз, основываясь на шаблонах, выбранных Вами из реестра шаблонов. Он требует добавления локальных и глобальных переменных памяти, а также настройки процедур средствами визуального конструирования и встраиваемого кода.

Генератор программ дает доступ к другим инструментам среды разработки, поэтому вы можете настраивать просмотр и функциональность окон, меню, отчетов и других элементов пользовательского интерфейса вашего приложения. Шаблоны предоставляют собственным элементам управления (тем, что появляются в Генераторе программ) возможность обеспечивать вас информацией, с помощью которой они настраиваются под требуемую функциональность.

Форматер окна Window Formatter

В форматере окна Window Formatter Вы можете визуальное сконструировать окна и элементы управления Вашей программы: на экране Вы будете видеть то, что увидит конечный пользователь программы.

Форматер отчета Report Formatter

Форматер отчета Report Formatter генератора программ Application Generator работает точно также, как и форматер окна Window Formatter. На простой странице отчета Вы размещаете элементы управления. Во время выполнения средство печати обрабатывает указанные записи, концы страниц и групп, заголовки и колонтитулы.

Текстовый редактор

Текстовый редактор является редактором для программирования функций, в котором Вы может создавать исходный текст программы вручную. При работе в генераторе программ Application Generator текстовый редактор обычно используется для создания встроенного исходного текста (кода), настраивающего работу процедур. Конечно, Вы можете и написать полностью всю программу с нуля.

Редактор оснащен цветовым синтаксическим выделением, облегчая распознавание различных частей языковых конструкций Clarion. Этот редактор, как и все стандартные инструменты редактирования, содержит полные процедуры поиска (search) и замены (replace) текста.

Редактор формул

Редактор формул помогает быстро создавать и использовать как простые, так и сложные конструкции, вычисляющие значения. Этот редактор обеспечивает проверку синтаксиса, а так же одновременный доступ ко всем используемым при вычислении переменным, функциям и операторам.

Проектная система Project System

Генератор программ Application Generator автоматически создает проект (.PRJ) для вашей прикладной программы. Проект содержит параметры компилятора и компоновщика, такие, как включение кода отладчика, выбор оптимизации, дополнительный файл драйвера, библиотеки и другие внешние файлы для процессов компиляции и компоновки.

Отладчик Debbuger

При отладке программа обычно запускается на выполнение, затем выполнение программы многократно приостанавливается для проверки значений различных переменных и возобновляется вновь. Отладчик Debbuger (16 и 32 битный) состоит из набора окон, отображающих исходный текст программы, содержание переменных, активные процедуры и другую информацию.

Существуют три главы по использованию отладчика в Руководстве пользователя. Мы рекомендуем вам изучить их после освоения материала, предоставленного в *Getting Started and Learning Clarion*.

Встроенная система подсказок

Почти из каждого экрана Clarion for Windows можно обратиться к расширенной контекстнозависимой системе подсказок. Для этого достаточно нажать кнопку Help (Помощь) или клавишу F1.

В разделе “Общие вопросы” встроенной системы подсказок содержатся советы и примеры решения наиболее общих задач приложений. Поместите код требуемого примера в буфер обмена, вырезав его непосредственно из системы подсказок, а затем вставьте из буфера в вашу программу. Для этого выберите команду Help - Contents (Подсказка - Содержание), а затем нажмите кнопку Common Questions (Общие вопросы).

Встроенная система подсказок Clarion состоит из основного файла подсказки и нескольких вспомогательных файлов. Сведения в файлах системы подсказки можно искать по ключевому слову или по алфавитному указателю. Чтобы воспользоваться поиском по ключевому слову или алфавитному указателю во вспомогательном файле, откройте требуемый файл непосредственно в Диспетчере программ (Program Manager) или в Windows Explorer.

Что дальше?

Итак, куда двигаться дальше для совершенствования знаний о Clarion?

- Лучшее для этого место - книга “Изучение Clarion” (Learning Clarion). Две его части содержат описание всех инструментов разработчика.
- TopSpeed предлагает вам посетить семинары; звоните в пользовательский сервис (800) 354-5444 или (954) 785-4555, чтобы вас включили в списки обучающихся.
- Присоединяйтесь к своей региональной Clarion User Group и участвуйте в общем образовательном процессе, общаясь с другими пользователями.
- Участвуйте в TopSpeed’овских конференциях на CompuServe (GO TOPSPEED) или в Internet *comp.lang.clarion* и общайтесь с пользователями всего мира (Настоятельно рекомендуем!).

Счастливо и до встречи !