

Container Dynamic Engine (CDE)

При работе с данными приходится иметь дело со следующими размещениями информации

- ✓ Сервер базы данных
- ✓ Настольные базы данных
- ✓ Данные в оперативной памяти

Каждое размещение информации имеет массу реализаций. Но если для первых двух существует стандартный подход, опробованные драйверы и общепризнанные стандарты, то последний вариант может вести себя как угодно в зависимости от реализации конкретного приложения. Со временем наша команда пришла к выводу, что необходимо стандартизировать механизмы доступа к данным в оперативной памяти и реализовать его в виде собственных библиотек.

Новая технология была названа CDE - Container Dynamic Engine. То есть это - динамичный механизм контейнеров. Именно в такой контейнер и размещаются данные. Представленный механизм имеет стандартный интерфейс взаимодействия с другими компонентами, с файловой системой и интернет-серверами. Он выполнен в виде готового Delphi-компонента, который встраивается практически во все наши программные продукты.

Технология CDE поддерживает отложенные изменения, синхронизацию с помощью встроенных автоматических первичных ключей, контроль типа вводимых данных и пропущенных полей. Помимо серверных библиотек CDE имеет подсистему сохранения в виде файлов (форматы: txt, xml, dbf). Это дает возможность эффективной работы с небольшими объемами данных в монопольном режиме с сохранением потоков данных непосредственно на компьютере пользователя. Это бывает особенно полезно при сохранении выборки для дальнейшего анализа данных в виде отчетов, графиков, многомерных кубов и так далее.



Рис 1. CDE в оперативной памяти. Простой вариант

CDE так же имеет серверную часть написанную на языке PHP. Она которая может быть установлена практически у любого провайдера либо на сервере клиента. Поддерживаются семейства операционных систем как Windows, так и Linux.

Данные могут передаваться в двух направлениях:

- ✓ С рабочей станции на интернет-сервер
- ✓ С интернет-сервера на рабочую станцию

С помощью CDS на рабочей станции данные могут:

- ✓ Обработываться непосредственно в памяти
- ✓ Сохраняться в настольную базу данных
- ✓ Сохраняться на SQL сервере
- ✓ Сохраняться в виде текстового файла
- ✓ Сохраняться в виде специальных составных файлов, включающих метаданные (например, ETN-файлов)

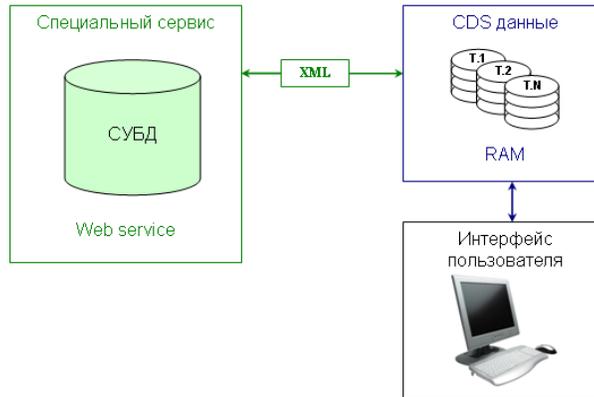


Рис 2. CDE с использованием web-сервиса.

Используя технологию CDE, разработчик может спроектировать таблицы базы данных точно так же, как если бы он это делал для настольных (DBF, Paradox) систем или для SQL-серверов. Точно так же создаются таблицы, контролируется типизация данных, необходимость заполнения полей и так далее. Однако такие таблицы не требуют никак инсталляции программ для их обработки.

Такой подход имеет и положительные и отрицательные стороны. Так на пример CDE пока не поддерживает ссылочную целостность таблиц, что может быть серьезным недостатком при вводе (модификации) данных. На сегодняшний день CDE не использует индексов, что так же может увеличивать скорость обработки данных.

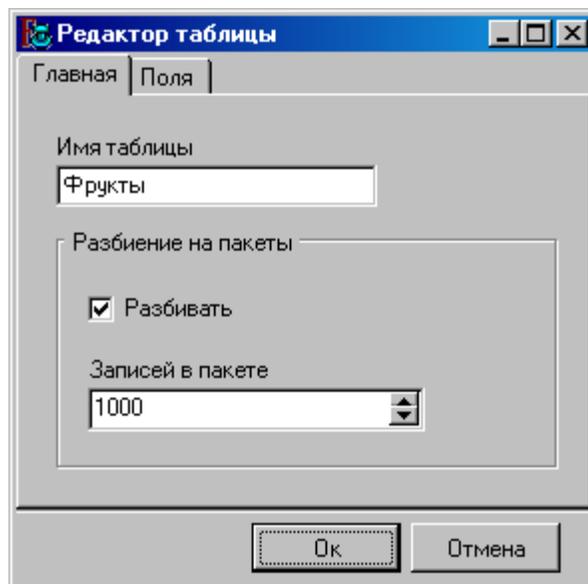


Рис 3. Создание таблицы с использованием технологии CDE.

На рисунках 3 и 4 представлены иллюстрации создания таких таблиц для их последующей обработки в оперативной памяти компьютера на рабочем месте. Как видно из первой закладки для новой таблицы достаточно указать ее название и ее разбику на пакеты, которые играют роль для медленных каналов связи сети Интернет.

Чем меньше потенциальная скорость передачи, тем меньше количество записей следует вставлять пакет. Это же касается и стабильности канала и организации подтверждения транзакции при прохождении пакета (настраивается на стороне web-сервера). Размер пакета так же важен при сохранении в файловой системе для больших объемов информации.

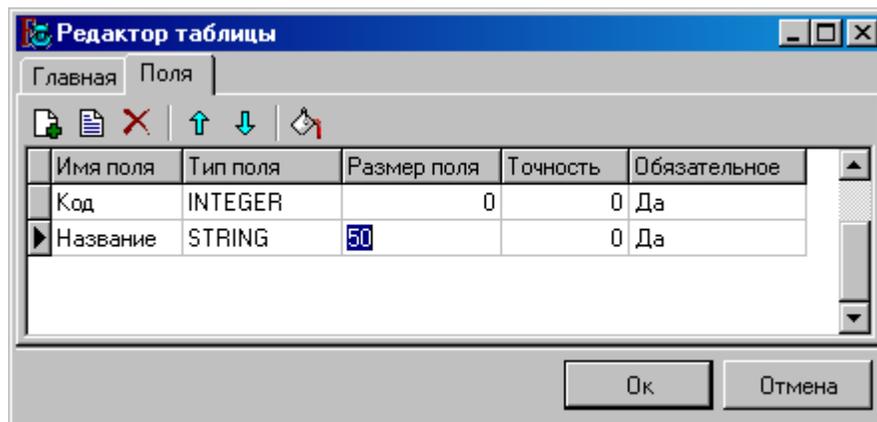


Рис 3. Настройка полей таблицы CDE.

Для существующих виртуальных таблиц CDS (Client Data Set) в оперативной памяти предусмотрен все тот же набор возможностей работы с реляционными базами данных. Это и связывание таблиц, и построение диалоговых окон для взаимодействия с пользователем, сохранение данных на жесткий диск, экспорт данных в другие форматы, передача на интернет-сервер. Важная особенность представляемой технологии в том, что заполнение виртуальных таблиц может происходить путем выполнения SQL-запроса на стороне провайдера, но результаты будут представлены в памяти, а затем сохраняться (по желанию) на локальном диске пользователя.

Перечень основных модулей CDE на стороне сервера

Модуль	Назначение
custom.php	Базовый контроллер
fly.php	Контроллер Fly
miniweb.php	Контроллер MiniWeb
si_cdstable.php	База данных в памяти (XML)
si_cdstable.php	CDS таблица
si_dataset.php	Интерфейс набора данных
si_dbcharge.php	Класс для работы с передачей данных (репликация)
si_dbchargelist.php	Список заливок (репликаций)
si_editabledataset.php	Интерфейс редактируемого набора данных
si_fly.php	Основной класс для работы с etni файлами
si_table.php	Таблица. Базовый класс работы с таблицами.

Перечень основных модулей CDE в настольных приложениях

Модуль	Назначение
SiCashLoader.pas	Реализует выбор провайдера для загрузки данных
SiCashLoaderEditor.pas	Редактор компонента TSiCashLoader
SiCashProvider.pas	Обеспечивает загрузку данных из БД (BDE, ADO) в клиентскую БД (XML)
SiCDSDatabase.pas	База с клиентскими источниками
SiCDSQuery.pas	Выполнение SQL запросов на выборку данных
SiCDSSession.pas	Сессия с клиентскими базами
SiCDSStorage.pas	Обеспечивает загрузку/выгрузку данных клиентской БД в файл
SiCDSTable.pas	Таблица для клиентской БД
SiCDSTableEditor.pas	Редактор свойств полей клиентской таблицы (SiCDSTable)
SiClientFusionProvider.pas	Динамический провайдер подгружается из DLL
SiDBCharge.pas	Переброска данных из источника в БД
SiDBChargeEditor.pas	Переброска данных из источника в БД

Для того чтобы познакомиться на практике с технологией CDE можно обратиться к описанию ETNI Fly на нашем сайте http://www.silentium.ru/main.php?part=etni_fly