59 Опишите технологию DDE

В Windows имеется несколько технологий, позволяющих организовать эффективное взаимодействие группы приложений. Еще в первых 16-разрядных версиях Windows была реализована технология DDE (Dynamic Data Exchange, динамический обмен данными). С ее помощью программа-сервер может обмениваться информацией  (текстовыми строками) с программами-клиентами, подключенными к этому серверу с помощью протокола DDE. Абсолютное большинство приложений корпорации Microsoft поддерживают этот протокол, что позволяет обращаться к ним из других программ, передавать и получать информацию.

В системе Delphi 7 для поддержки технологии DDE созданы четыре компонента, расположенные на панели System (Системные).

* TDdeClientConv- устанавливает связь с серверам DDE;
* TDdeClientltem- конкретный объект, содержимое которого передается (или  принимается) за один сеанс связи с сервером DDE, обеспечивает транэакционную работу по обмену информацией;
* TDdeServerConv- сервер DDE;
* TDdeServerltem- объект стороны сервера, содержимое которого предназначено для обмена.

Компонент TDdetLientConv обладает двумя важнейшими свойствами, лежащими в основе протокола *DDE.* Свойство DdeService описывает серверное приложение, с которым устанавливается связь, а свойство DdeTopic — так называемый «топик» (topic), характеризует имя серверного объекта, непосредственно участвующего в обмене информацией.

С помощью компонентов TDdeServerConv/TDdeServerltem можно включить в свою программу богатые возможности обмена данными, хотя для этих целей компания Microsoft предлагает значительно более мощную и гибкую технологию OLE. Тем не менее, поддержка механизма DDE в продуктах корпорации сохраняется, и в некоторых случаях использовать этот протокол обмена значительно проще, чем вес остальные.

Сервером DDE является, например, электронная таблица Excel. Написав несколько строк кода, можно организовать простой обмен содержимым между программой и ячейками страниц Excel.

Пример. Разместим на форме текстовое поле, две кнопки (Послать и Принять) — и компонент TDdeQientConv.

Основная задача при использовании компонентов DDE — установить связь между сервером и клиентом и, вообще, выяснить, поддерживает ли некоторое приложение этот протокол. Проще всего сделать это так. Запустите программу-сервер (в нашем случае — Excel), выберите произвольную ячейку и скопируйте ее содержимое в буфер обмена Windows. Затем перейдите к Инспектору объектов и для любого из свойств DdeService/DdeTopIc вызовите редактор связи с сервером DDE Info, щелкнув на кнопке  вызова в соответствующей строке Инспектора.

Если программа, данные из которой были скопированы в буфер, способна работать как сервер DDE, то кнопка Paste Link (Установить связь) доступна. В противном случае она окажется в сером «отключенном» состоянии.

После щелчка на этой кнопке в поля DdeServke/QdeTopic будут внесены названия сервера (Excel) и объекта — первого листа таблицы [Книга1]Лист1, если в программе Excel был открыт новый файл. Теперь надо подтвердить связь щелчком на кнопке ОК.

Обмен данными может происходить в обе стороны: как в сторону электронной таблицы, так и от нее к клиентскому приложению. Это приложение управляет работой сервера, отдавая ему соответствующие команды.

Чтобы послать данные серверной программе, используется следующий метод

function PokeData (Item; string,- Data: PChar) : Boolean;

Первый параметр описывает принимающий объект стороны сервера (этот объект обычно выделен или имеет фокус). Для некоторых программ можно указать принимающий элемент явно. В частности, для программы *Excel*можно указать номер ячейки в формате ≪RnCm≫. Здесь где п — номер строки, начиная с 1, a m — номер столбца, начиная с 1. Например, ячейка с координатами (5,12) обозначается как R5C12.

Второй параметр — передаваемая строка в формате PChar.

Сделаем так, чтобы по щелчку на кнопке *Послать* с помощью этого метода в ячейку электронной таблицы с координатами (2,3) (программа Excel должна быть запущена заранее) записывалось содержимое поля Editl нашей программы

**procedure**TForml.ButtonlClick(Sender: TObjecti;

var ToExcel: array [0 .. 100] of char;

begin

StrPCopy(ToExcel, Editl.Text);

DDEClientConvl.PokeDatal 'R2C3' , ToExcel) ,-

end;

Чтобы принять информацию от программы-сервера, надо воспользоваться следующим методом.

function RequestData(const Item: string]: PChar;

Единственный параметр — объект сервера, содержащий принимаемые данные (в нашем случае — ячейка таблицы).

Функция возвращает в формате PChar указатель на строку, которая хранит данные, полученные от сервера.

Для отображения в поле Editl программы содержимого ячейки (1,1) надо сформировав следующий обработчик щелчка на кнопке *Принять*.

procedure TForml.Button2ClickfSender: TObject);

var FromExcel: PChar;

begin

FromExcel := DDEClientConvl.ReguestData('R1C1') ;

Editl.Text := StrPas(FromExcel);

end;

Как видим, использование технологии *DDE* не вызывает особых сложностей.

С помощью компонентов TDdeServerConv и TOdeServerltem можно расширить возможности своего приложения, чтобы оно также было способно работать как сервер DDE. К недостаткам технологии DDE надо отнести отсутствие единого подхода к работе с различными серверами DDE.