

Вы можете включить любой тип данных как RCDATA или пользовательский тип ресурса. Это очень просто. Данный совет покажет вам общую технику создания такого ресурса.

```
    Type  TStrItem = String[39];    { 39 символов + байт длины -> 40 байтов }
TdataArray = Array [0..7, 0..24] of TStrItem;  Const  Data: TdataArray = ( ('.', .., '.'), { 25
строк на строку } ... { 8 таких строк } ('.', .., '.')); { 25 строк на строку }
```

Данные размещаются в вашем сегменте данных и занимают в нем 8К. Если это слишком много для вашего приложения, поместите реальные данные в ресурс

RCDATA

. Следующие шаги демонстрируют данный подход. Создайте небольшую безоконную программку, объявляющую типизированную константу как показано выше, и запишите результат в файл на локальный диск:

```
    program MakeData; type  TStrItem = string[39];    { 39 символов + байт длины -> 40
байтов }  TdataArray = array[0..7, 0..24] of TStrItem;  const  Data: TdataArray = ( ('.',
.., '.'), { 25 строк на строку } ... { 8 таких строк } ('.', .., '.')); { 25 строк на строку }
var  F: file of TdataArray; begin  Assign(F, 'data.dat');  Rewrite(F);  Write(F, Data);
Close(F); end.
```

Теперь подготовьте файл ресурса и назовите его *DATA.RC*. Он должен содержать только следующую строчку:

```
DATAARRAY RCDATA "data.dat"
```

Сохраните это, откройте сессию DOS, перейдите в каталог где вы сохранили data.rc (там же, где и data.dat!) и выполните следующую команду:

```
brcc data.rc (brcc32 для Delphi 2.0)
```

Теперь вы имеете файл data.res, который можете подключить к своему Delphi-проекту. Во время выполнения приложения вы можете генерировать указатель на данные этого ресурса и иметь к ним доступ, что и требовалось.

```
    { в секции interface модуля } type  TStrItem = string[39];    { 39 символов + байт
длины -> 40 байтов }  TdataArray = array[0..7, 0..24] of TStrItem;  PdataArray =
^TdataArray; const  pData: PdataArray = nil; { в Delphi 2.0 используем Var }
implementation {$R DATA.RES}  procedure LoadDataResource; var  dHandle: THandle;
begin  { pData := Nil; если pData - Var }  dHandle := FindResource(hInstance,
'DATAARRAY', RT_RCDATA);  if dHandle 0 then  begin  dhandle :=
LoadResource(hInstance, dHandle);  if dHandle 0 then  pData :=
LockResource(dHandle);  end;  if pData = nil then  { неудача, получаем сообщение об
ошибке с помощью WinProcs.MessageBox, без помощи VCL, поскольку здесь код
выполняется как часть инициализации программы и VCL  возможно еще не
```

инициализирован! } end; initialization LoadDataResource; end.

Теперь вы можете ссылаться на элементы массива с помощью синтаксиса `pData^[i,j]`.

Источник: DelphiWorld