Листер плагин на Borland C++ Builder 6 для начинающих

Автор: Evgeniy Savich

Не боги горшки обжигают…

Наверное, данная статья является не учебным пособием, а попыткой обобщить опыт, полученный автором в процессе разработки плагина xBaseView на Delphi 7, когда пришлось столкнуться с про-блемами, довольно неприятными программисту, привыкшему мощной поддержке VCL среды.

Задача: создать плагин для просмотра RTF файлов.

Создаем проект DLL библиотеки и сохраняем его под именем ListSimpleBcb.bpr в отдельной папке, изменим расширение имени плагина на WLX в опциях проекта.

Модуль Unit1.cpp сохраняем под именем ListSimple.cpp.

Плагин должен экспортировать из библиотеки три функции:

```
ListGetDetectString,
ListLoad,
ListCloseWindow.
```

ListGetDetectString должна записать в параметр DetectString строку, которая содержит RTF.

ListLoad вызывает плагин для работы и передает параметры:

```
ListerWin – дескриптор окна Листера;
FileToLoad – полное имя RTF файла.
```

Плагин должен создать свое окно в качестве дочерней по отношению к окну Листера и вернуть дескриптор этого окна. Эту инициализацию мы будем осуществлять в функции ShowRTF:

```
HWND ShowRTF(HWND ListerWin, char* FileToLoad);
```

ListCloseWindow требует от плагина завершения работы, передавая дескриптор окна плагина. Это освобождение ресурсов мы будем делать в функции HideRTF:

```
void HideRTF(HWND PluginWin);
```

Надо добавить в проект форму, изменить ее имя на fmMain и сохранить модуль формы под именем unMain.cpp. Эта форма и есть окно плагина.

В заголовочный файл unMain.h надо добавить определения двух вышеприведенных функции и уда-лить бесполезную глобальную переменную:

```
extern PACKAGE TfmMain *fmMain;
```

Замечание: ПЛАГИН НЕ ДОЛЖЕН ИСПОЛЬЗОВАТЬ ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ!

Бросив на форму компонент RichEdit, надо установить ему такие свойства:

```
Align = alClient,
ReadOnly = True,
ScrollBars = ssBoth.
```

Объект RichEdit1 обеспечивает всю работу с RTF файлом.

В плагинах часто используется контекстное меню, поэтому надо бросить на форму компонент PopupMenu и создать два пункта меню Help и About, назначив им клавиши F1 и F2, соответственно. Необходимо прикрепить контекстное меню объекту RichEdit1, установив свойство RichEdit1:

```
PopupMenu = PopupMenu1.
```

Окно плагина является дочерним, следовательно, оно не должно иметь рамку и заголовок. Для этого надо переопределить виртуальную функцию CreateParams класса TCustomForm. Здесь, мы кроме настройки стилей окна:

```
Params.Style = WS_CHILD | WS_MAXIMIZE & !WS_CAPTION & !WS_BORDER;
```

зарезервируем память для хранения указателя на нашу форму:

```
Params.WindowClass.cbWndExtra = sizeof(void *);
```

Указатель нужен функции HideRTF для корректного закрытия плагина. В функции ShowRTF мы сохраняем указатель в WinAPI структуре окна плагина следующим вызовом:

```
SetWindowLong(fmMain->Handle, GWL_USERDATA, (LONG)p);
```

Позднее мы вытащим указатель из структуры, используя дескриптор нашего окна, который является параметром HideRTF, вызывая функцию GetWindowLong:

```
GetWindowLong(PluginWin, GWL_USERDATA);
```

Чтобы создать дочернее окно воспользуемся статической функцией CreateParentedControl класса TWinControl, т.е. в теле функции функция ShowRTF воспользуемся следующим оператором:

```
fmMain = (TfmMain *)(TWinControl::CreateParentedControl(__classid(TfmMain),
ListerWin));
```

Для работы плагина нам понадобятся следующие данные: дескрипторы окон Тотал Командира и Листера, и режим работы плагина: в обычном окне или на одной из двух панелей Тотал Командира (так называемый Quick View - режим быстрого просмотра). Дескрипторы нужны, чтобы избежать краха по клавишам выхода Alt+X и для реагирования на быстрые клавиши Листера. Эти данные будем хранить, соответственно, в переменных TotCmdWin, ParentWin и QuickView. Добавим их определения в public раздел класса формы в файле unMain.h. Дескриптор окна Тотал Командира мы можем найти по имени класса этого окна (это не VCL класс!) с помощью WinAPI функции FindWindow.

Немного теорий. Особенностью библиотеки визуальных компонентов Borland VCL является ис-пользование глобального объекта "приложение" - Application. Этот Application имеет скрытое окно, которое должно быть владельцем всех окон приложения,

что обеспечивает корректное поведение всех окон программы. И DLL библиотека плагина, и исполняемый EXE файл Тотал Командира имеют собственный глобальный объект Application, следовательно, требуется их синхронизация. Для этого надо присвоить дескриптор скрытого окна объекта Application исполняемого файла к дескриптору объекта Application библиотеки. К сожалению, Тотал Командир, хотя он написан на Delphi и использует все удобства и прелести VCL, не передает нам дескриптор данного скрытого окна. (Может причиной этого является проблема версии компиляторов Delphi?). За неимением оно-го, вместо него будем использовать для синхронизации дескриптор окна Листера, в противном слу-чае, наше модальное окно (например, окно About) поведет себя неестественно, оно появится на панели задач Windows. Эта синхронизация является задачей для функции ShowRTF.

Чтобы защитить, Тотал Командир от сбоев плагина, будем использовать событие OnException объ-екта Application, которое перехватывает необработанное исключение. Кинем на форму компонент ApplicationEvents, переименуем его на App и создаем обработчик события OnException, куда введем вызов функции MessageBox для выдачи сообщения о сбое. Сохранив проект и модуль, можно уда-лить из формы ненужный компонент ApplicationEvents. Установка адреса обработчика события в Application.OnException также является задачей для функции ShowRTF.

Все готово для ShowRTF? Увы, "маленькая неточность" в Plugin API Тотал Командира (включая v.6.02) значительно усложняет нашу работу:

Если пользователь откроет окно плагина, затем переключится на Тотал Командир и закроет Тотал Командир, то мы не получим вызов через ListCloseWindow для завершения своей работы. Вместо этого Тотал Командир или Листер (кто его знает?) уничтожает наше окно вызовом WinAPI функции DestroyWindow, а потом, то же самое попытается сделать наш объект Application, в конечном счете плагин вылетает с нарушением защиты памяти (General Protection Fault)!

Если мы не будем делать вышеописанную синхронизацию, этой проблемы удалось бы избежать. Что ж, будем расплачиваться за это и поставим ловушку для перехвата сообщений, которое получа-ет наше же окно, подобно змее, которая пожирает себя за свой собственный хвост! Не будем вда-ваться в подробности Windows API (кому же он нравиться?), вкратце суть дела такова.

Объявляем структуру TPlugInfo, где будем хранить данные, которые нужны для закрытия плагина. При инициализации выделяем память под эту структуру и заполняем ее. Определяем функцию HookDestroy, который будет перехватывать оконные сообщения, чтобы среагировать на сообщение WM_DESTROY (уничтожение окна). Вызовом SetWindowLong(…, GWL_WNDPROC, …) подменя-ем стандартный обработчик оконных сообщений на функцию HookDestroy. Похожим вызовом SetWindowLong в функции завершения HideRTF обратно восстанавливаем прежний обработчик.

Важно то, что когда функция HookDestroy поймает сообщение WM_DESTROY, она вызывает функцию завершения HideRTF точно так же, как это делает функция ListCloseWindow. В результате мы всегда успеваем нормально закрыть окно и убрать за собой, что, к примеру демонстрирует free(p) из HideRTF, освобождая память, выделенную в ShowRTF.

Остается создать диалоговую форму About, обработчики контекстного меню и обработчик RichEdit1KeyDown, который ловит нажатие специальных клавиш и передает их с

помощью PostMessage в соответствующее окно. Если здесь не перехватывать клавиши Alt+X, это приведет к краху плагина. Причем, здесь придется удерживать фокус ввода в RichEdit1, проверяя режим рабо-ты плагина, т.е. значение переменной QuickView.

Об отладке и тестировании плагина.

Листер имеет встроенную возможность просмотра RTF файлов, поэтому перед запуском плагина надо отключить ее, сняв пометку у флага RTF в окне Configure Lister.

Чтобы использовать интегрированный отладчик IDE BCB, закройте Total Commander, а в BCB через меню "Run/Parameters" откройте диалог и в поле Host Application введите путь к Total Commander с помощью кнопки [Browse]. Теперь можно запускать плагин из-под BCB.

Ссылка для скачивания архива с примером здесь

PS. Уважаемый Читатель! Программирование - многовариантно и я никак не претендую на абсо-лютную истину. В качестве дополнительного материала, предлагаю Вам ознакомиться с исходными текстами плагина xBaseView - "Просмотр DBF, DB, MDB, ADO и ODBC баз данных", который предоставляется с исходными текстами, и где применяется аналогичный подход.

PPS. Прошу Читателя-Полиглота помочь мне перевести статью на английский язык, чтобы послать ее Кристиану Гислеру, автору знаменитого файл менеджера Тотал Командир, с призрачной надеж-дой, что он предпримет какие-то шаги навстречу нам - VCL программистам.

Е. Савич
19.03.04

mutex@nm.ru