

Когда Вы пишете динамически загружаемые библиотеки и необходимо контролировать загрузкой и выгрузкой **DLL**, или если Вы хотите подготовить **DLL** при инициализации, у Вас может быть функция с именем

DllMain

, которая будет делать инициализацию. Windows вызывает функцию

DllMain

из

DLL

в четырех случаях:

1. Когда процесс присоединяет DLL
2. Когда поток присоединяет DLL
3. Когда процесс отсоединяет DLL
4. Когда нить отсоединяет DLL

Каждая DLL имеет точку входа. Эта точка входа реализована как функция обратного вызова. Она вызывается системой, когда событие происходит. Функция имеет имя по умолчанию *DllMain*. DLL, которая не должна быть загружена по определению, может просто возвращать **False** в функции *DllMain()* как только она обнаруживает, кто вызывает ее.

В Delphi, *DLLProc* используется для определения процедуры, которая вызывается каждый раз, когда вызывается точка входа библиотеки DLL. Процедура назначенная на *DLLProc* получает один параметр целое число (*Reason*).

Функция API *GetModuleFileName()* возвращает название вызывающего модуля, если Вы передаете 0 как его первый аргумент. Этот параметр - дескриптор модуля, имя которого Вы хотите знать.

Когда параметр *Reason* - **DLL_PROCESS_ATTACH**, устанавливая *ExitCode* в ненулевое значение, заставляет точку входа возвращать

False

.

Вот пример кода, который позволяет только MyCallingApp.exe загружать библиотеку (Примечание:

у DLL в этом примере нет НИКАКОГО кода кроме процедуры DLLMain):

```
library OnlyMyDLL; uses SysUtils, Windows; procedure DllMain(reason: integer) ;
var buf : array[0..MAX_PATH] of char; loader : string; begin case reason of
DLL_PROCESS_ATTACH: begin GetModuleFileName(0, buf, SizeOf(buf)) ;
loader := buf; if Pos('MyCallingApp.exe', loader) > 0 then ExitCode := -1 end;
DLL_PROCESS_DETACH begin // DLL выгружается... end; end; end;
(*DllMain*) begin DllProc := @DllMain; DllProc(DLL_PROCESS_ATTACH) ; //
другой код для этой DLL... end. pre
```