



LoadLibrary(...)

LoadLibrary maps the specified executable module into the address space of the calling process.

Prototype

HPINSTANCE

```
LoadLibrary( LPCTSTR lpLibFileName );
```

Parameters

lpLibFileName

Pointer to a null-terminated string that names the executable module RTDLL.

A path is not necessary to provide, but the RTDLL should be registered.

Return Values

If the function succeeds, the return value is a handle to the module.

If the function fails, the return value is **NULL**. To get extended error information, call `GetLastError`.

DLLs **must be** registered through "RTSSRun /dll <image name>"



FreeLibrary(...)

FreeLibrary decrements the reference count of the loaded dynamic-link library (DLL) module. When the reference count reaches zero, the module is unmapped from the address space of the calling process and the handle is no longer valid.

Prototype

BOOL

```
FreeLibrary(HMODULE hLibModule);
```

Parameters

hLibModule
function

Handle to the loaded library module. The `LoadLibrary` function returns this handle.

Return Values

If the function succeeds, the return value is `TRUE`.

If the function fails, the return value is `FALSE`. To get extended error information, call `GetLastError`.



GetProcAddress(...)

GetProcAddress returns the address of the specified exported dynamic-link library function.

Prototype

```
FARPROC GetProcAddress(HMODULE hModule, LPCSTR lpProcName);
```

Parameters

<i>hModule</i>	A handle to the DLL module that contains the function. The <code>LoadLibrary</code> function returns this handle.
<i>lpProcName</i>	A pointer to a null-terminated string containing the function name, or the function's ordinal value.

Return Values

If the function succeeds, the return value is the address of the DLL's exported function.

If the function fails, the return value is **NULL**. To get extended error information, call `GetLastError`.



MRTS – RTX6.5



```
FARPROC FunctionPtr = NULL;
void main () {
HANDLE hLibModule = NULL;
LPCSTR lpLibFileName = "VXI821.rtdll";
LPCSTR lpProcName = "_Trigger@2";
int ret;

/* load Library */
hLibModule = LoadLibrary(lpLibFileName);

/* check if loadLibrary returned correctly */
if(hLibModule== NULL) {
    /* ERROR */
    return;
}

/* Get function from Rtdll */
FunctionPtr = GetProcAddress(hLibModule, lpProcName);

/* check if function was found */
if(FunctionPtr == NULL) {
    /* ERROR */
    FreeLibrary(hLibModule);
    return;
}

/* Call function */
ret = FunctionPtr(0, NULL);

/* Free Library */
FreeLibrary(hLibModule);
}
```