
Subject: U++ core app as DLL: memory manager problems?

Posted by [Mindtraveller](#) on Tue, 22 Sep 2009 22:19:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'd like to use U++ as a platform for developing complex dynamic libraries for current work project. This dynamic library will be crossplatform (Win/POSIX) and will be attached to a number of applications written in U++ (Win, POSIX) and Delphi (Win).

Provided I export functions with POD parameters, should I expect any run-time casualties/problems with U++ memory manager?

Subject: Re: U++ core app as DLL: memory manager problems?

Posted by [mirek](#) on Thu, 24 Sep 2009 07:51:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Tue, 22 September 2009 18:19: I'd like to use U++ as a platform for developing complex dynamic libraries for current work project.

This dynamic library will be crossplatform (Win/POSIX) and will be attached to a number of applications written in U++ (Win, POSIX) and Delphi (Win).

Provided I export functions with POD parameters, should I expect any run-time casualties/problems with U++ memory manager?

As long as the interface does not expect that client code directly allocates or frees DLLs heap, everything should be OK.

In fact, at least in Win32, most DLLs I have seen (including good old OLE interfaces) are designed in a way that specifically avoid this problem.

Example: this interface would make things go mad:

Client code:

```
DIIX *x = DIICreateX();
```

```
....
```

```
delete x;
```

and this is correct:

```
DIIX *x = DIICreateX();
```

```
....
```

```
DIIDeleteX(x);
```

or

```
DllX *x = DllCreateX();  
....  
x->Delete();
```

Mirek
