
Subject: small dli enhancement

Posted by [Shire](#) on Sun, 14 Apr 2013 04:17:33 GMT

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

dli.h generates two global functions:

DLIMODULE() and DLIMODULE ## _()

First function has load .dll before return module pointer. Second function is just holder of static module variable.

There are many situations where additional module check is not necessary - for example, sequence of dll function calls, and DLIMODULE_() function is good for these situations. But it generates in source section, and compiler have to generate additional function call for each dll function call. I propose to move DLIMODULE_() to header section and make it inline:

```
DLLTYPE& DLIMODULE();  
-DLLTYPE& COMBINE(DLIMODULE, _());  
+inline DLLTYPE& COMBINE(DLIMODULE, _())  
+{  
+ static DLLTYPE out;  
+ return out;  
+}
```

```
#endif
```

```
#ifdef DLI_SOURCE  
#undef DLI_SOURCE
```

```
-DLLTYPE& COMBINE(DLIMODULE, _())  
- {  
- static DLLTYPE out;  
- return out;  
- }
```

This can avoid overhead for function call at every dll function invoke.
