Subject: WARNINGS: LOG -> empty statement in RELEASE
Posted by kohait00 on Wed, 30 Mar 2011 09:09:19 GMT

View Forum Message <> Reply to Message

hi all, LOG yields some warning in if statements when RELEASE.
change to:

```
diff --git a/bazaar/Urr/Urr.h b/bazaar/Urr/Urr.h
index 3bec7c3..3ca50ac 100644
--- a/bazaar/Urr/Urr.h
+++ b/bazaar/Urr/Urr.h
@@ -32,7 +32,7 @@ inline int b_sendto(
 ) {
  Sleep(rand() % 10);
  int bug = rand() % 10;
- if(bug == 0) LOG("[Emulated drop]");
+ if(bug == 0) { LOG("[Emulated drop]"); }
  return bug ? sendto(s, buf, len, flags, to, tolen) : 0;
 }

diff --git a/uppsrc/Core/Debug.cpp b/uppsrc/Core/Debug.cpp
index fe4b239..9b469e9 100644
--- a/uppsrc/Core/Debug.cpp
+++ b/uppsrc/Core/Debug.cpp
@@ -230,7 +230,7 @@ void DeleteUsrLog()
  }
  else {
   if(susrlog && !UsrLogStream().Delete())
-   LOG("Unable to delete UsrLog, " << GetLastErrorMessage());
+  { LOG("Unable to delete UsrLog, " << GetLastErrorMessage()); }
   susrlog = false;
  }
 }
```

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by dolik.rce on Wed, 30 Mar 2011 11:14:48 GMT

View Forum Message <> Reply to Message

Hi kohait,

There is lot more warnings like this all over U++ IMHO better solution (more general and future-proof) would be defining LOG as nonempty macro:

```
#ifndef flagDEBUG
#define LOG(X) {}
#endif
```

I don't recall where the LOG is originally defined, but I'm sure you can find it

Honza

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by kohait00 on Wed, 30 Mar 2011 12:01:31 GMT
View Forum Message <> Reply to Message

nice good idea

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by mirek on Wed, 30 Mar 2011 19:43:48 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Wed, 30 March 2011 07:14Hi kohait,

There is lot more warnings like this all over U++ IMHO better solution (more general and future-proof) would be defining LOG as nonempty macro:

```
#ifndef flagDEBUG
#define LOG(X) {}
#endif
```

I don't recall where the LOG is originally defined, but I'm sure you can find it

Honza

Are you sure that GCC guys will not warn about "empty block" someday?

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by dolik.rce on Wed, 30 Mar 2011 20:18:40 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 30 March 2011 21:43Are you sure that GCC guys will not warn about "empty block" someday?
If they do, we can make it "#define LOG(X) {int i=0;}", so later they can start warning us about "useless code block" Then I would really start to consider using -Wno-empty-body (or -Wno-useless-code).

Honza

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by mirek on Thu, 31 Mar 2011 07:29:26 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Wed, 30 March 2011 16:18mirek wrote on Wed, 30 March 2011 21:43Are you sure that GCC guys will not warn about "empty block" someday?
If they do, we can make it "#define LOG(X) {int i=0;}", so later they can start warning us about "useless code block" Then I would really start to consider using -Wno-empty-body (or -Wno-useless-code).

Honza

This one would yield "unused variable" even today...

I guess the one possible way around would be:

```
inline void Nop() {}

#define LOG(x) Nop()
```

Mirek

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by kohait00 on Thu, 31 Mar 2011 08:25:36 GMT
View Forum Message <> Reply to Message

tested: works and produces exact same executable in OPTIMAL.

```
diff --git a/uppsrc/Core/Defs.h b/uppsrc/Core/Defs.h
index 3f2bbca..a299889 100644
--- a/uppsrc/Core/Defs.h
+++ b/uppsrc/Core/Defs.h
@@ -16,6 +16,8 @@
  #pragma setlocale("C")
 #endif

+inline void Nop() {}
+
 bool    IsPanicMode();

 void    Panic(const char *msg);
@@ -34,8 +36,8 @@ void    PanicMessageBox(const char *title, const char *text);

 #else
```

```
-#define ASSERT_(x, msg)
-#define ASSERT(x)
+#define ASSERT_(x, msg) UPP::Nop()
+#define ASSERT(x) UPP::Nop()

 #endif

diff --git a/uppsrc/Core/Diag.h b/uppsrc/Core/Diag.h
index 0c22914..93a43b5 100644
--- a/uppsrc/Core/Diag.h
+++ b/uppsrc/Core/Diag.h
@@ -135,35 +135,35 @@ inline void UnlockLog() {}
 #define DDUMPM(x)       @
 #define DTIMING(x)      @

-#define ASSERT(x)
+#define ASSERT(x) UPP::Nop()

-#define DEBUGCODE(x)
+#define DEBUGCODE(x) UPP::Nop()

 inline void LOGF(const char *format, ...) {}

-#define LOG(a)
-#define LOGBEGIN()
-#define LOGEND()
-#define LOGBLOCK(n)
-#define LOGHEXDUMP(s, a)
+#define LOG(a) UPP::Nop()
+#define LOGBEGIN() UPP::Nop()
+#define LOGEND() UPP::Nop()
+#define LOGBLOCK(n) UPP::Nop()
+#define LOGHEXDUMP(s, a) UPP::Nop()
 #define QUOTE(a)        a
-#define LOGSRCPOS()
-#define DUMP(a)
-#define DUMPC(a)
-#define DUMPCC(a)
-#define DUMPCCC(a)
-#define DUMPM(a)
-#define XASSERT(c, d)
-#define NEVER()
-#define NEVER_(msg)
-#define XNEVER(d)
+#define LOGSRCPOS() UPP::Nop()
+#define DUMP(a) UPP::Nop()
+#define DUMPC(a) UPP::Nop()
```

```
+#define DUMPCC(a) UPP::Nop()
+#define DUMPCCC(a) UPP::Nop()
+#define DUMPM(a) UPP::Nop()
+#define XASSERT(c, d) UPP::Nop()
+#define NEVER() UPP::Nop()
+#define NEVER_(msg) UPP::Nop()
+#define XNEVER(d) UPP::Nop()
 #define CHECK(c)         (c)
 #define XCHECK(c, d)     (c)

-#define TIMING(x)
-#define HITCOUNT(x)
-#define ACTIVATE_TIMING()
-#define DEACTIVATE_TIMING()
+#define TIMING(x) UPP::Nop()
+#define HITCOUNT(x) UPP::Nop()
+#define ACTIVATE_TIMING() UPP::Nop()
+#define DEACTIVATE_TIMING() UPP::Nop()

 #endif
```

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by kohait00 on Tue, 05 Apr 2011 18:47:49 GMT
View Forum Message <> Reply to Message

gcc release: still mourns

```
C:\uppsvn\uppsrc/Core/Diag.h:142:0: warning: "ASSERT" redefined
C:\uppsvn\uppsrc/Core/Defs.h:38:0: note: this is the location of the previous definition
```

Subject: Re: WARNINGS: LOG -> empty statement in RELEASE
Posted by mirek on Thu, 07 Apr 2011 08:52:41 GMT
View Forum Message <> Reply to Message

Should be fixed now...