Subject: Store empty string to database
Posted by dolik.rce on Sat, 20 Oct 2012 14:29:07 GMT
View Forum Message <> Reply to Message

Hi everyone!

Is it possible to differentiate between empty string and NULL in a database? As far as I can tell, storing "", e.g. in Insert(), always results to NULL being inserted. I believe it is because in U++ empty string is considered to be Null.

Is there some workaround to this? I can easily imagine cases where one needs to differentiate between NULL and "" in databases. So far I only came up with ugly solution of forcing the literal value of """" into SqlVal instead of "", but that is not very nice solution (and by that I mean it is basically a hack).

Also, the same problem might arise in the Core itself. That is actually the reason why there is String::IsVoid(). So, wouldn't it be more logical to store Null in String as the "void state"? Or at least behave like this in the Sql layer, but then it would be kind of inconsistent behavior which is not very good either...

Best regards,
Honza

Subject: Re: Store empty string to database
Posted by dolik.rce on Sat, 20 Oct 2012 15:55:06 GMT
View Forum Message <> Reply to Message

Out of curiosity, I tried to alter String(const Nuller&) and IsNull(const String&) and compile TheIDE. It compiles well (not surprising) and it even mostly runs (very surprising). There are some artifacts caused by improper use of not-any-more-interchangeable calls to IsEmpty and IsNull, but most of the application somehow works. The most broken parts of U++ (with this patch) are not surprisingly CParser, CodeEditor and other classes heavily dealing with strings. I was expecting it to be much worse Maybe there is hope to make the switch and start using void as null

Honza

PS: The patch (for other curios people) :

```
--- upp-production/uppsrc/Core/String.h (revision 5441)
+++ upp-production/uppsrc/Core/String.h (working copy)
@@ -341,7 +341,7 @@
        int    GetCharCount() const;

        String()                                { Zero(); }
-       String(const Nuller&)                   { Zero(); }
+       String(const Nuller&)                   { ptr = (char *)(voidptr + 1); LLen() = 0; SLen() =
```

```
15; chr[KIND] = 50; }
       String(const String& s)                    { String0::Set(s); }
       String(const char *s);
       String(const String& s, int n)              { ASSERT(n >= 0 && n <= s.GetLength());
String0::Set(~s, n); }
@@ -495,7 +495,7 @@
 }

 template<>
-inline bool  IsNull(const String& s)       { return s.IsEmpty(); }
+inline bool  IsNull(const String& s)       { return s.IsVoid(); }

 template<>
 inline String AsString(const String& s)     { return s; }
```

Subject: Re: Store empty string to database
Posted by mirek on Sat, 20 Oct 2012 16:17:37 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 20 October 2012 10:29Hi everyone!

Is it possible to differentiate between empty string and NULL in a database? As far as I can tell, storing "", e.g. in Insert(), always results to NULL being inserted. I believe it is because in U++ empty string is considered to be Null.

Correct. This is long running issue. Originaly, U++ inherited this from Oracle. In around 2001, we have shortly experimented with !IsNull(String()), but we have found this causing even more problem (e.g. for integer values, it is logical that "empty" EditInt yields Null. So it is sort of expected that empty EditString is Null). So we have reverted and restored IsNull(String()) behaviour.

Quote:
Is there some workaround to this? I can easily imagine cases where one needs to differentiate between NULL and "" in databases.

At this moment, no. In the past, we tried String::GetZero() as special value representing empty string in DB... Never got actually used and supported in Sql, so was removed. If we are about to restore something like this, I think it should go along this line (e.g. SqlEmptyString() and IsSqlEmptyString()).

Mirek

Mirek

Subject: Re: Store empty string to database
Posted by dolik.rce on Sat, 20 Oct 2012 20:51:38 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 20 October 2012 18:17Quote:
Is there some workaround to this? I can easily imagine cases where one needs to differentiate between NULL and "" in databases.

At this moment, no. In the past, we tried String::GetZero() as special value representing empty string in DB... Never got actually used and supported in Sql, so was removed. If we are about to restore something like this, I think it should go along this line (e.g. SqlEmptyString() and IsSqlEmptyString()).

Mirek

Mirek
Ok, for now I will use this: SqlVal SqlEmptyString(){
 static SqlVal s("""",SqlS::HIGH);
 return s;
}
// which can be used as
sql * Insert(TBL)(SOMECOL, str.IsEmpty()?SqlEmptyString():str)In the end, it is quite simple and not as ugly as I initially expected

Honza