Subject: Re: The problem with 'Null'
Posted by gridem on Sun, 22 Mar 2009 08:15:57 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 20 March 2009 11:52

Two notes:

You cannot really define Null for bool, as it has only 2 values..

It is more correct to use "IsNull" instead of "== Null".

Oh, my mistake: IsNull of course instead of == Null.

luzr wrote on Fri, 20 March 2009 11:52

Well, I can see where you are heading, but I do not really like that path

My only apology at this moment is that U++ is "practice driven", and in the whole history (which now spans abuot 10 years), we never missing IsNull for containers...

BTW, as you have noticed, there is the small issue with String Null - empty string is considered Null.

I agree this is sort of controversial decision. Indeed, a couple of years ago, we identified it as mistake and tried

IsNull(String()) == false
IsNull(String(Null)) == true
String(Null) == String()

variant. Well, what happened is that in practice, this was found to be rather unfortunate. I guess the primary problem is that it is very convenient and natural, when working with databases and GUI, that all empty String gui fields are inserted as Nulls. With above, you would need to have additional GUI buttons to say whether the field is empty or whether it is null.

Similar issues can be found across the code. That is why we went back to

IsNull(String()) == true

As a sidenote, this equivalence was originally taken from Oracle.

Mirek

Thank you for your explanations. I caught the main idea. So for SQL programming it's possible to

introduce 'local IsNull':

```
template<typename T>
bool IsNullSql(const T& t)
{
  return IsNull(t);
}

bool IsNullSql(const String& s)
{
  return s.IsEmpty();
}
```