Subject: Strange exception raising with SQL.ExecuteX()
Posted by BioBytes on Thu, 09 Dec 2010 09:54:26 GMT
View Forum Message <> Reply to Message

Hi all,

The following code, which compiles correctly, raises an exception while the updating operations are correctly implemented in my database tables:

```
void qsSKillFormDlg::RefreshInfosProcOwner(String newValue,String owner,int fieldId)
{
 SqlId TBLOWN("OWNERS"),FIRSTNAME("OWNERFIRSTNAME"),FAMILYNAME(
"OWNERNAME");
 Vector<String> infos;

  try
   {
     infos.Clear();
  infos = Split(infosProcOwner.At(2),' ',false);

    switch(fieldId)
  {
    case 0: infosProcOwner.At(2).Clear();
            infosProcOwner.At(2)<<infos.At(0)<<' '<<newValue;
            SQL.ExecuteX("
Update(TBLOWN)(FAMILYNAME,newValue).Where(FAMILYNAME==owner) ");

            break;

    case 1:
SQL.ExecuteX("Update(TBLOWN)(FIRSTNAME,newValue).Where(FAMILYNAME==owner) ");
              break;
  }
   }
   catch(SqlExc& err)
   {
      PanicMessageBox(t_("Erreur SQL"),err);
   }
}
```

I use MySql Server 5.0 and U++ 2884. Of course, if I replace ExecuteX and no longer implement the exception Try/Catch, no error is shown in runtime.

Perhaps I misuse the SqlId syntax or is there a bug?. Is anyone has an idea ?

Thanks for suggestions

Regards

Subject: Re: Strange exception raising with SQL.ExecuteX()
Posted by mirek on Sun, 12 Dec 2010 08:39:35 GMT
View Forum Message <> Reply to Message

BioBytes wrote on Thu, 09 December 2010 04:54Hi all,

The following code, which compiles correctly, raises an exception while the updating operations are correctly implemented in my database tables:

```
void qsSKillFormDlg::RefreshInfosProcOwner(String newValue,String owner,int fieldId)
{
 SqlId TBLOWN("OWNERS"),FIRSTNAME("OWNERFIRSTNAME"),FAMILYNAME(
"OWNERNAME");
 Vector<String> infos;

  try
   {
     infos.Clear();
  infos = Split(infosProcOwner.At(2),' ',false);

    switch(fieldId)
  {
    case 0: infosProcOwner.At(2).Clear();
            infosProcOwner.At(2)<<infos.At(0)<<' '<<newValue;
            SQL.ExecuteX("
Update(TBLOWN)(FAMILYNAME,newValue).Where(FAMILYNAME==owner) ");

            break;

    case 1:
SQL.ExecuteX("Update(TBLOWN)(FIRSTNAME,newValue).Where(FAMILYNAME==owner) ");
              break;
  }
   }
   catch(SqlExc& err)
   {
      PanicMessageBox(t_("Erreur SQL"),err);
   }
}
```

I use MySql Server 5.0 and U++ 2884. Of course, if I replace ExecuteX and no longer implement the exception Try/Catch, no error is shown in runtime.

Perhaps I misuse the SqlId syntax or is there a bug?. Is anyone has an idea ?

Thanks for suggestions

Regards

Biobytes

What is SQL error? I mean, what is reported to be in 'err'?

Mirek

Subject: Re: Strange exception raising with SQL.ExecuteX()
Posted by BioBytes on Sun, 12 Dec 2010 20:28:33 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks for considering this question. Please find hereafter the Sql error (err) displayed in the PanicMessageBox:

[MySQL][ODBC 5.1 Driver][mysqld-5.0.45-community-nt]You have an error in your SQL syntax, check the manual that corresponds to your MySQL server version for he right syntax to use near '(FAMILYNAME,newValue).Where(FAMILYNAME==owner)'at line 1
SQL error:
Update(TBLOWN)(FAMILYNAME,newValue).Where(FAMILYNAME==owner)

I compile under MingW 5.16.

Hoping this could be helpful

Regards
Biobytes

Subject: Re: Strange exception raising with SQL.ExecuteX()
Posted by BioBytes on Tue, 21 Dec 2010 13:30:25 GMT
View Forum Message <> Reply to Message

Hi all,

I solved this problem by using SqlId syntax. The following code works perfectly and does not raise any unexpected exception :

```
void qsSKillFormDlg::RefreshInfosProcOwner(String newValue,String owner,int fieldId)
{
SqlId TBLOWN("OWNERS"),FIRSTNAME("OWNERFIRSTNAME"),FAMILYNAME(
```

```
"OWNERNAME");
Vector<String> infos;

try
{
infos.Clear();
infos = Split(infosProcOwner.At(2),' ',false);
infosProcOwner.At(2).Clear();

switch(fieldId)
{
case 0: infosProcOwner.At(2)<<infos.At(0)<<' '<<newValue;
SQL  &SqlUpdate(TBLOWN)(FAMILYNAME,newValue).Where(FAMILYNAME ==owner);

break;

case 1: infosProcOwner.At(2)<<newValue<<' '<<infos.At(1);
SQL  &SqlUpdate(TBLOWN)(FIRSTNAME,newValue).Where(FAMILYNAME= =owner);
break;
}
}
catch(SqlExc& err)
{
PanicMessageBox(t_("Erreur SQL"),err);
}
}
```

To avoid problems with the MySql engine, SqlId syntax must be used.

The problem should be considered closed.

Regards
Biobytes
