Subject: GridControl release memory
Posted by chis.adrian on Mon, 11 Dec 2017 06:56:43 GMT
View Forum Message <> Reply to Message

Hi,
I have the following function:


```
void SelSqliteQuery( GridCtrl * MyGridCtrl ,std::string Qstr)
 {

  MyGridCtrl->RemoveColumn(0,MyGridCtrl->GetColumnCount() );
  MyGridCtrl->Reset();
  MyGridCtrl->Ready(false);


 sqlite3 * db;

  if
(sqlite3_open_v2(path_db.ToString().ToStd().c_str(),&db,SQLITE_OPEN_READWRITE,NULL)!=
SQLITE_OK)
  {
   PromptOK("Can't create or open database file\n");
  return;
  }

   const char *quer = Qstr.c_str();
     sqlite3_stmt *statement;
 if(sqlite3_prepare_v2 (db, quer, -1, &statement, 0) != SQLITE_OK)
 {
  PromptOK("sqlite3_prepare_v2 FAILED \n");
  return;
 }

 int rows=0;
   int cols = sqlite3_column_count(statement);

 for (int CC=0;CC<cols;CC++)
 MyGridCtrl->AddColumn( (char*)sqlite3_column_name(statement, CC));

 while(true)
 {
  if( sqlite3_step(statement) == SQLITE_ROW)
  {
    for(int col = 0; col < cols; col++)
    MyGridCtrl->Set(rows,col, (char*)sqlite3_column_text(statement, col));
  rows++;
  } else break;
```

```
  }

  sqlite3_finalize(statement);
   sqlite3_close(db);
   MyGridCtrl->Ready(true);

  }
```

Well it work and its fast nice,
Problem is that this function is called many times as I want to create a small data base browser.
(I checked the function in other C++ app and for what I can tell ain't leaking)
Now when I select a big table memory goes up 400 mb acceptable its big but when I select a smaller one
memory remains same usage.
How can I reset ,delete ,clear whatever the memory usage.
Thank You.

Subject: Re: GridControl release memory
Posted by mirek on Mon, 11 Dec 2017 08:43:06 GMT
View Forum Message <> Reply to Message

If this is within single application run and if you deduce those 400MB from e.g. task manager, then the behaviour is as expected and it is a feature of memory allocator - it retains the memory allocated from the system to boost the performance.

I believe such behaviour is pretty standard for all malloc/new implementations, not U++ specific.

In U++, we have planned MemoryShrink function that is supposed to release as much memory as possible back to the system, but it is not really developed yet and pressure to do it is low. (The real reason why it is not yet developed is that Win32 has hard limit on number of memory commit blocks which is not quite compatible with what we need).

Subject: Re: GridControl release memory
Posted by chis.adrian on Mon, 11 Dec 2017 09:06:41 GMT
View Forum Message <> Reply to Message

Thank You,
I understand I was afraid of this answer since I saw on other C,C++ Grid,Table ... implementations.

Anyway is there a way to access Grids Vector<Vector (Exposed) so I can try shrink it my self?

I already done it in a C program and new C++ has std::vector::shrink_to_fit maybe I can use it.

Subject: Re: GridControl release memory
Posted by mirek on Mon, 11 Dec 2017 09:27:04 GMT
View Forum Message <> Reply to Message

chis.adrian wrote on Mon, 11 December 2017 10:06Thank You,
I understand I was afraid of this answer since I saw on other C,C++ Grid,Table ...
implementations.

Anyway is there a way to access Grids Vector<Vector (Exposed) so I can try shrink it my self?
I already done it in a C program and new C++ has std::vector::shrink_to_fit maybe I can use it.

Ah, you do not understand. That would not help. Memory is correctly released, there are no leaks
and nothing to be shrunk.

However, that released memory is kept in reserve by U++ memory management subsystem (aka
"memory allocator") for further allocations.

The reason is that a) retrieving memory from system is somewhat expensive b) U++ allocator
"formats" memory blocks, which is something that can be avoided if memory is kept in reserve
instead being returned to the system and then retrieved again

(and then there is that technical problem with Win32, which only allows limited granularity)

Subject: Re: GridControl release memory
Posted by chis.adrian on Mon, 11 Dec 2017 09:49:22 GMT
View Forum Message <> Reply to Message

OK.
I guess I'll have to live with that :d .
Not such a big problem application will be used by me only I can eventually
restart if it's required.
Thank you for quick answers.