

---

Subject: SQL refactored, adding per-thread option, added second SQLR  
Posted by [mirek](#) on Sat, 10 Dec 2011 16:10:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

SQL, 'global SQL session/cursor', is now refactored so that each thread is allowed to assign its own SqlSession to SQL.

Unless this is assigned, SQL defaults to 'global SQL', the first one that was assigned.

This way, it is possible to support both "single DB session per whole program" (only main thread assigns SQL) - in that case using SQL has to be serialized (mutexes). And it is also possible to have per-thread SQL (if thread assigns it).

I have also noticed that there is quite common pattern to use two SQL sessions - one for read/write master DB, second for read-only slaves. Read operations are then performed on slaves, which unloads master DB.

So support this, I have introduced secondary 'global cursor/session', SQLR. If not assigned, it defaults to session defined for SQL. Of course, global/per-thread behaviour is the same.

---

---

Subject: Re: SQL refactored, adding per-thread option, added second SQLR  
Posted by [mirek](#) on Tue, 28 Feb 2012 09:39:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

After gaining some experience, I have changed the semantics to be more explicit: You have to call Sql::PerThread or SqlSession::PerThread (or SQL.PerThread) to enable per-thread global sessions, otherwise there is just one global.

---

---

Subject: Re: SQL refactored, adding per-thread option, added second SQLR  
Posted by [BioBytes](#) on Tue, 28 Feb 2012 10:14:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi Mirek,

Are these changes implemented in the next release? I mean after the #4428.

Does it mean that we will use the following syntax ?

```
SQL = MySession;  
SQL.PerThread();
```

Thank you for continuously improving U++

Biobytes

---

---

Subject: Re: SQL refactored, adding per-thread option, added second SQLR

Posted by [mirek](#) on Tue, 28 Feb 2012 11:09:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

BioBytes wrote on Tue, 28 February 2012 05:14Hi Mirek,

Are these changes implemented in the next release? I mean after the #4428.

They are committed, so they will be in next nightly, and in next release.

Quote:

```
SQL = MySession;  
SQL.PerThread();
```

The syntax is valid, but it might not do what you expect. You need to create and assign a session in each thread again.

---

---

Subject: Re: SQL refactored, adding per-thread option, added second SQLR

Posted by [bushman](#) on Sun, 09 Dec 2012 23:10:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Tue, 28 February 2012 08:09

```
SQL = MySession;  
SQL.PerThread();
```

The syntax is valid, but it might not do what you expect. You need to create and assign a session in each thread again.

Is there an example on how to use it? If not, could you please briefly provide us with a few code lines on how to do it?

thanks!

---

---

Subject: Re: SQL refactored, adding per-thread option, added second SQLR  
Posted by [mirek](#) on Mon, 10 Dec 2012 04:24:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

kropniczki wrote on Sun, 09 December 2012 18:10mirek wrote on Tue, 28 February 2012 08:09

```
SQL = MySession;  
SQL.PerThread();
```

The syntax is valid, but it might not do what you expect. You need to create and assign a session in each thread again.

Is there an example on how to use it? If not, could you please briefly provide us with a few code lines on how to do it?

thanks!

It is used in e.g. Skylark. See [examples/AddressBookWeb](#).

Mirek

---