Subject: OkCommit problem
Posted by zsolt on Wed, 06 Sep 2006 16:53:02 GMT
View Forum Message <> Reply to Message

The problem is this method:

```
bool OkCommit(SqlSession& session, const char *msg) {
 if(ErrorRollback(session, msg))
  return false;
 session.Commit();
 return true;
}
```

In the case of unsuccessful commit, nothing happens. This way it is very hard to detect the problem, because the next execution generates an error.

So I changed the method to this:

```
bool OkCommit(SqlSession& session, const char *msg) {
 if(ErrorRollback(session, msg))
  return false;
 do{
  session.Commit();
  if(!session.WasError())
  {
   session.ClearError();
   return true;
  }
 }while(IDRETRY==SqlError(t_("Unable to commit changes to database."), session, true));
 return false;
}
```

Subject: Re: OkCommit problem
Posted by mirek on Wed, 06 Sep 2006 17:51:33 GMT
View Forum Message <> Reply to Message

Seems like a good idea. However, "Retry" is a bit strange IMO - if you are not able to commit, there is something pretty wrong with your database connection...

Maybe, in that case, application abort would be in place.

Mirek

Subject: Re: OkCommit problem
Posted by zsolt on Wed, 06 Sep 2006 18:37:40 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 06 September 2006 19:51Seems like a good idea. However, "Retry" is a bit strange IMO - if you are not able to commit, there is something pretty wrong with your database connection...

Maybe, in that case, application abort would be in place.

Mirek

Yes, you are right. I changed it to this:

```
bool OkCommit(SqlSession& session, const char *msg) {
 if(ErrorRollback(session, msg))
  return false;
 session.Commit();
 if(session.WasError())
  return false;
 session.ClearError();
 return true;
}
```

Application abort would be too drastic, I think, because this is one of the connections only. Programmer can use return value of OkCommit().

Subject: Re: OkCommit problem
Posted by mirek on Wed, 06 Sep 2006 18:52:22 GMT
View Forum Message <> Reply to Message

Well, this is a bit confused... ClearError when there is no error? Should not be there warning message and then ClearError in case of failure?

As for abort/not abort, for some time I am thinking about some form of option that would abort SQL app in this and other cases, well, basically, after any unsuccesful unrecovered (by rollback) DML command. Checking for return code after each command is tedious (I usually just call OkCommit at the end of processing and rarely check the result).

Mirek

Subject: Re: OkCommit problem
Posted by zsolt on Wed, 06 Sep 2006 19:53:26 GMT
View Forum Message <> Reply to Message

Yes maybe raising an Exception would be good here. What do you think about it?

Subject: Re: OkCommit problem
Posted by mirek on Sat, 09 Sep 2006 10:37:28 GMT
View Forum Message <> Reply to Message

OK, for now:

```
bool OkCommit(SqlSession& session, const char *msg) {
 if(ErrorRollback(session, msg))
  return false;
 session.Commit();
 if(ShowError(session, msg ? msg : t_("SQL error!")))
  return false;
 return true;
}
```

Raising exception is a good idea. But I wish to have that configurable... (return false/raise exception/abort).

Mirek

Subject: Re: OkCommit problem
Posted by zsolt on Mon, 11 Sep 2006 14:06:17 GMT
View Forum Message <> Reply to Message

Thanks, this should be good enough, I think.