

## Wt - Support #2174

### Trouble running tutorial5.C

09/04/2013 05:16 AM - Anonymous

<b>Status:</b>	Feedback	<b>Start date:</b>	09/04/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Koen Deforche	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			

#### Description

I get these errors when working on tutorial5.C code:

```
Error 4 error C2766: explicit specialization; 'Wt::Dbo::dbo_traits<User>' has already been defined c:\users\plug\documents\visual studio 2010\projects\st\stapp\main\source\model\user.h 25
```

...

```
Error 1 error C2908: explicit specialization; 'Wt::Dbo::dbo_traits<C>' has already been instantiated c:\users\plug\documents\visual studio 2010\projects\st\stapp\main\source\model\user.h 21
```

They occur when I change this code in my User.h file:

```
namespace Wt {
    namespace Dbo {

    }
}
```

To this here:

```
namespace Wt {
    namespace Dbo {

        template<>
        struct dbo_traits<User> : public dbo_default_traits {

            static const char *surrogateIdField() { return 0; }
        };

    }
}
```

This doesn't make sense to me since the definition doesn't exist anywhere in my solution or include subfolders, as shown by these search results:

```
Find all "struct dbo_traits<User>", Subfolders, Find Results 1, "Visual C++ Include Directories"
Matching lines: 0 Matching files: 0 Total files searched: 2668
```

...

```
Find all "struct dbo_traits<User>", Subfolders, Find Results 1, "Entire Solution"
C:\Users\user\Documents\Visual Studio 2010\Projects\app\main\source\model\User.h(21): struct
dbo_traits<User> : public dbo_default_traits {
Matching lines: 1 Matching files: 1 Total files searched: 30
```

It is only there once so how is it being defined and instantiated more than one time? These are the includes in my app.cpp which contains main():

```
...
#include <Wt/WAnchor>
#include <Wt/WApplication>
#include <Wt/WContainerWidget>
#include <Wt/WHBoxLayout>
#include <Wt/WCssStyleSheet>
```

```

#include <Wt/WLink>
#include <Wt/WText>
#include <Wt/WColor>
#include <Wt/WServer>
#include <Wt/Dbo/backend/Sqlite3>

#include "model/User.h"
#include "model/Post.h"
#include "model/Tag.h"
#include "model/Settings.h"

using namespace Wt;
using namespace std;

class User;
typedef Wt::Auth::Dbo::AuthInfo<User> AuthInfo;
...

```

And this from User.h:

```

...
#ifndef USER_
#define USER_

#include <Wt/WGlobal>
#include <Wt/Dbo/Dbo>
#include <string>

#include "Post.h"
#include "Settings.h"

namespace dbo = Wt::Dbo;

namespace Wt {
    namespace Dbo {
...

```

## History

### #1 - 09/04/2013 09:04 PM - Bud T

I was able to get rid of the VisualStudio application crash associated with this "Visual C Package Server has stopped working" error

Problem Event Name: APPCRASH  
Application Name: VCPkgSrv.exe

The solution requires reset of VisualStudio settings as explained here:

<http://stackoverflow.com/questions/2856185/visual-studio-has-stopped-working-on-win-7>

Aside from the appcrash, getting rid of the error requires making sure that all classes used in a header file are forward declared, checking that all necessary includes are placed in associated header files, and changing the order of the code so that everything is defined before used. But after all of that I was only able to exchange the "explicit specialization" errors within one new one:

```

Error 1 error C2504: 'Wt::Dbo::dbo_default_traits' : base class undefined c:\users\user\documents\visual
studio 2010\projects\app\main\source\app.cpp 30

```

The following lines of code are from this file, and line 30 is the one that begins with "struct". I'm not sure how to fix this problem since the file containing the struct is 'ptr' and using '#include <Wt/Dbo/ptr\_impl.h>' only creates more errors.

```

/ *!
 *
 */
#include <Wt/WAnchor>
#include <Wt/WApplication>
#include <Wt/WContainerWidget>
#include <Wt/WHBoxLayout>
#include <Wt/WCssStyleSheet>
#include <Wt/WLink>
#include <Wt/WText>
#include <Wt/WColor>
#include <Wt/WServer>

```

```

#include <Wt/Dbo/backend/Sqlite3>

using namespace Wt;
namespace dbo = Wt::Dbo;
using namespace std;

class User;

typedef Wt::Auth::Dbo::AuthInfo<User> AuthInfo;

namespace Wt {
    namespace Dbo {

        template<>
        struct dbo_traits<User> : public dbo_default_traits {
            typedef std::string IdType;

            static IdType invalidId() {
                return std::string();
            }

            static const char *surrogateIdField() { return 0; }
        };

    }
}

#include "model/User.h"
#include "model/Post.h"
#include "model/Tag.h"
#include "model/Settings.h"

class Settings;
class Tag;
class Post;

class App: public WApplication {
    //...
    void run()
    {
        dbo::backend::Sqlite3 sqlite3(":memory:");

        sqlite3.setProperty("show-queries", "true");
        dbo::Session session;
        session.setConnection(sqlite3);

        session.mapClass<User>("user");
        session.mapClass<Post>("post");
        session.mapClass<Tag>("tag");
        session.mapClass<Settings>("settings");
        session.createTables();

        {
            dbo::Transaction transaction(session);

            User *user = new User();
            user->name = "Joe";
            user->password = "Secret";
            user->role = User::Visitor;
            user->karma = 13;

            dbo::ptr<User> userPtr = session.add(user);
        }
        //...
    };
};

Wt::WApplication* createApplication(const Wt::WEnvironment &env) {
    return new App(env);
}

int main(int argc, char** argv) {
    return Wt::WRun(argc, argv, &createApplication);
}

```

## #2 - 09/10/2013 07:55 PM - Koen Deforche

- Status changed from New to Feedback

- Assignee set to Koen Deforche

Hey,

Including in your top includes should fix this? You are not supposed to include an `_impl.h` header.

The trick with specialization of `dbo_traits` is that you need to define them before a 'ptr' is used.

Regards,

koen

## #3 - 09/11/2013 11:57 PM - Bud T

When attempt to include there is an error stating that the library cannot be found. The include right above it:

```
#include <Wt/Dbp/backend/Sqlite3>
```

Does not give this error. The include works when I change to:

```
#include <Wt/Dbp/Dbp>
#include <Wt/Dbp/backend/Sqlite3>
```

But then there is still a problem:

```
Error 2 error C2664: 'Wt::Dbp::MetaDbp<C>::setId' : cannot convert parameter 1 from 'std::string' to 'const __int64 &' c:\program files (x86)\wt\include\wt\dbp\dbaction_impl.h 516
```

Which is from this line of the tutorial code:

```
class User {
...
    std::string userId;
...
    template<class Action>
    void persist(Action& a)
    {
        dbo::id(a, userId, "user_id", 20);
    }
};
```

Which, according to the debugger, uses this from `dbaction_impl.h`

```
template<class C>
template<typename V>
void SaveDbAction<C>::actId(V& value, const std::string& name, int size)
{
    field(*this, value, name, size);

    /* Later, we may also want to support id changes ? */
    if (pass_ == Self && isInsert_)
        dbo_.setId(value);
}
```

## #4 - 09/12/2013 01:06 AM - Koen Deforche

Hey,

I mean `Wt/Dbp/Dbp` (obviously).

I went back to look at what you really are trying to do --- is it simply to split `tutorial5.C` in multiple files?

koen

## #5 - 09/12/2013 11:40 PM - Bud T

Koen:

Yes, I'm trying to do exactly that: split the `dbo` tutorial into multiple files.

Thanks.

Brad

## #6 - 09/16/2013 12:50 AM - Bud T

Tried this fix as suggested toward the bottom of this page: <http://stackoverflow.com/questions/2130838/convert-stdstring-to-msvc-specific-int64>

```
#include <boost/cstdint.hpp>
typedef boost::int64_t int64;
```

But without success. Same error results:

```
error C2664: 'Wt::Dbo::MetaDbo<C>::setId' : cannot convert parameter 1 from 'std::string' to 'const __int64 &'
c:\program files (x86)\wt\include\wt\dbo\dbaction_impl.h 516
```

I don't think that this error is directly related to splitting of example into multiple files.

## #7 - 09/21/2013 11:25 AM - Bud T

Any thoughts?

## #8 - 09/23/2013 01:53 PM - Wim Dumon

Hello Brad,

I tried your example on my computer with a few modifications to make it compile. I did not run into any unexpected problems. This is the code that compiles on my computer:

```
/*!
 *
 */
#include <Wt/WAnchor>
#include <Wt/WApplication>
#include <Wt/WContainerWidget>
#include <Wt/WHBoxLayout>
#include <Wt/WCssStyleSheet>
#include <Wt/WLink>
#include <Wt/WText>
#include <Wt/WColor>
#include <Wt/WServer>
#include <Wt/Dbo/backend/Sqlite3>
#include <Wt/Dbo/Dbo>

using namespace Wt;
namespace dbo = Wt::Dbo;
using namespace std;

class User;

typedef Wt::Auth::Dbo::AuthInfo<User> AuthInfo;

namespace Wt {
    namespace Dbo {

        template<>
        struct dbo_traits<User> : public dbo_default_traits {
            typedef std::string IdType;

            static IdType invalidId() {
                return std::string();
            }

            static const char *surrogateIdField() { return 0; }
        };

    }
}

#include "../blog/model/User.h"
#include "../blog/model/Post.h"
#include "../blog/model/Tag.h"
#include "../blog/model/Token.h"
// #include "../blog/model/Settings.h"

class Settings;
class Tag;
class Post;

class App: public WApplication {
```

```

public:
    //...
    App(const Wt::WEnvironment &e): WApplication(e) {}
    void run()
    {
        dbo::backend::Sqlite3 sqlite3(":memory:");

        sqlite3.setProperty("show-queries", "true");
        dbo::Session session;
        session.setConnection(sqlite3);

        session.mapClass<User>("user");
        session.mapClass<Post>("post");
        session.mapClass<Tag>("tag");
        //session.mapClass<Settings>("settings");
        session.createTables();

        {
            dbo::Transaction transaction(session);

            User *user = new User();
            user->name = "Joe";
            user->password = "Secret";
            user->role = User::Visitor;
            //user->karma = 13;

            dbo::ptr<User> userPtr = session.add(user);
        }
        //...
    }
};

Wt::WApplication* createApplication(const Wt::WEnvironment &env) {
    return new App(env);
}

int main(int argc, char** argv) {
    return Wt::WRun(argc, argv, &createApplication);
}

```

The model classes are the ones from the blog example.

You did not include `Wt/Dbo/Dbo`, which could certainly have caused the problems you see. The other changes were only related to the model. Does this version work for you?

BR,

Wim.