
Subject: Re: GridCtrl: Adding column dynamically – How to make editable

Posted by [sergeynikitin](#) on Tue, 13 Oct 2009 19:57:54 GMT

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

I have included for the convenience of full original code. Quote:

```
void MyApp::QQuery()
{
    Sql sql ( SQL.GetSession() );
    String current_stmt_string;
    QResultTab.Add ( QResultTab1.SizePos(), t_ ( "QResultTab1" ) );

    QResultTab1.Add ( QResultArray.SizePos(), t_ ( "QResultArray" ) );
    QResultArray.Reset();
    QResultArray.ResizeColMode ( 0 ); // This will change created cloumn's width to use absolute
mode from ratio mode.
//GridControl->
QResultArray.Appending().Removing().Editing().Accepting().Ca nceling().EditCell();
QResultArray.RejectNullRow();

    current_stmt_string = "SELECT * FROM BOOK T1 LEFT JOIN BORROW_RECORD T2 ON
T1.ID = T2.BOOK_ID";

    if ( sql.Execute ( current_stmt_string ) )
    {
        int colCount = sql.GetColumns(); // Craete columns dynamically in the grid

        for ( int i = 0; i < colCount; ++i )
        {
            QResultArray.AddColumn ( sql.GetColumnInfo ( i ).name, t_ ( sql.GetColumnInfo ( i ).name ), 50
).Edit ( ? ? ? ? ? ? ? ? );
        }

        QResultArray.EditCell();

        for ( ;; )
        {
            Vector<Value> row;

            if ( !sql.Fetch ( row ) )
                break;

            QResultArray.Add ( row );
        }
    }

    Sizeable().Zoomable();
}
```

Well, firstly. In this piece of code - the program reads the line, not columns.

Quote:

```
for ( ;; )
{
    Vector<Value> row;

    if ( !sql.Fetch ( row ) )
        break;

    QResultArray.Add ( row );
}
}

Sizeable().Zoomable();
}
```

In my humble opinion there is a lack of understanding of containers U++.

I recommend 1 times per day to sit down and reread the article [http://www.ultimatepp.org/srcdoc\\$Core\\$Tutorial\\$en-us.html](http://www.ultimatepp.org/srcdoc$Core$Tutorial$en-us.html) to a full understanding. There are not very clear written for beginners, but with no understanding of the containers will not be able to write such a program using containers.

For example, in the last quoted piece should be
Vector<Vector<Value> >
instead of
Vector<Value>

And if we talk about editing, then this definition should be in the class definition, but not in the local context. And then you can insert a variable EditString in the definition of the vector.

If the field is of indefinite type and want to use the type of Value, it is better to use the type "EditField", but not the type "EditString".

Then, while reading the lines and the recognition of the fields you want included in the definition of the class definition line for editing:
Vector<EditField> editline

When you add a column should write something like:

```
EditField& f = editline.Add();
QResultArray.AddColumn ( sql.GetColumnInfo ( i ).name, t_ ( sql.GetColumnInfo ( i ).name ), 50
).Edit ( f );
```

or, in short the same:

```
QResultArray.AddColumn ( sql.GetColumnInfo ( i ).name, t_ ( sql.GetColumnInfo ( i ).name ), 50  
) .Edit ( editline.Add() );
```
