

FORMULATOR

v3.8

MathML Suite

Mastering Formulator 3.8
ActiveX Control

Published By

Hermitech, Laboratory of Mathematical and Modeling Software,
Zhytomyr, UKRAINE

E-mail: info@mmlsoft.com

Web: <http://www.mmlsoft.com>

Copyright © 2003-2008 by Hermitech, Laboratory of Mathematical and
Modeling Software. All rights reserved.

Mastering Formulator ActiveX Control

There is a special edition of our software for people seeking to develop new software that is acquainted with presentation or content side of mathematics. This component edition, Formulator ActiveX Control, is the perfect way for software developers to insert mathematics rendering/editing/processing functionality in their applications.

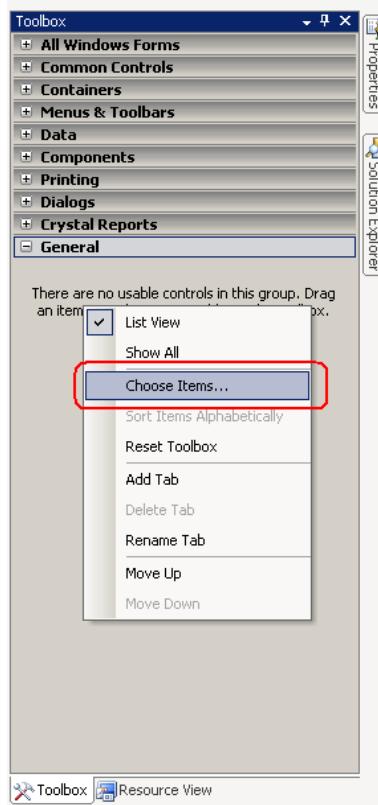
As in the case of Formulator MathML Weaver and Formulator Express products, Hermitech Laboratory provides both commercial and free versions of the component edition: Formulator ActiveX Control and Formulator Express ActiveX Control. The last version has all the restrictions of the Formulator Express (since it is built on the top of this limited free version). Additionally, Formulator Express ActiveX Control is limited in its API, providing only means of exporting mathematical expressions into MathML 2.0. In spite of the reductive level of functionality, Formulator Express ActiveX Control remains extremely useful software component. It allows developers to add a new value to their products by giving a software the competence in math and ready easy-to-use graphical interface that is keenly aware of the mathematical typesetting and semantics rules.

The following tutorial briefly describe how to make use of Formulator ActiveX Control by developing a very simple Visual C# Windows Application, capable of:

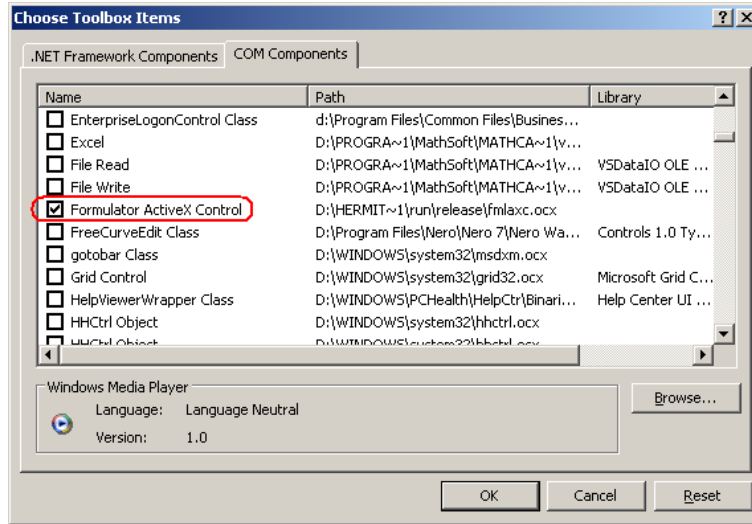
- editing mathematical formulas,
- importing them from MathML 2.0, and
- exporting formulas into MathML 2.0, raster and vector graphics.

This example is part of the standard samples pack, distributing along with Formulator ActiveX Control.

1. On the first step we add a custom Toolbox item for Formulator ActiveX Control.

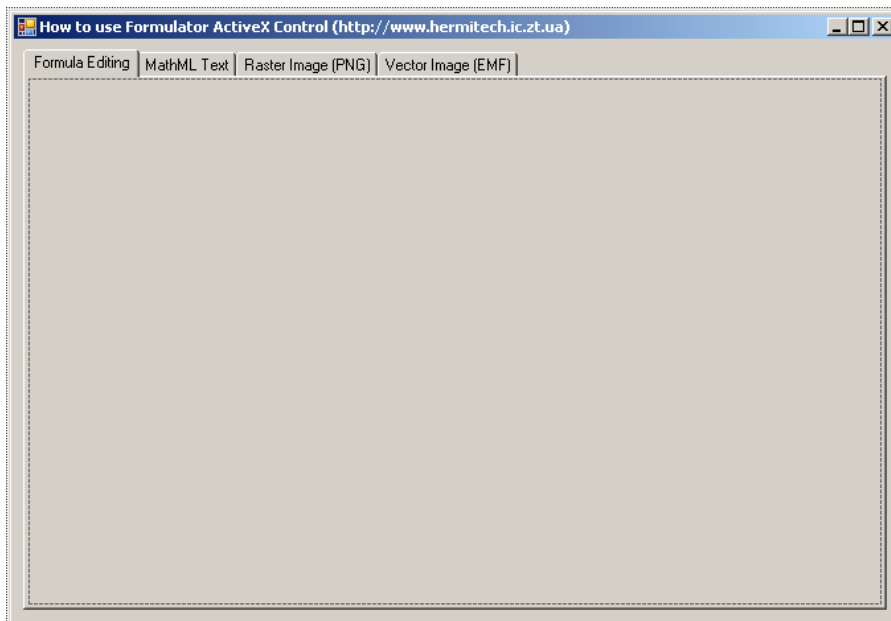


Microsoft Visual Studio environment shows the dialog “Choose Toolbox Items”. Switch to the tab page “COM Components” and in the list of all software components registered on your computer choose the entry for Formulator ActiveX Control.

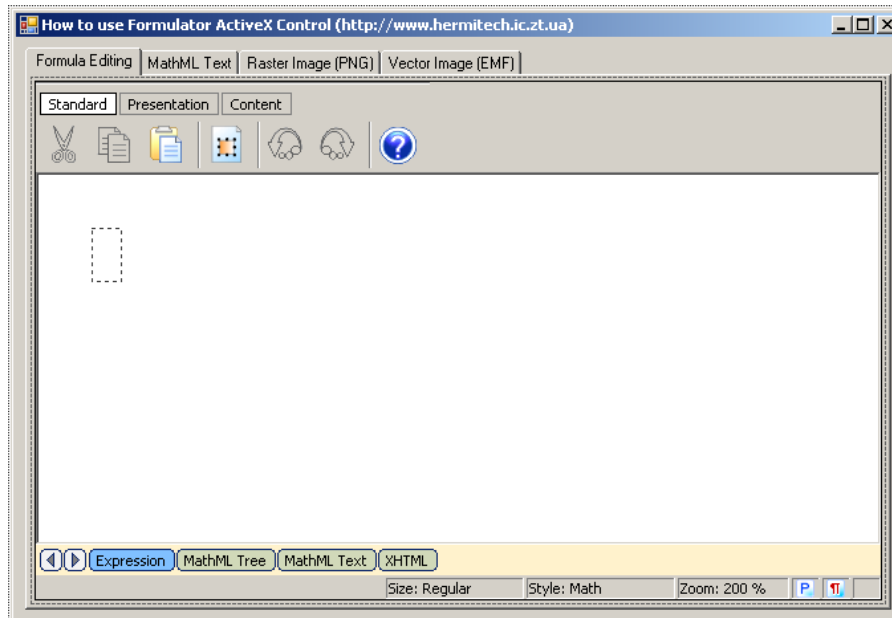


Now there is a new item on the Toolbox. You can use it anytime when you want to enrich your application with mathematical editing and MathML processing functionality.

2. In this example we plan to have a dialog with a tab control and four tab pages. We use a separate tab page for editing mathematics, viewing and editing plain MathML 2.0 text, viewing a raster image of the just typed formula, viewing the image in the vector format (EMF).



3. Now using the corresponding item of the Toolbox, place Formulator ActiveX Control on the first tab page of the form.



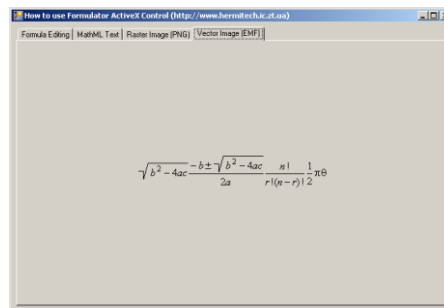
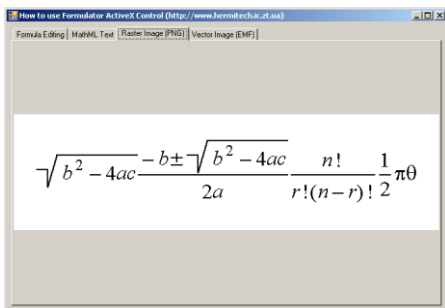
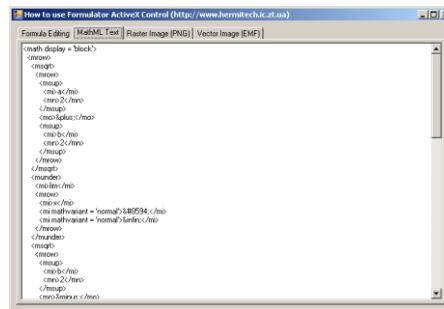
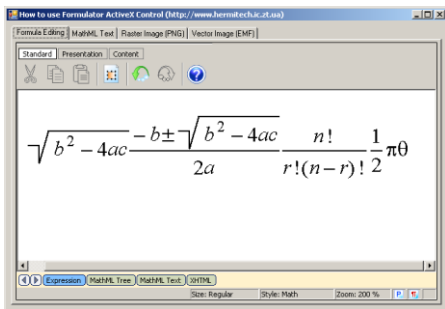
4. Our application is almost ready and the only thing we should do now is to select an event for import/export actions (e.g., switching tab pages of the form) and to call the needed methods of Formulator ActiveX Control (*SetMathML()*, *GetMathMLStr()*, *ExportImage()*, *ExportEMF()*).

```
protected int PreviousSelectedIndex = -1; // the last selected tab page
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch ( tabControl1.SelectedIndex )
    {
        case 0:
            // check whether our ActiveX is licensed as the Express Edition
            // (free version with limited functionality)
            if ( PreviousSelectedIndex == 1 && axFmlaxcl.GetFormulatorLicenseInfo() != 1 )
            {
                // update formula according to the contents of the MathML editing page
                axFmlaxcl.SetMathML( richTextBox1.Text );
            }
            break;
        case 1:
            // check whether our ActiveX is licensed as the Express Edition
            // (free version with limited functionality)
            if ( axFmlaxcl.GetFormulatorLicenseInfo() == 1 )
            {
                richTextBox1.ReadOnly = true;
            }
            // update MathML 2.0 text according to the edited formula
            richTextBox1.Text = axFmlaxcl.GetMathMLStr();
            break;
        case 2:
            // check whether our ActiveX is licensed as the Express Edition
            // (free version with limited functionality)
            if ( axFmlaxcl.GetFormulatorLicenseInfo() == 1 )
            {
                pictureBox1.Enabled = false;
            }
            else
            {
                // create a raster image for the mathematical formula
                // type of the image file is detected by the given file extension
                axFmlaxcl.ExportImage( "out.png" );
                pictureBox1.Load( "out.png" );
            }
            break;
    }
}
```

```

case 3:
    // check whether our ActiveX is licensed as the Express Edition
    // (free version with limited functionality)
    if (axFmlaxcl.GetFormulatorLicenseInfo() == 1)
    {
        pictureBox1.Enabled = false;
    }
    else
    {
        // create a vector image for the mathematical formula
        axFmlaxcl.ExportEMF("out.emf");
        pictureBox2.Load("out.emf");
    }
    break;
}
PreviousSelectedIndex = tabControl1.SelectedIndex;
}
    
```

5. Start our application and see how it works.



6. In several very simple steps we get ready to use application that has features of a professional mathematical editor. Our example shows how Formulator ActiveX Control helps to rapidly develop a software that is aware of the mathematical typesetting and semantics rules and is a good start for projects in such a different domains as computer-aided education, computer algebra systems, authoring tools, and many other applications for mathematics, science, business, economics, etc.