```
/*-----------------------------------------------------------------------------*
 * File Name: CurveFittingWithErrorBars.c                                      *
 * Creation: ER, 03/04/05                                                      *
 * Purpose: Programming Example                                                *
 * Copyright (c) OriginLab Corp.2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010  *
 * All Rights Reserved                                                         *
 *                                                                             *
 * Modification Log:                                                           *
 *-----------------------------------------------------------------------------*/

#include <Origin.h>

////////////////////////////////////////////////////////////////////////////////
// This example shows how to perform curve fitting with error bars.
//
// NOTE:
// 1. It is assumed that a graph is active, with a XYYErr data plot.
// 2. Before compile this file, need run "run.LoadOC(Originlab\nlsf utils.c, 16)"
// to build it into current workspace, else will get linking error.
//
//
#include <..\Originlab\NLFitSession.h>
#include <..\Originlab\nlsf_utils.h>

void curve fitting with error bars()
{
    GraphLayer gl = Project.ActiveLayer();
    if( !gl )
        return;
    DataPlot dp = gl.DataPlots(0);

    DataRange dr;
    dp.GetDataRange(dr);

    vector vX, vY, vW;
    if( dr.GetData(DRR GET DEPENDENT, 0, NULL, NULL, &vY, &vX, NULL, NULL, &vW) < 0 )
        return;

    NLFitSession nlfit;

    // Set function
    string strFunction = "expdec1";
    if ( !nlfit.SetFunction(strFunction) )
        return;

    // Set data
    if ( !nlfit.SetData(vY, vX, NULL, 0, 1, INVALID DATA MODE, vW) )
        return;

    // Set weight method
    if( !nlfit.SetWeightData(WEIGHT INSTRUMENTAL, 0.0, 0.0, 0.0) )
        return;

    // Parameter initialization
    if ( !nlfit.ParamsInitValues() )
        return;

    // Do fit and output fit outcome
    int nFitOutcome;
    nlfit.Fit(&nFitOutcome);
    string strOutcome = nlfit.GetFitOutCome(nFitOutcome);
    out str("Outcome of the fitting session: " + strOutcome);

    // Get parameters values and names
    vector<double>  vParamValues;
    vector<int>     vParamOffsets;
    vector<string>  vsParamNames;
    nlfit.GetParamValuesAndOffsets(vParamValues, vParamOffsets);
    nlfit.GetParamNamesInFunction(vsParamNames);

    // Output paramter values with names:
    for ( int nParam = 0; nParam < vsParamNames.GetSize(); nParam++ )
    {
        printf("%s = %lf\n", vsParamNames[nParam], vParamValues[nParam]);
    }

    // Evaluate fit Y data and put into a hidden worksheet
    Worksheet wksFit;
    wksFit.Create("Origin", CREATE_HIDDEN);
    wksFit.Columns(1).SetLongName("Fitted Y Data");

    vector vFitY(vX.GetSize());
    if( !nlsf_evaluate(strFunction, "", vX, vFitY, vParamValues) )
```

```
        return;

    Dataset dsFitX(wksFit, 0);
    dsFitX = vX;
    Dataset dsFitY(wksFit, 1);
    dsFitY = vFitY;

    // plot fitted curve on source graph
    Curve crvFit(wksFit, 0, 1);
    int nPlotIndex = gl.AddPlot(crvFit, IDM_PLOT LINE);
    DataPlot dpFit = gl.DataPlots(nPlotIndex);
    dpFit.SetColor(SYSCOLOR RED);

    legend update(gl); // Refresh legend
}
////////////////////////////////////////////////////////////////////////////////
```