

```

/*-----*
* File Name: SimpleImportAndPlot.c                               *
* Creation: ER, 01/24/05                                         *
* 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 a simple import of a data file into
// the current worksheet and then create a graph from the imported data.
//
void simple import and plot()
{
    // Bring up file dialog for user to select the file.
    // Replace *.dat with your desired extension as needed.
    string strFile = GetOpenBox("*.dat");
    if( strFile.IsEmpty() )
    {
        out_str("No file was selected!");
        return;
    }

    // Create a new worksheet
    Worksheet wks;
    // If you wish to use a custom worksheet template, then specify the name of
    // your template in place of Origin.OTW
    wks.Create("Origin.OTW");

    // Import the file into the worksheet using the current ASCIMP
    // (Import ASCII Options) stored in the worksheet.
    // You can edit the ASCIMP options from the GUI and save it back to
    // the worksheet and then save the worksheet as a custom template.
    // You can then create a new instance of your custom template and use it
    // to import the data next time.
    BOOL bRet = wks.ImportASCII(strFile);
    if( !bRet )
    {
        out_str("Failed to import file!");
        // Destroy newly created worksheet
        wks.Destroy();
        return;
    }

    // Create a default graph and plot all Y columns in the first layer
    // as a grouped plot.
    GraphPage gpg;
    // If you wish to use a custom graph template, then specify the name
    // of your template in place of Origin.OTP
    gpg.Create("Origin.OTP");
    // Point to the first layer in the graph
    // If you use a custom template that has multiple layers, you can
    // then declare separate layer objects for each layer and move
    // data plots into various layers as desired
    GraphLayer gly = gpg.Layers(0);
    // Loop over all columns and add to layer if Y column
    foreach(Column col in wks.Columns)
    {
        if(OKDATAOBJ DESIGNATION Y == col.GetType())
        {
            // Create a curve object using this column
            Curve crv(wks, col.GetIndex());
            // Add as a line plot
            gly.AddPlot(crv, IDM PLOT LINE);
        }
    }
    // Group all the plots and rescale
    gly.GroupPlots(0);
    gly.Rescale();

    legend update(gly); // Refresh legend
}
////////////////////////////////////

```