```
public partial class frMain : Form
    {
        public frMain()
        {
            frMain.CheckForIllegalCrossThreadCalls = false;
            InitializeComponent();
        }

        frDialog frDialog = new frDialog();

        private void btnFrom_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFile = new OpenFileDialog();
            openFile.Filter = "(*.pdf,*.doc,*.docx)|*.pdf;*.doc;*.docx;";
            var r= openFile.ShowDialog();
            if (openFile.FileName == null)
            {
                MessageBox.Show("请选择文件");
                return;
            }
            string src = openFile.FileName;

            textBox1.Text = src;
        }

        private (string,bool) CheckFileSize(string src)
        {
            try
            {
                if (string.IsNullOrWhiteSpace(src))
                {
                    return ("文件不能为空", false);
                }
                var size = new FileInfo(src).Length;
                return size > 1 * 1024 * 1024 ? ("此版本转换文件不能大于 1M，请联系作者
升级！ ", false) : ("OK", true);
            }
            catch (Exception)
            {

                return ("源文件错误，请检查！ ", false);
            }

        }
```

```
        private void btnTo_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog path = new FolderBrowserDialog();
            path.ShowDialog();

            if (path.SelectedPath == null)
            {
                MessageBox.Show("请选择目录");
                return;
            }
            string src = path.SelectedPath;

            textBox2.Text = src;
        }

        private void btnConvert_Click(object sender, EventArgs e)
        {

            if
( string.IsNullOrWhiteSpace(textBox1.Text)||string.IsNullOrWhiteSpace(textBox2.Text))
            {
                MessageBox.Show("未选择文件或目录");
                return;
            }

            var rSize = CheckFileSize(textBox1.Text);
            if (!rSize.Item2)
            {
                MessageBox.Show(rSize.Item1);
                return;
            }

            string src = textBox1.Text;
            string fileName = @"\" + System.IO.Path.GetFileName(src).Split('.')[0];
            string                                    path                                    =
$"{textBox2.Text}{fileName}_{DateTime.Now.ToString("yyyyMMddHHmmss")}";


            frDialog.Exec("正在处理，请稍等...", this, () =>
            {
                ConvertTo(src, path);
            });
```

```
        //EasyDoc.Util.Dialog.Show("正在处理中，请稍候...", this, (obj) =>
        //{
        //    //这里写处理耗时的代码，代码处理完成则自动关闭该窗口
        //    ConvertTo(src, path);
        //}, null);


    }


    public void SetEnable(bool enable=false)
    {
        this.btnConvert.Enabled = enable;
        this.btnConvert2.Enabled = enable;
    }

    private void ConvertTo(string src, string path)
    {
        try
        {
            SetEnable(false);
            bool r = false;
            if (System.IO.Path.GetExtension(src) == ".pdf")
            {
                r = Pdf.ToDocx(src, path + ".docx");
            }
            else if (System.IO.Path.GetExtension(src) == ".doc" ||
System.IO.Path.GetExtension(src) == ".docx")
            {

                r = Word.ToPdf(src, path + ".pdf");
            }
            else
            {
                SetEnable(true);
                MessageBox.Show("所选文件格式必须是 pdf、doc、docx!");
                return;
            }
            if (r)
            {
                MessageBox.Show("转换成功!");
            }
            else
            {
```

```
                MessageBox.Show("转换失败!");
            }
        }
        catch (Exception ex)
        {

            MessageBox.Show($"转换异常:{ex.Message}!");
        }
        finally
        {
            SetEnable(true);
        }

    }



    private void btnFrom2_Click(object sender, EventArgs e)
    {
        OpenFileDialog openFile = new OpenFileDialog();
        openFile.Filter = "(*.pdf,*.doc,*.docx,*.xls,*.xlsx)|*.pdf;*.doc;*.docx;*.xls;*.xlsx";
        var r= openFile.ShowDialog();
        if (openFile.FileName == null)
        {
            MessageBox.Show("请选择文件");
            return;
        }

        string src = openFile.FileName;

        txtSrc2.Text = src;
    }

    private void btnTo2_Click(object sender, EventArgs e)
    {
        var src = txtSrc2.Text;
        if (string.IsNullOrWhiteSpace(src)|| (src.LastIndexOf("\\")<0))
        {
            MessageBox.Show("请先选择源文件");
            return;
        }
        SaveFileDialog saveFile = new SaveFileDialog();
        saveFile.Filter                                                                                    =
"(*.doc)|*.doc|(*.docx)|*.docx|(*.xls)|*.xls|(*.xlsx)|*.xlsx|(*.pdf)|*.pdf|(*.txt)|*.txt|(*.jpg)|*.j
```

```
pg";
                var srcFileName= src.Substring(src.LastIndexOf("\\")+1);
                var srcFileType = System.IO.Path.GetExtension(src);
                var fileName = srcFileName.Substring(0,srcFileName.LastIndexOf("."));
                saveFile.Filter = saveFile.Filter.Replace($"(*{srcFileType})|*{srcFileType}|","");//相
同文件类型不转换
                if (srcFileType.Contains(".xls"))
                {
                    saveFile.Filter = "(*.pdf)|*.pdf";
                }


                saveFile.FileName                                                                    =
$"{fileName}_{DateTime.Now.ToString("yyyyMMddHHmmss")}";
                var r= saveFile.ShowDialog();

                if (r==DialogResult.OK)
                {
                    string dec = saveFile.FileName;
                    txtDec2.Text = dec;
                }

            }

            List<string> srcTypeEnable = new List<string>() { ".pdf", ".doc", ".docx", ".xls", ".xlsx" };
            List<string> decTypeEnable = new List<string>() { ".pdf", ".doc", ".docx", ".xls", ".xlsx",
".txt", ".jpg" };
            private void btnConvert2_Click(object sender, EventArgs e)
            {
                frDialog.Exec("正在处理，请稍等...", this, () =>
                {
                    ConverTo2();
                });
            }

            public void ConverTo2()
            {

                try
                {

                    bool r = false;
                    var src = txtSrc2.Text;
                    var dec = txtDec2.Text;
```

```
var srcType = System.IO.Path.GetExtension(src);
var decType = System.IO.Path.GetExtension(dec);
if (string.IsNullOrWhiteSpace(src))
{
    MessageBox.Show("请先选择源文件!");
    return;
}
if (string.IsNullOrWhiteSpace(dec))
{
    MessageBox.Show("请先选择另存为文件!");
    return;
}
if (!srcTypeEnable.Contains(srcType))
{
    MessageBox.Show("源文件格式不支持!");
    return;
}
if (!decTypeEnable.Contains(decType))
{
    MessageBox.Show("另存为文件格式不支持!");
    return;
}
if (srcType == decType)
{
    MessageBox.Show("源文件和另存为文件格式相同不支持!");
    return;
}
var rSize = CheckFileSize(src);
if (!rSize.Item2)
{
    MessageBox.Show(rSize.Item1);
    return;
}
SetEnable(false);
switch (srcType)
{
    case ".pdf": r = PdfTo(decType); break;
    case ".doc": case ".docx": r = DocTo(decType); break;
    case ".xls": case ".xlsx": r = XlsTo(decType); break;
    default:
        break;
}
if (r)
{
```

```
                MessageBox.Show("转换成功!");
            }
            else
            {
                MessageBox.Show("转换失败!");
            }



        }
        catch (Exception ex)
        {

            MessageBox.Show($"转换异常:{ex.Message}!");
        }
        finally
        {
            SetEnable(true);
        }
}

private bool PdfTo(string decType)
{
        var r = false;
        var src = txtSrc2.Text;
        var dec = txtDec2.Text;
        switch (decType)
        {
            case ".doc": r = Pdf.ToDoc(src,dec); break;
            case ".docx": r=Pdf.ToDocx(src, dec); break;
            case ".xls":
            case ".xlsx":r = Pdf.ToExcel(src,dec); break;
            case ".txt": r = Pdf.ToTxt(src, dec); break;
            case ".jpg": r = Pdf.ToJpg(src, dec); break;
        }
        return r;
}

private bool DocTo(string decType)
{
        var r = false;
        var src = txtSrc2.Text;
        var dec = txtDec2.Text;
        switch (decType)
```

```
            {
                case ".pdf": r = Word.ToPdf(src, dec); break;
                case ".doc": r = Word.ToDoc(src, dec); break;
                case ".docx": r = Word.ToDocx(src, dec); break;
                case ".xls":
                case ".xlsx": r = Word.ToExcel(src, dec); break;
                case ".txt": r = Word.ToTxt(src, dec); break;
                case ".jpg": r = Word.ToJpg(src, dec); break;
            }
            return r;
        }
        private bool XlsTo(string decType)
        {
            var r = false;
            var src = txtSrc2.Text;
            var dec = txtDec2.Text;
            switch (decType)
            {
                //case ".doc": r = Excel.ToDoc(src, dec); break;
                //case ".docx": r = Excel.ToDocx(src, dec); break;
                case ".pdf": r = Excel.ToPdf(src, dec); break;
                //case ".txt": r = Excel.ToTxt(src, dec); break;
               // case ".jpg": r = Excel.ToJpg(src, dec); break;
            }
            return r;
        }

        private void 关于ToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            var frAbout = new frAbout();
            frAbout.ShowDialog(this);
            //frAbout.StartPosition = FormStartPosition.CenterParent;
        }
    }
```

```
namespace EasyDoc
{
    partial class frAbout
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // label1
            //
```

```
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("宋体", 9F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(134)));
            this.label1.Location = new System.Drawing.Point(36, 19);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(31, 12);
            this.label1.TabIndex = 0;
            this.label1.Text = "版本";
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(78, 20);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(77, 12);
            this.label2.TabIndex = 1;
            this.label2.Text = "试用版 v1.0.0";
            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Font = new System.Drawing.Font("宋体", 9F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(134)));
            this.label3.Location = new System.Drawing.Point(35, 50);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(31, 12);
            this.label3.TabIndex = 0;
            this.label3.Text = "作者";
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(80, 50);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(53, 12);
            this.label4.TabIndex = 1;
            this.label4.Text = "chidreal";
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.Font = new System.Drawing.Font("宋体", 9F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(134)));
            this.label5.ForeColor = System.Drawing.Color.Red;
```

```
            this.label5.Location = new System.Drawing.Point(163, 114);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(116, 12);
            this.label5.TabIndex = 0;
            this.label5.Text = "仅供学习 禁止商用";
            //
            // label6
            //
            this.label6.AutoSize = true;
            this.label6.Location = new System.Drawing.Point(73, 147);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(335, 12);
            this.label6.TabIndex = 1;
            this.label6.Text = "Copyright @2022-2027 office 小助手 chidreal 保留所有权利
";

            //
            // frAbout
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(466, 182);
            this.Controls.Add(this.label6);
            this.Controls.Add(this.label4);
            this.Controls.Add(this.label5);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.label1);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D;
            this.ImeMode = System.Windows.Forms.ImeMode.NoControl;
            this.MaximizeBox = false;
            this.Name = "frAbout";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "关于";
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;
```

```
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label6;
    }
    }




        public static bool ToWord2(string src, string dec)
        {

            using (var pdfDocument = new PdfDocument(new PdfReader(src)))
            {

                using (var fos = System.IO.File.OpenWrite(dec))
                {

                    for (var pageIndex = 1; pageIndex <= pdfDocument.GetNumberOfPages();
pageIndex++)
                    {

                        var strategy = new LocationTextExtractionStrategy();
                        var parser = new PdfCanvasProcessor(strategy);
                        parser.ProcessPageContent(pdfDocument.GetPage(pageIndex));
                        var array = Encoding.UTF8.GetBytes(strategy.GetResultantText());

                        fos.Write(array, 0, array.Length);
                        fos.Flush();
                    }
                    return true;
                }


            }

            }
```

```
        public static bool ToDoc(string src, string dec)
        {
            File.Copy(src, dec);

            return true;
        }

        public static bool ToDocx(string src, string dec)
        {
            File.Copy(src, dec);
            return true;
        }
```

```
  public static bool ToPdf2(string sourcePath, string dec)
        {
            try
            {
                bool result = false;
                 Microsoft.Office.Interop.Word.Application application = new
Microsoft.Office.Interop.Word.Application();
                Microsoft.Office.Interop.Word.Document document = null;
                try
                {
                    application.Visible = false;
                    document = application.Documents.Open(sourcePath);
                    string lastChar = sourcePath.Substring(sourcePath.Length - 1, 1);
                    string PDFPath = string.Empty;
                    if (lastChar == "x")
                    {
                        PDFPath = sourcePath.Replace(".docx", ".pdf");//pdf 存放位置
                    }
                    else
                    {
                        PDFPath = sourcePath.Replace(".doc", ".pdf");//pdf 存放位置
                    }
                    //if (!File.Exists(@PDFPath))//存在 PDF，不需要继续转换
                    //{
```

```
                    //    document.ExportAsFixedFormat(PDFPath,
Microsoft.Office.Interop.Word.WdExportFormat.wdExportFormatPDF);
                    //}
                    document.ExportAsFixedFormat(dec,
Microsoft.Office.Interop.Word.WdExportFormat.wdExportFormatPDF);
                    result = true;
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.Message);
                    result = false;

                }
                finally
                {
                    document.Close();
                }
                return result;
            }
            catch (Exception ex)
            {
                return false;
            }


            }
```

```
  public class Dialog
    {
        public static void Show(string msg, Form owner, ParameterizedThreadStart work,
object workArg = null)
        {
            FrmProcessing processingForm = new FrmProcessing(msg);
            dynamic expObj = new ExpandoObject();
            expObj.Form = processingForm;
            expObj.WorkArg = workArg;
            processingForm.SetWorkAction(work, expObj);
            processingForm.ShowDialog(owner);
            if (processingForm.WorkException != null)
            {
                throw processingForm.WorkException;
            }
        }
        }
```