NPOI创建DOCX常用操作

1. 创建文档

XWPFDocument m\_Docx = new XWPFDocument();

1. 页面设置

//1‘=1440twip=25.4mm=72pt(磅point)=96px(像素pixel)

//1px(像素pixel)=0.75pt(磅point)

// A4:W=11906 twip=8.269''=210mm,h=16838 twip=11.693''=297mm

//A5:W=8390 twip=5.827''=148mm,h=11906 twip=8.269''=210mm

//A6:W=5953 twip=4.134''=105mm,h=8390 twip=5.827''=1148mm

//16k195mmX270mm:

//16k184mmX260mm:

//16k197mmX273mm:

CT\_SectPr m\_SectPr = new CT\_SectPr();

//页面设置A4横向

m\_SectPr.pgSz.w = (ulong)16838;

m\_SectPr.pgSz.h = (ulong)11906;

m\_Docx.Document.body.sectPr = m\_SectPr;

1. 创建段落

1) XWPFParagraph gp = m\_Docx.CreateParagraph();

2) CT\_P m\_p = m\_Docx.Document.body.AddNewP();

m\_p.AddNewPPr().AddNewJc().val = ST\_Jc.center;//段落水平居中

XWPFParagraph gp = new XWPFParagraph(m\_p, m\_Docx); //创建XWPFParagraph

1. 段首行缩进

gp.IndentationFirstLine=(int)100；

可以用一个函数计算

protected int Indentation(String fontname, int fontsize, int Indentationfonts, FontStyle fs)

{

//字显示宽度，用于段首行缩进

//字号与fontsize关系

//初号（0号）=84，小初=72，1号=52，2号=44，小2=36，3号=32，小3=30，4号=28，

//小4=24，5号=21，小5=18，6号=15，小6=13，7号=11，8号=10

Graphics m\_tmpGr = this.CreateGraphics();

m\_tmpGr.PageUnit = GraphicsUnit.Point;

SizeF size = m\_tmpGr.MeasureString("好", new Font(fontname, fontsize \* 0.75F, fs));

return (int)size.Width \* Indentationfonts \* 10;

}

gp.IndentationFirstLine = Indentation("宋体", 21, 2, FontStyle.Regular);//段首行缩进2字符

1. 行距设置

//单倍为默认值（240twip）不需设置，1.5倍=240X1.5=360twip，2倍=240X2=480twip

m\_p.AddNewPPr().AddNewSpacing().line = "400";//行距固定20磅

m\_p.AddNewPPr().AddNewSpacing().lineRule = ST\_LineSpacingRule.exact;

1. 创建RUN

1) XWPFRun gr = gp.CreateRun();

gr.GetCTR().AddNewRPr().AddNewRFonts().ascii = "黑体";

gr.GetCTR().AddNewRPr().AddNewRFonts().eastAsia = "黑体";

gr.GetCTR().AddNewRPr().AddNewRFonts().hint = ST\_Hint.eastAsia;

gr.GetCTR().AddNewRPr().AddNewSz().val = (ulong)44;//2号字体

gr.GetCTR().AddNewRPr().AddNewSzCs().val = (ulong)44;

gr.GetCTR().AddNewRPr().AddNewB().val = true; //加粗

gr.GetCTR().AddNewRPr().AddNewColor().val = "red";//字体颜色

gr.SetText("DOCX表");

2) CT\_R = m\_p.AddNewR();

1. 创建表
2. 创建表

有两种方法：

a.方法1

XWPFTable table = m\_Docx.CreateTable(1, 1);//创建1行1列表

CT\_Tbl m\_CTTbl = m\_Docx.Document.body.GetTblArray()[0];//获得文档第一张表

b.方法2

CT\_Tbl m\_CTTbl = m\_Docx.Document.body.AddNewTbl();

XWPFTable table = new XWPFTable(m\_CTTbl, m\_Docx);//创建1行1列表

1. 表水平居中

m\_CTTbl.AddNewTblPr().jc = new CT\_Jc();

m\_CTTbl.AddNewTblPr().jc.val = ST\_Jc.center;//表在页面水平居中

1. 表宽度

m\_CTTbl.AddNewTblPr().AddNewTblW().w = "2000"; //表宽度

m\_CTTbl.AddNewTblPr().AddNewTblW().type = ST\_TblWidth.dxa;

1. 表定位

//若tblpXSpec、tblpX同时存在，则tblpXSpec优先tblpX；

//若tblpYSpec、tblpY同时存在，则tblpYSpec优先tblpY；

m\_CTTblPr.tblpPr = new CT\_TblPPr();//表定位

m\_CTTblPr.tblpPr.tblpX = "4003";//表左上角坐标

m\_CTTblPr.tblpPr.tblpY = "365";

//m\_CTTblPr.tblpPr.tblpXSpec = ST\_XAlign.center;// tblpXSpec优先tblpX

//m\_CTTblPr.tblpPr.tblpYSpec = ST\_YAlign.top;// tblpYSpec优先tblpY

m\_CTTblPr.tblpPr.leftFromText = (ulong)180;

m\_CTTblPr.tblpPr.rightFromText = (ulong)180;

m\_CTTblPr.tblpPr.vertAnchor = ST\_VAnchor.text;

m\_CTTblPr.tblpPr.horzAnchor = ST\_HAnchor.page;

1. 列宽设置

//列宽设置

CT\_TcPr m\_Pr = table.GetRow(0).GetCell(0).GetCTTc().AddNewTcPr();

m\_Pr.tcW = new CT\_TblWidth();

m\_Pr.tcW.w = "1500";//单元格宽

m\_Pr.tcW.type = ST\_TblWidth.dxa;

m\_Pr = table.GetRow(0).GetCell(1).GetCTTc().AddNewTcPr();

m\_Pr.tcW = new CT\_TblWidth();

m\_Pr.tcW.w = "1000";//单元格宽

m\_Pr.tcW.type = ST\_TblWidth.dxa;

1. 创建行
2. XWPFTableRow m\_Row = table.CreateRow();//创建一行
3. XWPFTableRow m\_Row = table.InsertNewTableRow(0);//表头插入一行
4. XWPFTableRow td3 = table.InsertNewTableRow(table.Rows.Count - 1);//插入行
5. CT\_Row m\_NewRow = new CT\_Row();

XWPFTableRow m\_Row = new XWPFTableRow(m\_NewRow, table);

table.AddRow(m\_Row);

1. 行高设置
2. m\_Row.GetCTRow().AddNewTrPr().AddNewTrHeight().val = (ulong)426;
3. m\_NewRow.AddNewTrPr().AddNewTrHeight().val = (ulong)426;
4. 创建单元格
5. XWPFTableCell cell = m\_Row.CreateCell();//创建一单元格,创建单元格时就创建了一个CT\_P
6. XWPFTableCell cell = m\_Row.AddNewTableCell();//创建单元格时创建了一个CT\_P
7. 单元格设置文字

table.GetRow(0).GetCell(0).SetText("111");

1. 列合并

//表增加行，合并列

CT\_Row m\_NewRow = new CT\_Row();

XWPFTableRow m\_Row = new XWPFTableRow(m\_NewRow, table);

table.AddRow(m\_Row);

XWPFTableCell cell = m\_Row.CreateCell();

CT\_Tc cttc = cell.GetCTTc();

CT\_TcPr ctPr = cttc.AddNewTcPr();

ctPr.gridSpan = new CT\_DecimalNumber();

ctPr.gridSpan.val = "3"; //合并3列

cttc.GetPList()[0].AddNewPPr().AddNewJc().val = ST\_Jc.center;

cttc.GetPList()[0].AddNewR().AddNewT().Value = "sss";

1. 行合并

//1行

CT\_Row m\_NewRow = new CT\_Row();

XWPFTableRow m\_Row = new XWPFTableRow(m\_NewRow, table);

table.AddRow(m\_Row);

XWPFTableCell cell = m\_Row.CreateCell();

CT\_Tc cttc = cell.GetCTTc();

CT\_TcPr ctPr = cttc.AddNewTcPr();

ctPr.AddNewVMerge().val = ST\_Merge.restart;//合并行

ctPr.AddNewVAlign().val = ST\_VerticalJc.center;//垂直

cttc.GetPList()[0].AddNewPPr().AddNewJc().val = ST\_Jc.center;

cttc.GetPList()[0].AddNewR().AddNewT().Value = "xxx";

//2行，多行合并类似

m\_NewRow = new CT\_Row();

m\_Row = new XWPFTableRow(m\_NewRow, table);

table.AddRow(m\_Row);

cell = m\_Row.CreateCell();

cttc = cell.GetCTTc();

ctPr = cttc.AddNewTcPr();

ctPr.AddNewVMerge().val = ST\_Merge.@continue;//合并行

1. 插图
2. 内联式插图(inline)

此种插图方式对插入的图片位置不能灵活控制，只能通过段设置，对应word的嵌入型插图。宽和高数值换算：1cm=360000 EMUS（English Metric Unit）。

FileStream gfs = null;

gfs = new FileStream("f:\\pic\\1.jpg", FileMode.Open, FileAccess.Read);

m\_p = m\_Docx.Document.body.AddNewP();

m\_p.AddNewPPr().AddNewJc().val = ST\_Jc.center;//段落水平居中

gp = new XWPFParagraph(m\_p, m\_Docx);

gr = gp.CreateRun();

gr.AddPicture(gfs, (int)NPOI.XWPF.UserModel.PictureType.JPEG, "1.jpg", 1000000, 1000000);

gfs.Close();

1. 锚式插图(anchor)

此种插图方式对插入的图片位置能灵活控制，对应word的四周型、紧密型、穿越型等。图的左上角坐标及宽和高数值换算：1cm=360000 EMUS（English Metric Unit）。

gfs = new FileStream("f:\\pic\\1.jpg", FileMode.Open, FileAccess.Read);

m\_p = m\_Docx.Document.body.AddNewP();

m\_p.AddNewPPr().AddNewJc().val = ST\_Jc.center;

gp = new XWPFParagraph(m\_p, m\_Docx);

gr = gp.CreateRun();

CT\_Anchor an = new CT\_Anchor();

//图片距正文上(distT)、下(distB)、左(distL)、右(distR)的距离。114300EMUS=3.1mm

an.distB = (uint)(0);

an.distL = 114300u;

an.distR = 114300U;

an.distT = 0U;

an.relativeHeight = 251658240u;

an.behindDoc = false; //"0"，图与文字的上下关系

an.locked = false; //"0"

an.layoutInCell = true; //"1"

an.allowOverlap = true; //"1"

CT\_Positive2D simplePos = new CT\_Positive2D();

simplePos.x = (long)0;

simplePos.y = (long)0;

CT\_EffectExtent effectExtent = new CT\_EffectExtent();

effectExtent.b = 0L;

effectExtent.l = 0L;

effectExtent.r = 0L;

effectExtent.t = 0L;

//图左上角坐标

CT\_PosH posH = new CT\_PosH();

posH.relativeFrom = ST\_RelFromH.column;

posH.posOffset = 4000000;//单位：EMUS，1CM=360000EMUS

CT\_PosV posV = new CT\_PosV();

posV.relativeFrom = ST\_RelFromV.paragraph;

posV.posOffset = 200000;

1. 四周型

CT\_WrapSquare wrapSquare = new CT\_WrapSquare();

wrapSquare.wrapText = ST\_WrapText.bothSides;

gr.AddPicture(gfs, (int)NPOI.XWPF.UserModel.PictureType.JPEG, "1.jpg", 1000000, 1000000,

posH, posV, wrapSquare,anchor,simplePos,effectExtent);

1. 紧密型

CT\_WrapTight wrapTight = new CT\_WrapTight();

wrapTight.wrapText = ST\_WrapText.bothSides;

wrapTight.wrapPolygon = new CT\_WrapPath();

wrapTight.wrapPolygon.edited = false;

wrapTight.wrapPolygon.start = new CT\_Positive2D();

wrapTight.wrapPolygon.start.x = 0;

wrapTight.wrapPolygon.start.y = 0;

CT\_Positive2D lineTo = new CT\_Positive2D();

wrapTight.wrapPolygon.lineTo = new List<CT\_Positive2D>();

lineTo = new CT\_Positive2D();

lineTo.x = 0;

lineTo.y = 21394;

wrapTight.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 21806;

lineTo.y = 21394;

wrapTight.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 21806;

lineTo.y = 0;

wrapTight.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 0;

lineTo.y = 0;

wrapTight.wrapPolygon.lineTo.Add(lineTo);

gr.AddPicture(gfs, (int)NPOI.XWPF.UserModel.PictureType.JPEG, "1.jpg", 720000, 720000,

posH, posV, wrapTight, anchor, simplePos, effectExtent);

1. 穿越型

CT\_WrapThrough wrapThrough = new CT\_WrapThrough();

wrapThrough.wrapText = ST\_WrapText.bothSides;

wrapThrough.wrapPolygon = new CT\_WrapPath();

wrapThrough.wrapPolygon.edited = false;

wrapThrough.wrapPolygon.start = new CT\_Positive2D();

wrapThrough.wrapPolygon.start.x = 0;

wrapThrough.wrapPolygon.start.y = 0;

CT\_Positive2D lineTo = new CT\_Positive2D();

wrapThrough.wrapPolygon.lineTo = new List<CT\_Positive2D>();

lineTo = new CT\_Positive2D();

lineTo.x = 0;

lineTo.y = 21394;

wrapThrough.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 21806;

lineTo.y = 21394;

wrapThrough.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 21806;

lineTo.y = 0;

wrapThrough.wrapPolygon.lineTo.Add(lineTo);

lineTo = new CT\_Positive2D();

lineTo.x = 0;

lineTo.y = 0;

wrapThrough.wrapPolygon.lineTo.Add(lineTo);

gr.AddPicture(gfs, (int)NPOI.XWPF.UserModel.PictureType.JPEG, "1.jpg", 720000, 720000,

posH, posV, wrapThrough, anchor, simplePos, effectExtent);

1. 页眉页脚设置
2. 页眉设置

XWPFDocument m\_Docx = new XWPFDocument();

m\_Docx.Document.body.sectPr = new CT\_SectPr();

CT\_SectPr m\_SectPr = m\_Docx.Document.body.sectPr;

//创建页眉

CT\_Hdr m\_Hdr = new CT\_Hdr();

m\_Hdr.AddNewP().AddNewR().AddNewT().Value = "hhh";//页眉内容

//创建页眉关系（headern.xml）

XWPFRelation Hrelation = XWPFRelation.HEADER;

XWPFHeader m\_h = (XWPFHeader)m\_Docx.CreateRelationship(Hrelation, XWPFFactory.GetInstance(),

m\_Docx.HeaderList.Count + 1);

//设置页眉

m\_h.SetHeaderFooter(m\_Hdr);

CT\_HdrFtrRef m\_HdrFtr = m\_SectPr.AddNewHeaderReference();

m\_HdrFtr.type = ST\_HdrFtr.@default;

m\_HdrFtr.id = m\_h.GetPackageRelationship().Id;

1. 页脚设置

XWPFDocument m\_Docx = new XWPFDocument();

//页面设置

m\_Docx.Document.body.sectPr = new CT\_SectPr();

CT\_SectPr m\_SectPr = m\_Docx.Document.body.sectPr;

//创建页脚

CT\_Ftr m\_ftr = new CT\_Ftr();

m\_ftr.AddNewP().AddNewR().AddNewT().Value = "fff";//页脚内容

//创建页脚关系（footern.xml）

XWPFRelation Frelation = XWPFRelation.FOOTER;

XWPFFooter m\_f = (XWPFFooter)m\_Docx.CreateRelationship(Frelation, XWPFFactory.GetInstance(),

m\_Docx.FooterList.Count + 1);

//设置页脚

m\_f.SetHeaderFooter(m\_ftr);

CT\_HdrFtrRef m\_HdrFtr = m\_SectPr.AddNewFooterReference();

m\_HdrFtr.type = ST\_HdrFtr.@default;

m\_HdrFtr.id = m\_f.GetPackageRelationship().Id;

1. 脚注尾注

创建脚注和尾注，首先要设置格式，其次创建脚注和尾注内容，最后在正文中标注。

1. 格式设置

在正文中标注脚注采用阿拉伯数字且为上标，而标注尾注采用罗马数字且为上标，在word中可以事先用格式存储在格式xml中，以后可以在正文引用其格式即可。

创建格式文件（styles.xml）：XWPFStyles m\_styles = m\_Docx.CreateStyles();

创建格式xml：CT\_Styles m\_ctstyles = new CT\_Styles();可以根据需要创建相应的格式。在此只列举与脚注有关的格式a6和a7。

格式a6设置如下，其中需要格式a和Char2。

//footnote text

m\_ctstyle = new CT\_Style();

m\_ctstyle.type = ST\_StyleType.paragraph;

m\_ctstyle.customStyle = ST\_OnOff.True;

m\_ctstyle.styleId = "a6";

m\_ctstyle.name = new CT\_String();

m\_ctstyle.name.val = "footnote text";

m\_ctstyle.basedOn = new CT\_String();

m\_ctstyle.basedOn.val = "a";

m\_ctstyle.link = new CT\_String();

m\_ctstyle.link.val = "Char2";

m\_ctstyle.uiPriority = new CT\_DecimalNumber();

m\_ctstyle.uiPriority.val = "99";

m\_ctstyle.semiHidden = new CT\_OnOff();

m\_ctstyle.semiHidden.val = true;

m\_ctstyle.unhideWhenUsed = new CT\_OnOff();

m\_ctstyle.unhideWhenUsed.val = true;

m\_ctstyle.rsid = new CT\_LongHexNumber();

byte[] m\_bytefootnoteText = { 0x00, 0xF0, 0x43, 0x96 };

m\_ctstyle.rsid.val = m\_bytefootnoteText;

m\_ctstyle.pPr = new CT\_PPr();

m\_ctstyle.pPr.snapToGrid = new CT\_OnOff();

m\_ctstyle.pPr.snapToGrid.val = false;

m\_ctstyle.pPr.jc = new CT\_Jc();

m\_ctstyle.pPr.jc.val = ST\_Jc.left;

m\_ctstyle.rPr = new CT\_RPr();

m\_ctstyle.rPr.sz = new CT\_HpsMeasure();

m\_ctstyle.rPr.sz.val = 18;

m\_ctstyle.rPr.szCs = new CT\_HpsMeasure();

m\_ctstyle.rPr.szCs.val = 18;

m\_ctstyles.style.Add(m\_ctstyle);

格式a7设置如下，其中需要格式a0。

//footnote reference

m\_ctstyle = new CT\_Style();

m\_ctstyle.type = ST\_StyleType.character;

m\_ctstyle.styleId = "a7";

m\_ctstyle.name = new CT\_String();

m\_ctstyle.name.val = "footnote reference";

m\_ctstyle.basedOn = new CT\_String();

m\_ctstyle.basedOn.val = "a0";

m\_ctstyle.uiPriority = new CT\_DecimalNumber();

m\_ctstyle.uiPriority.val = "99";

m\_ctstyle.semiHidden = new CT\_OnOff();

m\_ctstyle.semiHidden.val = true;

m\_ctstyle.unhideWhenUsed = new CT\_OnOff();

m\_ctstyle.unhideWhenUsed.val = true;

m\_ctstyle.rsid = new CT\_LongHexNumber();

m\_ctstyle.rsid.val = m\_bytefootnoteText;

m\_ctstyle.rPr = new CT\_RPr();

m\_ctstyle.rPr.vertAlign = new CT\_VerticalAlignRun();

m\_ctstyle.rPr.vertAlign.val = ST\_VerticalAlignRun.superscript;

m\_ctstyles.style.Add(m\_ctstyle);

把格式添加到格式文件中：m\_styles.SetStyles(m\_ctstyles);

1. 脚注
2. 创建脚注内容

实际上脚注内容的格式就是引用前面所述的格式设置中的定义，即格式a6和a7。

创建脚注内容文件：XWPFFootnotes m\_ftns = m\_Docx.CreateFootnotes()。

//创建脚注内容

int Id = m\_ftns.GetFootnotesList().Count;

CT\_FtnEdn m\_ftnedn = new CT\_FtnEdn();

m\_ftnedn.id = Id.ToString();

CT\_P m\_FtnEdnxmlP = m\_ftnedn.AddNewP();

CT\_PPr m\_FtnEdnxmlPPr = m\_FtnEdnxmlP.AddNewPPr();

m\_FtnEdnxmlPPr.AddNewPStyle().val = "a6";

m\_FtnEdnxmlPPr.AddNewRPr().rFonts = new CT\_Fonts();

m\_FtnEdnxmlPPr.AddNewRPr().rFonts.hint = ST\_Hint.eastAsia;

CT\_R m\_FtnEdnxmlR = m\_FtnEdnxmlP.AddNewR();

m\_FtnEdnxmlR.AddNewRPr().rStyle = new CT\_String();

m\_FtnEdnxmlR.AddNewRPr().rStyle.val = "a7";

m\_FtnEdnxmlR.Items = new System.Collections.ArrayList();

m\_FtnEdnxmlR.Items.Add(new CT\_Empty());

m\_FtnEdnxmlR.ItemsElementName = new List<RunItemsChoiceType>();

m\_FtnEdnxmlR.ItemsElementName.Add(RunItemsChoiceType.footnoteRef);

m\_FtnEdnxmlR = m\_FtnEdnxmlP.AddNewR();

m\_FtnEdnxmlR.AddNewT().Value = " ";

m\_FtnEdnxmlR = m\_FtnEdnxmlP.AddNewR();

m\_FtnEdnxmlR.AddNewT().Value = strFtnEdn; //"脚注test内容

XWPFFootnote m\_fn = m\_ftns.AddFootnote(m\_ftnedn);

1. 在正文中标注

最好用CT\_P m\_p = m\_Docx.Document.body.AddNewP();方式创建段，在m\_p中可以不断创建CT\_R。

CT\_R m\_r = m\_p.AddNewR();

m\_r.AddNewT().Value = "NPOI";

//标注脚注

CT\_R m\_FtnEdnR = m\_p.AddNewR();

m\_FtnEdnR.AddNewRPr().rStyle = new CT\_String();

m\_FtnEdnR.AddNewRPr().rStyle.val = "a7";

m\_FtnEdnR.Items = new System.Collections.ArrayList();

CT\_FtnEdnRef m\_ftnref = new CT\_FtnEdnRef();

m\_ftnref.id = m\_FtnId;//创建脚注内容得到的Id

m\_FtnEdnR.Items.Add(m\_ftnref);

m\_FtnEdnR.ItemsElementName = new List<RunItemsChoiceType>();

m\_FtnEdnR.ItemsElementName.Add(RunItemsChoiceType.footnoteReference);

m\_r = m\_p.AddNewR();

m\_r.AddNewT().Value = "……";

1. 尾注

NPOI中的OpenXmlFormats提供了较为完善的尾注所有功能，但在XWPF中没有提供创建尾注的方法。

1. 超链接书签

利用NPOI创建超链接书签分两个步骤。一是创建与书签关联的超链接；二是创建书签。

* 1. 创建与书签关联的超链接

NPOI提供两种超链接，一种是超链接到另一文件；另一种是超链接到书签。下面仅介绍创建超链接到书签的方法。

创建文档：XWPFDocument m\_Docx = new XWPFDocument();

创建段落：CT\_P m\_p = m\_Docx.Document.body.AddNewP();

创建超链接集合：m\_p.Items = new System.Collections.ArrayList();

创建超链接：

CT\_Hyperlink1 m\_hyperlink = new CT\_Hyperlink1();

m\_hyperlink.anchor = "NPOI1";//书签名

m\_hyperlink.history = ST\_OnOff.True;

m\_hyperlink.Items = new System.Collections.ArrayList();

CT\_R m\_r = new CT\_R();

m\_r.AddNewT().Value = "书签1";

m\_hyperlink.Items.Add(m\_r);

m\_hyperlink.ItemsElementName = new List<ItemsChoiceType12>();

m\_hyperlink.ItemsElementName.Add(ItemsChoiceType12.hyperlink);

m\_p.Items.Add(m\_hyperlink);

* 1. 创建书签

书签分开始和结束两部分组成。

//书签0开始

int m\_bookId = 0;//同一段内有多个书签，需要不同的Id，不同段的书签Id可以相同

m\_p = m\_Docx.Document.body.AddNewP();

m\_p.AddNewPPr().AddNewJc().val = ST\_Jc.both;

m\_p.AddNewPPr().AddNewSpacing().line = "400";//固定行距20磅

m\_p.AddNewPPr().AddNewSpacing().lineRule = ST\_LineSpacingRule.exact;

m\_p.Items = new System.Collections.ArrayList();

CT\_Bookmark m\_ctbook1 = new CT\_Bookmark();

m\_bookId = m\_p.Items.Count;

m\_ctbook1.id = m\_bookId.ToString(); //"0";

m\_ctbook1.name = "NPOI1";//书签名，超链接用

m\_p.Items.Add(m\_ctbook1);

m\_p.ItemsElementName = new List<ParagraphItemsChoiceType>();

m\_p.ItemsElementName.Add(ParagraphItemsChoiceType.bookmarkStart);

m\_p.AddNewR().AddNewT().Value = "1、NPOI介绍";

//书签0结束

m\_ctbook1 = new CT\_Bookmark();

m\_ctbook1.id = m\_bookId.ToString();//"0";

m\_p.Items.Add(m\_ctbook1);

m\_p.ItemsElementName.Add(ParagraphItemsChoiceType.bookmarkEnd);

1. 插入图表

在docx中插入图表分三步实现。一是创建xlsx格式的图表原始数据，二是创建图表类型，三是在正文中插入图表。每一个图表对应一个xlsx文件，下面以饼图为例说明。

1. 创建xlsx格式的图表原始数据

//创建xlsx

          XSSFWorkbook workbook = new XSSFWorkbook();

  //创建表单1（饼图）

            I Sheet sheet = workbook.CreateSheet("Sheet1");

            //表单1饼图数据

//销售额

            //第一季度 8.2

            //第二季度 3.2

            //第三季度 1.4

            //第四季度 1.2

            IRow row = sheet.CreateRow(0);

            ICell cell = row.CreateCell(0);

            cell = row.CreateCell(0);

            cell = row.CreateCell(1);

            cell.SetCellValue("销售额");

            row = sheet.CreateRow(1);

            cell = row.CreateCell(0);

            cell.SetCellValue("第一季度");

            cell = row.CreateCell(1);

            cell.SetCellValue(8.2);

            row = sheet.CreateRow(2);

            cell = row.CreateCell(0);

            cell.SetCellValue("第二季度");

            cell = row.CreateCell(1);

            cell.SetCellValue(3.2);

            row = sheet.CreateRow(3);

            cell = row.CreateCell(0);

            cell.SetCellValue("第三季度");

            cell = row.CreateCell(1);

            cell.SetCellValue(1.4);

            row = sheet.CreateRow(4);

            cell = row.CreateCell(0);

            cell.SetCellValue("第四季度");

            cell = row.CreateCell(1);

            cell.SetCellValue(1.2);

1. 创建图表类型

//创建\word\charts\chartn.xml内容（简单饼图）

            CT\_ChartSpace ctpiechartspace = new CT\_ChartSpace();

            ctpiechartspace.date1904 = new CT\_Boolean();

            ctpiechartspace.date1904.val = 1;

            ctpiechartspace.lang = new CT\_TextLanguageID();

            ctpiechartspace.lang.val = "zh-CN";

            CT\_Chart m\_chart = ctpiechartspace.AddNewChart();

            m\_chart.plotArea = new CT\_PlotArea();

            m\_chart.plotArea.pieChart = new List<CT\_PieChart>();

            //饼图

            CT\_PieChart m\_piechart = new CT\_PieChart();

            m\_piechart.varyColors = new CT\_Boolean();

            m\_piechart.varyColors.val = 1;

            m\_piechart.ser = new List<CT\_PieSer>();

            CT\_PieSer m\_pieser = new CT\_PieSer();

            //标题

            m\_pieser.tx = new CT\_SerTx();

            m\_pieser.tx.strRef = new CT\_StrRef();

            m\_pieser.tx.strRef.f = "Sheet1!$B$1";

            m\_pieser.tx.strRef.strCache = new CT\_StrData();

            m\_pieser.tx.strRef.strCache.ptCount = new CT\_UnsignedInt();

            m\_pieser.tx.strRef.strCache.ptCount.val = 1;

            CT\_StrVal m\_strval = new CT\_StrVal();

            m\_strval.idx = 0;

            m\_strval.v = "销售额";

            m\_pieser.tx.strRef.strCache.pt = new List<CT\_StrVal>();

            m\_pieser.tx.strRef.strCache.pt.Add(m\_strval);

            //行标题

            m\_pieser.cat = new CT\_AxDataSource();

            m\_pieser.cat.strRef = new CT\_StrRef();

            m\_pieser.cat.strRef.f = "Sheet1!$A$2:$A$5";

            m\_pieser.cat.strRef.strCache = new CT\_StrData();

            m\_pieser.cat.strRef.strCache.ptCount = new CT\_UnsignedInt();

            m\_pieser.cat.strRef.strCache.ptCount.val = 4;

            m\_pieser.cat.strRef.strCache.pt = new List<CT\_StrVal>();

            m\_strval = new CT\_StrVal();

            m\_strval.idx = 0;

            m\_strval.v = "第一季度";

            m\_pieser.cat.strRef.strCache.pt.Add(m\_strval);

            m\_strval = new CT\_StrVal();

            m\_strval.idx = 1;

            m\_strval.v = "第二季度";

            m\_pieser.cat.strRef.strCache.pt.Add(m\_strval);

            m\_strval = new CT\_StrVal();

            m\_strval.idx = 2;

            m\_strval.v = "第三季度";

            m\_pieser.cat.strRef.strCache.pt.Add(m\_strval);

            m\_strval = new CT\_StrVal();

            m\_strval.idx = 3;

            m\_strval.v = "第四季度";

            m\_pieser.cat.strRef.strCache.pt.Add(m\_strval);

            //值

            m\_pieser.val = new CT\_NumDataSource();

            m\_pieser.val.numRef = new CT\_NumRef();

            m\_pieser.val.numRef.f = "Sheet1!$B$2:$B$5";

            m\_pieser.val.numRef.numCache = new CT\_NumData();

            m\_pieser.val.numRef.numCache.formatCode = "General";

            m\_pieser.val.numRef.numCache.ptCount = new CT\_UnsignedInt();

            m\_pieser.val.numRef.numCache.ptCount.val = 4;

            m\_pieser.val.numRef.numCache.pt = new List<CT\_NumVal>();

            CT\_NumVal m\_numval = new CT\_NumVal();

            m\_numval.idx = 0;

            m\_numval.v = "8.2";

            m\_pieser.val.numRef.numCache.pt.Add(m\_numval);

            m\_numval = new CT\_NumVal();

            m\_numval.idx = 1;

            m\_numval.v = "3.2";

            m\_pieser.val.numRef.numCache.pt.Add(m\_numval);

            m\_numval = new CT\_NumVal();

            m\_numval.idx = 2;

            m\_numval.v = "1.4";

            m\_pieser.val.numRef.numCache.pt.Add(m\_numval);

            m\_numval = new CT\_NumVal();

            m\_numval.idx = 3;

            m\_numval.v = "1.2";

            m\_pieser.val.numRef.numCache.pt.Add(m\_numval);

            m\_piechart.ser.Add(m\_pieser);

            m\_chart.plotArea.pieChart.Add(m\_piechart);

            m\_chart.legend = new CT\_Legend();

            m\_chart.legend.legendPos = new CT\_LegendPos();

            m\_chart.legend.legendPos.val = ST\_LegendPos.r;

            m\_chart.plotVisOnly = new CT\_Boolean();

            m\_chart.plotVisOnly.val = 1;

1. 页面中插入图表

以inline式为例。

            XWPFParagraph gp = m\_Docx.CreateParagraph();

            XWPFRun gr = gp.CreateRun();

            gp = m\_Docx.CreateParagraph();

            gr = gp.CreateRun();

            gr.AddChartSpace(workbook , ctpiechartspace, 5274310, 3076575);

1. 其它