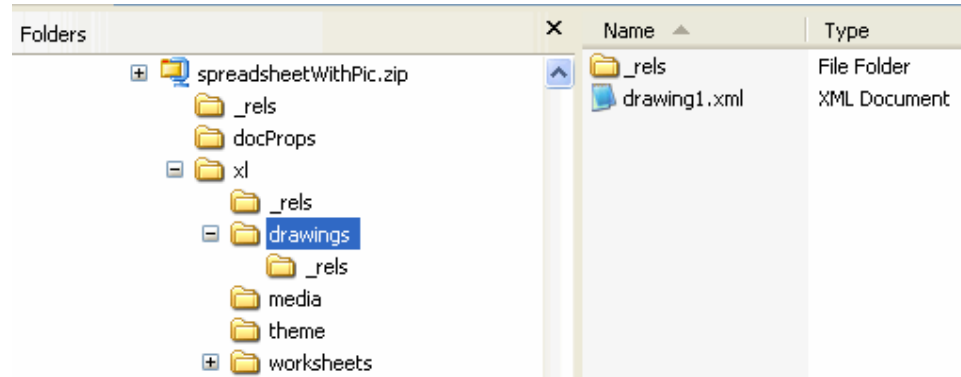


Shape Export in OOXML Export

1. Document structure in OOOXML

All drawing objects for a worksheet are within one drawing part. There are as many drawing parts as there are worksheets containing drawing objects. (Note that the actual data for, e.g., an image, is stored separately, typically within a media sub-folder of the package--for Word, within a media sub-folder.)

File structure for drawing in spreadsheet



The worksheet part will contain a <drawing> element with a relationship ID attribute referencing a relationship.

```
<worksheet ...>
...
<drawing r:id="rId1"/>
...
</worksheet>
```

The relationships (.rels) part for the worksheet contains the relationship referencing the target drawing part.

```
<Relationships ...>
<Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/drawing" Target="../drawings/drawing1.xml" />
</Relationships>
```

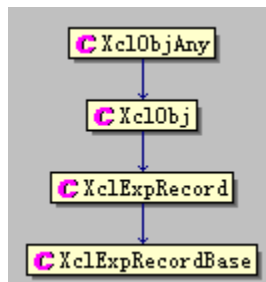
Within the drawings part (drawing1.xml) the container element is a root <xdr:wsDr> element. The namespace for pictures within a worksheet is: [xlm:sxdr="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing"](http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing). There are three possible child elements representing three different ways that an object can be anchored within a worksheet.

<wp:absoluteAnchor> - used to anchor the object absolutely in the same position in the sheet; does not move or size with cells
 <wp:oneCellAnchor> - used to anchor to a single cell; moves with cells
 <wp:twoCellAnchor> - used to anchor to two cells; moves with cells

```
<xdr:wsDr xlm:sxdr="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDrawing" xlm:a="http://schemas.openxmlformats.org/drawingml/2006/main">
  <xdr:twoCellAnchor editAs="oneCell">
    ...
  </xdr:twoCellAnchor>
</xdr:wsDr>
```

2. Implement

the class of XclObjAny is used to export shape of OOOXML.



```
class XclObjAny : public XclObj
{
private:
com::sun::star::uno::Reference< com::sun::star::drawing::XShape > mxShape;
virtual void WriteSubRecs( XclExpStream& rStrm );
```

```

public:
XclObjAny( XclExpObjectManager& rObjMgr );
XclObjAny( XclExpObjectManager& rObjMgr ,
com::sun::star::uno::Reference< com::sun::star::drawing::XShape > xShape);
virtual ~XclObjAny();

virtual void Save( XclExpStream& rStrm );
void SaveXml( XclExpXmlStream& rStrm );
com::sun::star::uno::Reference< com::sun::star::drawing::XShape > GetShape() const { return mxShape; }
static void WriteFromTo( XclExpXmlStream& rStrm, const XclObjAny& rObj );
static void WriteFromTo( XclExpXmlStream& rStrm, const com::sun::star::uno::Reference< com::sun::star::drawing::XShape >& rShape, SCTAB
nTab );
};

```

the function of SaveXml is used to execute the exporting process.