

xml

xml

WorkBookIOFileAttrMark

xmlcpt

xml
<?xml version="1.0" encoding="UTF-8"?> <WorkBook xmlVersion="20151125" releaseVersion="8.0.0"> <Report class="com.fr.report.worksheet.WorkSheet" name="sheet1"> <ReportPageAttr> <HR/> <FR/> <HC/> <FC/> </ReportPageAttr> <ColumnPrivilegeControl/> <RowPrivilegeControl/> <RowHeight defaultValue="723900"> <![CDATA [723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900, , 723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900, 723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900,723900, 723900,723900,723900]]></RowHeight> <ColumnWidth defaultValue="2743200"> <![CDATA [2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200, ,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200,2743200, ,2743200,2743200,2743200]]></ColumnWidth> <CellElementList> <C c="0" r="0"> <PrivilegeControl/> <Expand/> </C> </CellElementList> <ReportAttrSet> <ReportSettings headerHeight="0" footerHeight="0"> <PaperSetting/> </ReportSettings> </ReportAttrSet> <PrivilegeControl/> </Report> <StyleList/> <DesignerVersion DesignerVersion="IAA"/> <PreviewType PreviewType="0"/> </WorkBook>

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkBook xmlVersion="20151125" releaseVersion="8.0.0">
<Report class="com.fr.report.worksheet.WorkSheet" name="sheet1">
<DesignerVersion DesignerVersion="IAA"/>
<PreviewType PreviewType="0"/>
</WorkBook>
```

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkBook xmlVersion="20151125" releaseVersion="8.0.0">
<Report class="com.fr.report.worksheet.WorkSheet" name="sheet1">
<DesignerVersion DesignerVersion="IAA"/>
<PreviewType PreviewType="0"/>
<SomeConfig class="com.fr.plugin.io.file.SomeConfig">
<Attr status="100"/>
</SomeConfig>
</WorkBook>
```

SomeConfigwriteXML

xml

```
public void writeConfig(XMLPrintWriter writer, SomeConfig config) {
    GeneralXMLTools.writeXMLLabel(writer, config, "SomeConfig");
}
```

xmlWorkBookSomeConfig

xml

```
protected void readConfig(XMLLabelReader reader) {
    if (reader.isChildNode()) {
        String tagName = reader.getTagName();
        if (tagName.equals("SomeConfig")) {
            SomeConfig config = new SomeConfig();
            reader.readXMLObject(config);
            this.addAttrMark(config);
        }
    }
}
```

reader.readXMLObject(config);xmlconfig

WorkBookSomeConfigSomeConfig

SomeConfig

```
public class SomeConfig extends AbstractIOFileAttrMark {

    private int status = 0;

    public SomeConfig() {

    }

    public SomeConfig(int status) {
        this.status = status;
    }

    public int getStatus() {
        return status;
    }

    @Override
    public String xmlTag() {
        return "SomeConfig";
    }

    @Override
    public SomeConfig clone() {
        SomeConfig cloned = (SomeConfig) super.clone();
        cloned.status = status;
        return cloned;
    }

    @Override
    public void readXML(XMLableReader reader) {
        if (reader.isChildNode()) {
            String tagName = reader.getTagName();
            if ("Attr".equals(tagName)) {
                status = reader.getAttrAsInt("status", 0);
            }
        }
    }

    @Override
    public void writeXML(XMLPrintWriter writer) {
        writer.startTAG("Attr");
        writer.attr("status", status);
        writer.end();
    }
}
```

SomeConfigXMLablexmlXMLableXMLPrintWriterstartTAGXMLPrintWriterendwriteXMLxmlSomeConfigAttr

```
<SomeConfig class="com.fr.plugin.io.file.SomeConfig">
<Attr status="100"/>
</SomeConfig>
```

xmlAttr