unoofc Reference

Willis L. Boughton
Apr 2026

unoofc is a Java class framework that enables building, writing, and reading LibreOffice Writer, Calc, Impress, Draw, and chart documents without any knowledge of the LibreOffice Java API. unoofc encapsulates LibreOffice API functionality. This document is a reference for unoofc. Section 1 is for programmers who are familiar with the LibreOffice API and for programmers who are not. Section 2 is for programmers who are familiar with the API. It is assumed the LibreOffice SDK is installed.

Section 1

Class Summary

Table 1 summarizes unoofc classes. All classes are in package unoofc. In many situations, the only import necessary for application code is unoofc. Class details are given in the unoofc API.

Table 1. Classes

Group	*Name**	*Description*
Document	*ORootDoc*	Encapsulates behavior and state common to Writer, Impress, Calc, and Draw documents. This class is the parent of OCalcDoc, ODrawDoc, OImpressDoc, and OWriterDoc.
	OCalcDoc	Encapsulates a Calc document. This class has nested class OCalcDoc.Sheetthat encapsulates a Calc sheet.
	ODrawDoc	Encapsulates a Draw document. This class has multiple nested classes for Draw components, e.g., ODrawDoc.Layer and ODrawDoc.LineShape.
	OImpressDoc	Enacapsulates an Impress document. This class has multiple nested classes for Impress components, e.g., OImpressDoc.Slide and OImpressDoc.Bullet.
	OWriterDoc	Enacapsulates a Writer document. This class has multiple nested classes, e.g., OWriterDoc.Field and OWriterDoc.WordCursor.
	OChartDoc	Enacapsulates a Calc chart. This class has multiple nested classes, e.g., OChartDoc.Axis and OChartDoc.Legend.
Support	OColor	Represents a color.
	ODrawArea	Represents a drawing area defined by a point and a size.
	OEnumPropObj	Represents an object with enumerated properties, and provides query and modify methods for these properties.
	OFileFilter	Encapsulates read/write document filters.
	OLogger	Represents a logging output. Nested classes represent types of loggers, e.g., OLogger.System for the console.

	ONumbering	Encapsulates numbering rules for styles.
	OPresentation	Encapsulates an OImpressDoc viewing.
	OProps	Definesan enumeration type for each type of object property, such as properties for a line.
	OShape	Encapsulates behavior and state common to all Draw shapes.
	OStyle	Encapsulates behavior and state common to all document styles. This class has multiple concrete nested classes, one for each type of style, e.g., OStyle.Char.
	OTable	Encapsulates a row and column text table. Nested class OTable.Cell encapsulates a text table cell or spreadsheet cell.
	OUtil	Provides (static) utility functions and concrete nested utility classes.
Other	OEnv	Encapsulates the LibreOffice (UNO) run-time environment as a singleton object.
* Italics denotes an abstract class.		

Object Properties

All unoofc classes that extend class *OEnumPropObj* have object properties that can be queried and modified using the enumeration types in class *OProps*. These enumerations provide compile-time type checking of property value validity. For each enumeration type, *OEnumPropObj* provides methods of the form:

```
Object prop(OProps.Type prop); // query

void setProp(OProps.Type type, Object value); // modify
```

where *OProps.Type* is the enumeration type. For example, the enumeration for a line property is

```
public enum Line
   {LineStyle, LineDash, LineDashName, LineColor, LineTransparence,
   LineWidth,LineJoint, LineCap, LineStartName, LineEndName, LineStart,
   LineEnd, LineStartCenter, LineStartWidth, LineEndCenter,
   LineEndWidth};
```

The code to query the LineWidth of object anObject would be:

```
int width = (Integer) anObject.prop(OProps.Line.LineWidth);
```

and the code to set the LineWidth of the object to 10 would be:

```
anObject.setProp(OProps.Line.LineWidth, 10);
```

The unoofc demonstration programs show examples of property usage.

Logging

The OEnv and OLogger classes provide execution logging functionality. By default, logging to the

console is enabled. The code to write the message "started" to the console would be:

```
OEnv.obj().log("started", false);
```

`OEnv.obj()` obtains the run-time singleton object, and `false` indicates the message is not an error. The code to set the logger to text file "logger.txt", not appending, would be:

```
OEnv.obj().setLogger(new OLogger.File("logger.txt", false));
```

Imports

Some `unoofc` functionality requires a constant, enumeration value, or data structure defined in the LibreOffice API. An example is API data structure `com.sun.star.awt.Point`. In the distribution demonstration programs, these cases are indicated by use of the fully qualified class name from the API. The demonstration programs use no import statements for the LibreOffice API since `unoofc` encapsulates API functionality.

Demonstration Programs

File `demo.zip` contains demonstration programs and files required by them. There is one program for Writer, one for Calc, one for Draw, and one for Impress. Each program shows "integrated" use of `unoofc` rather than just code snippets and uses most but not necessarily all `unoofc` functionality. To setup the demonstration programs, unzip `demo.zip` into a folder. To run a demonstration program, compile and run it with the unzipped `demo.zip` folder as the only argument and `unoofc.jar` in the classpath. The program will write an output document to that folder and write logging information to the console.

Section 2

This section is intended only for programmers who are familiar with the LibreOffice API and want to use API functionality along with `unoofc`.

UNO Run-time

The UNO runtime is started the first time the `OEnv` singleton object is accessed. This happens automatically, transparently, the first time a document is created or opened.

Properties

`unoofc` does not provide query or modification of `XInterface` object properties using text strings as property names. The *`OProps`* enumeration types must be used. To use text string names, the encapsulated `XInterface` objects must be used.

Functionality

`unoofc` provides much basic LibreOffice API functionality but not all. The best way to see the functionality is by examining and running the demonstration programs.

Interface Object Access

Each `unoofc` encapsulation class provides access to its encapsulated `XInterface` object or objects, so an application can use these objects directly. Table 2 lists the encapsulations.

Table 2. Class Encapsulation

Class	*Encapsulated XInterface Object(s)*
OCalcDoc	XSpreadsheetDocument
OWriterDoc	XTextDocument XTextCursor
OChartDoc	XChartDocument
OColor	none
ODrawArea	none
OEnumPropObj	none
OFileFilter	none
OLogger	none
ONumbering	none
OPresentation	XPresentation
OProps	none
OShape	XShape
OStyle	XStyle
ORootDoc	XComponent XMultiServiceFactory
OTable	XTextTable
OUtil	none
OEnv	XComponentContext XComponentLoader