unoofc Reference

Willis L. Boughton
August 2026

---

unoofc is a Java package that enables building, writing, and reading LibreOffice Writer, Calc, Impress, Draw, and chart documents without any knowledge of the LibreOffice Java API or the UNO XInterface API. The unoofc classes encapsulate LibreOffice API functionality.

License

unoofc is distributed as freeware. It can be used on any number of computers by any number of users and may be distributed freely. unoofc is provided "as is", and any warranties of any kind, express or implied, for any use or intended use, on any computing system, are disclaimed, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. In no event shall the software provider be liable for any damages of any form arising in any way out of the use of this software, even if advised of the possibility of such damage. By using unoofc, you acknowledge that you have read these terms, understand them, and agree to them.

Requirements

The LibreOffice SDK must have been installed.

Distribution

unoofc is distributed in Zip file unoofc-distrib.zip, which contains the following:

- JAR file unoofc.jar, a non-executable JAR containing all unoofc class files. unoofc.jar must be put in the Java classpath.
- Package unoofc.demo, which contains documented 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 all unoofc functionality. These program must be run from the unoofo.demo directory, and each program writes an output document to that directory and information to the console.

javadoc for the unoofc API are at http://www.bluemontsw.com/unoofc/api/.

Getting Started

To get started with unoofc:

- Study this document.
- Study and run the four example programs described above, and study the javadoc.

Class Summary

Table 1 summarizes `unoofc` classes. All classes are in package `unoofc`, and italics denotes an abstract class.

Table 1. Classes		
Group	*Name**	*Description*
Document	*ORootDoc*	The parent of `OCalcDoc`, `ODrawDoc`, `OImpressDoc`, and `OWriterDoc`.
	`OCalcDoc`	A Calc document. This class has nested class `OCalcDoc.Sheet` that is a Calc sheet.
	`ODrawDoc`	A Draw document. This class has multiple nested classes for Draw components, e.g., `ODrawDoc.Layer` and `ODrawDoc.LineShape`.
	`OImpressDoc`	An Impress document. This class has multiple nested classes for Impress components, e.g., `OImpressDoc.Slide` and `OImpressDoc.Bullet`.
	`OWriterDoc`	A Writer document. This class has multiple nested classes, e.g., `OWriterDoc.Field` and `OWriterDoc.WordCursor`.
	`OChartDoc`	A Calc chart. This class has multiple nested classes, e.g., `OChartDoc.Axis` and `OChartDoc.Legend`.
Support	`OColor`	A UNO-compatible color.
	`ODrawArea`	A drawing area defined by a point and a size.
	OEnumPropObj	An object with UNO enumerated properties, each with query and modify methods.
	`OFileFilter`	A document read/write filter.
	OLogger	A logger. Nested classes represent types of loggers, e.g., `OLogger.System` for the console.
	`ONumbering`	Numbering rules for styles.
	`OPresentation`	An `OImpressDoc` viewing.
	OProps	Enumeration types for UNO properties, such as for a line.
	OShape	A Draw shape.
	OStyle	A document style. This class has multiple nested classes, one for each type of style, e.g., `OStyle.Char`.
	`OTable`	A row and column text table. Nested class `OTable.Cell` is a text table cell or spreadsheet cell.
	OUtil	Provides (static) utility functions and classes.
Other	`OEnv`	The LibreOffice (UNO) run-time environment, a singleton object.

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));
```

UNO Use

- 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.
- `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. See Table 2.
- 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.

- `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.
- 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