

Redactado con LibreOffice, v. 5.3.3 – Plataforma: Todas

Manipulación de archivos y directorios

Mediante las instrucciones nativas

Dir()	Devuelve el nombre de un archivo o directorio o también de todos los archivos y directorios existentes en una unidad o directorio correspondiente a la ruta especificada.
FileCopy()	Copia un archivo.
FileDateTime()	Devuelve una cadena de caracteres que contiene la fecha y hora de creación o última modificación de un archivo.
FileExists()	Devuelve True si existe el archivo o directorio.
FileLen()	Devuelve el tamaño de un archivo (en bytes).
GetAttr()	Devuelve el tipo de archivo, unidad o directorio.
GetPathSeparator()	Devuelve el separador de ruta usado en el sistema operativo.
Kill()	Elimina un archivo de la unidad.
MkDir()	Crea un directorio nuevo.
Name()	Renombra un archivo o directorio existente
Rmdir()	Elimina un directorio existente
SetAttr()	Define los atributos del archivo indicado.

Mediante un objeto SimpleFileAccess

Métodos del servicio com.sun.star.ucb.SimpleFileAccess:

```
oSFA = createUNOService("com.sun.star.ucb.SimpleFileAccess")
```

copy	Copia un archivo.
createFolder	Crea un directorio nuevo.
exists	Verifica si un archivo o directorio existe.
getContent	Devuelve el tipo de contenido del archivo.
getDateModified	Devuelve la fecha de la última modificación del archivo.
getFolderContents	Devuelve el contenido de un directorio.
getSize	Devuelve el tamaño de un archivo.
isFolder	Verifica si una URL es un directorio.
isReadOnly	Verifica si un archivo es de solo lectura.
kill	Elimina un archivo o directorio. Incluso elimina un directorio aunque no esté vacío.
move	Mueve un archivo.
setInteractionHandler	Establece un controlador de interacción para operaciones.
setReadOnly	Activa el indicador de sólo lectura de un archivo (se necesitan los permisos adecuados).

Rutas de los archivos

Por ser multiplataforma, las rutas de los archivos se expresan en formato URL:
 file:///soporte/Ruta/a/l/archivo.ext

Funciones de conversión

De nativo a URL	NombreUrl = ConvertToURL(NombreArchivoNativo)
De URL a nativo	NombreOS = ConvertFromURL(NombreArchivoURL)

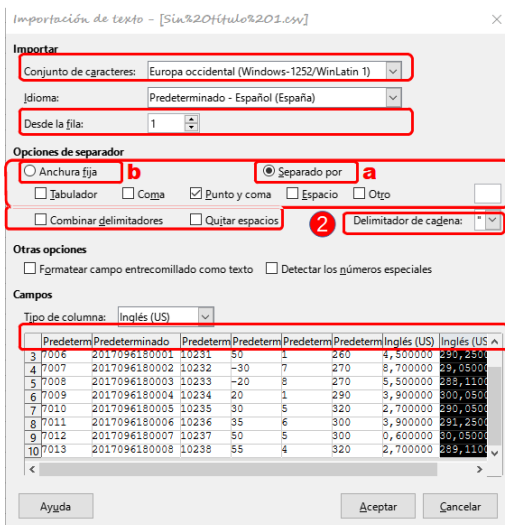
Importación de archivos de texto (CSV)

Manualmente

Diálogo de importación de Calc:

Elementos de los filtros CSV

Los filtros CSV necesitan 5 parámetros (Tome como referencia el diálogo mostrado).



Separador de campo

a. Formato variable (separado por)

Código Ascii del separador (ver tabla Ascii).	"9"
Si hay alternativas sepárelas mediante /	"9/36"
Si son varios caracteres fusionados agregue /MRG	"36/36/MRG"
b. Formato fijo (Anchura fija)	"FIX"

Delimitador de cadena

Código Ascii del delimitador (vea tabla Ascii) "34" (" si no hay ninguno)

conjunto de caracteres

Conjunto de caracteres más usuales (si se indica "" se asume que es UTF-8):

Windows-1252/WinLatin1	ANSI	ISO-8859-15/EURO	22
ISO-8859-1	12	UTF-8	76

Lista actualizada de conjuntos: servicio "com.sun.star.document.FilterFactory"

Primera línea de datos del archivo a importar (desde la fila)

Número de la línea (empezando por 1) "2" ("1" o "" si es la 1ª línea)

Formato de las columnas (tipo de columna)

- a. Variable** Por cada columna secuencia de 4 caracteres.: "1/2/ 2/2/ 3/1/"
rang (base 1) / format / (" si solo utiliza valores predeterminados) (vea la tabla Formatos)
- b. Fijo** Por cada columna secuencia de 4 car.: "0/1/ 8/2/ 13/1/"
pos. 1º car (base 0) / format / (vea la tabla Formatos)

☞ Puede intercalar espacios para hacer la secuencia más legible.

☞ Puede especificar solamente las columnas útiles.

Filtro de importación CSV

El filtro está formado por la concatenación de sus 5 parámetros separados por comas: Filtro = "9,34,76,1,1/2/2/2/3/1"

☞ Según el contexto, determinados valores se pueden omitir (vea detalles de los parámetros): Filtre = "59,,76,,,"

Información adicional

Códigos Ascii más frecuentes (notación decimal)

9	Tabulador	34 "	36 \$	44 ,	59 ;
32	Espacio	35 #	39 '	58 :	

Formato de las columnas

1	Predeterminado (automático Calc)	5	YY/MM/DD
2	Texto	9	(descartar columna (ocultar)
3	MM/DD/YY	10	Formato US (sep. decimales y millares)
4	DD/MM/YY		

Importar un archivo CSV en una hoja de cálculo Calc

Se quiere copiar el contenido de un archivo CSV con nombre MiArchivo.csv en la hoja con nombre NombreHoja del libro en uso.

El proceso se efectúa mediante el empleo de una memoria intermedia (buffer) y después la eliminación de la misma.

```
Dim CsvURL As String 'ruta del archivo .csv origen
Dim Filtro As String
Dim Hoja As Object 'hoja destino en el libro

Hoja = ThisComponent.Sheets.getByIndex("Nombre_de_la_Hoja")
CsvURL = ConvertToURL("C:\ruta\MiArchivo.csv")
'opciones de lectura del archivo .csv
Filtro = "9,34,ANSI,1,1/2/ 2/2/ 3/1/ 4/1/"
'importación mediante un buffer entre la hoja y el origen .csv
Hoja.link(CsvURL, "", "Text - txt - csv (StarCalc)", _
    Filtro, com.sun.star.sheet.SheetLinkMode.VALUE)
'liberación del buffer para volver el libro autonomo
Hoja.setLinkMode(com.sun.star.sheet.SheetLinkMode.NONE)
```

☞ El contenido anterior de la hoja se elimina sin previo aviso.

Crear un libro Calc partiendo de un archivo CSV

Se quiere crear un libro Calc a partir de un archivo CSV con nombre MiArchivo.csv.

```
Dim props1(1) As New com.sun.star.beans.PropertyValue
Dim props2()
Dim CsvURL As String 'ruta del archivo .csv origen
Dim DocURL As String 'ruta del archivo .ods destino
Dim oDoc As Object 'Libro destino

CsvURL = ConvertToURL("C:\ruta\MiArchivo.csv")
'opciones de lectura del archivo .csv
props1(0).Name = "FilterName"
props1(0).Value = "Text - txt - csv (StarCalc)"
props1(1).Name = "FilterOptions"
props1(1).Value = "9,34,ANSI,1,1/2/ 2/2/ 3/1/ 4/1/"
'carga del archivo origen .csv en la primera hoja
oDoc = StarDesktop.loadComponentFromURL(CsvURL, "_blank", 0,
    props1())
'guardado del libro en formato .ods
DocURL = ConvertToURL("C:\ruta\MiLibro.ods")
oDoc.storeAsURL(DocURL, props2())
```

☞ El documento creado contendrá sólo una hoja que toma nombre del origen CSV.

☞ En el ejemplo, el documento creado se muestra en pantalla. Para evitar esto:
 – utilice la opción Hidden (valor True) en props1()
 – agregue oDoc.close(True) al final del proceso.

☞ Si existe un documento Calc con el mismo nombre se eliminará sin previo aviso.

Gestión de contenidos – Instrucciones nativas

Pasos del proceso

1. **Obtener** el n.º identificativo interno del archivo con (FreeFile),
2. **Abrir** el archivo (Open),
3. **Escribir** en archivo (Print, Put o Write) ou **leerlo** (Get, Line Input# ou Input#),
4. **Cerrar** el archivo (Close)

Acceso al contenido de los archivos mediante su n.º identificativo (handle)

Close	Cierra un archivo previamente abierto con Open.
Eof()	Determina si el cursor del archivo ha llegado al final del archivo
FileAttr()	Devuelve el modo de acceso en que el archivo se ha abierto (<i>handle</i>)

FreeFile	Devuelve un n.º <i>handle</i> disponible antes de la apertura de un archivo.
Get	Lee un registro de archivo para insertarlo en una variable.
Input	Lee los datos de un archivo secuencial abierto.
Line Input	Lee una línea de un archivo.
Loc()	Devuelve la posición actual en un archivo abierto.
Lof()	Devuelve el tamaño de un archivo abierto (en bytes)
Open	Abre un canal de datos
Print	Escribe los datos en un archivo secuencial (línea no delimitada).
Put	Escribe un registro en un archivo.
Reset	Cierra todos los archivos abiertos y fuerza la escritura en disco del contenido de la memoria intermedia (buffers) de los archivos.
Seek()	Devuelve la posición de la siguiente escritura o lectura en un archivo abierto con la instrucción Open For Random.
Write	Escribe los datos en un archivo secuencial (línea, delimitador).

¿Print o Write?
Print graba el texto mientras que Write inserta el texto con delimitadores que caracterizan el tipo de información (texto: ", hora y booleano: #). Estos delimitadores se desprecian cuando se leen posteriormente con Input

Modos de apertura (archivos de texto)

Escritura secuencial Open For Output Lectura secuencial Open For Input

Para otros tipos de acceso (acceso directo o binario), utilice el «flujo» de la API.

Ejemplo de lectura secuencial de un archivo de texto

Lectura de un archivo identificado por su dirección URL.

```
Dim Handle As Integer, Linea As String, URLArchivo As String
Handle = FreeFile 'obtiene el n.º identificativo del archivo abierto
Open URLArchivo For Input As #Handle
'bucla para lectura línea a línea
Do While Not Eof(Handle)
    Line Input #Handle, Line 'lectura de cada línea
Loop
Close #Handle
```

Ejemplo de escritura secuencial de un archivo de texto

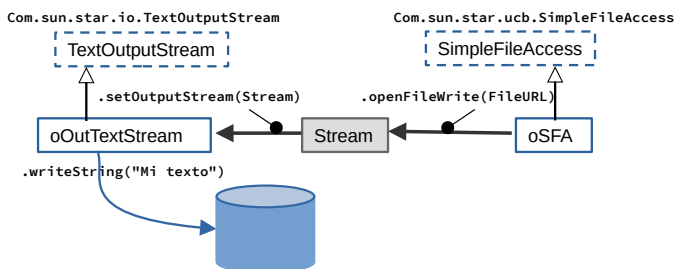
Escritura en un archivo identificado por su dirección URL.

```
Dim Handle As Integer
Handle = FreeFile 'obtiene el n.º identificativo de archivo abierto
Open URLArchivo For Output As #Handle
'escritura línea a línea
Print #Handle, "Un Texto."
Print #Handle, "De nuevo otro texto..."
Print #Handle, "Para acabar."
Close #Handle
```

Gestión de contenidos – «flujo» de la API

Usa la llamada a los servicios SimpleFileAccess y Stream de la API de LibreOffice.

Principio



Ejemplo: escritura de un archivo de texto (vea el código en el ejemplo siguiente).

Pasos del proceso

1. **Crear el objeto** de acceso a los archivos.
oSFA = createUNOService ("com.sun.star.ubc.SimpleFileAccess")
2. **Conectar** el flujo correspondiente al tratamiento (según tipo de acceso).
3. **Escribir** en el archivo o **leerlo** (según el tipo de archivo).
4. En caso de escritura se debe **purgar** el flujo (.flush),
5. **Cerrar** el archivo (.closeXXX).

Acceso al contenido de los archivos

Servicio SimpleFileAccess (SFA)

openFileRead	Abre el archivo en modo lectura.
openFileWrite	Abre el archivo en modo escritura.
openFileReadWrite	Abre el archivo en modo lectura y escritura.

Servicios Flujo (InputStream, OutputStream y Stream)

Son los servicios «activos».

Correspondencia entre los métodos del servicio SFA y los flujos:

Método SFA	Servicio flujo
openFileRead	com.sun.star.io.InputStream
openFileWrite	com.sun.star.io.OutputStream
openFileReadWrite	com.sun.star.io.Stream

Servicio Stream

getInputStream	Devuelve la parte InputStream del flujo lectura/escritura. ⚠ El cierre de este flujo implica la del flujo OutputStream.
getOutputStream	Devuelve la parte OutputStream del flujo lectura/escritura. ⚠ El cierre de este flujo implica la del flujo InputStream.

Servicio InputStream

readBytes	Lee el número especificado de bytes.
readSomeBytes	Lee el número de bytes disponibles con el máximo especificado.

skipBytes	Salta el número especificado de bytes (valor positivo).
available	Indica el n.º de bytes que pueden ser leídos o saltados sin bloqueo
closeInput	Cierra el flujo.

Servicio TextInputStream

Heredado de InputStream.

readLine	Lee el texto hasta encontrar un salto de línea (CR, LF, ou CRLF) o EOF y devuelve la cadena correspondiente (omitiendo CR ou LF). ⚠ Los caracteres leídos se convierten según la codificación especificada por setEncoding. Si se alcanza EOF antes de la ejecución del método devuelve una cadena vacía.
----------	---

readString	Lee el texto hasta encontrar uno de los delimitadores o EOF. Y devuelve la cadena correspondiente. ⚠ CRLF No es el delimitador predeterminado Si no se ha indicado ningún delimitador o no se encuentra, se supone que es hasta EOF. Los caracteres leídos se convierten según la codificación especificada por setEncoding. Si se alcanza EOF antes de la ejecución del método devuelve una cadena vacía..
------------	---

isEOF	Devuelve el estado de EOF. ⚠ Este estado no se puede detectar al intentar leer una cadena vacía porque esta respuesta puede ser válida para readLine() (cuando la línea está vacía) y readString() (cuando se suceden dos delimitadores).
-------	--

setEncoding	Establece la codificación de caracteres (UTF-8 predeterminada). Los nombres utilizados son los que se encuentran en el documento: http://www.iana.org/assignments/character-sets (columna Name). El conjunto de caracteres disponibles dependen de la implementación.
-------------	--

Servicio OutputStream

writeBytes	Escribe una secuencia completa en el flujo (llamada de bloqueo).
flush	Vacía la memoria intermedia (buffers) usada por el flujo
closeOutput	Se utiliza par indicar que se han escrito todos los datos.

Service TextOutputStream

Heredado de OutputStream.

writeString	Escribe una cadena en el flujo utilizando la codificación definida por setEncoding. Los saltos de línea y delimitadores necesarios se tienen que añadir manualmente a la cadena.
setEncoding	Establece la codificación de caracteres (UTF-8 predeterminada). Los nombres utilizados son los que se encuentran en el documento: http://www.iana.org/assignments/character-sets (col. Name). El conjunto de caracteres disponibles dependen de la implementación.

Ejemplo: Lectura de un archivo de texto

```
Dim oSFA As Object, oInText As Object
Dim URLArchivo As String, Linea As String

oSFA = createUNOService("com.sun.star.ubc.SimpleFileAccess")
URLArchivo = ConvertToURL("C:\ruta\MiArchivo.txt")
oInText = createUNOService("com.sun.star.io.TextInputStream")
oInText.setInputStream(oSFA.openFileRead(URLArchivo))
Linea = oInText.readLine()
oInText.closeInput()
```

Ejemplo: Escritura en un archivo de texto

```
Dim oSFA As Object, oOutText As Object
Dim URLArchivo As String

oSFA = createUNOService("com.sun.star.ubc.SimpleFileAccess")
URLArchivo = ConvertToURL("C:\ruta\MiArchivo.txt")
oOutText = createUNOService("com.sun.star.io.TextOutputStream")
oOutText.setOutputStream(oSFA.openFileWrite(URLArchivo))
'escritura (se debe especificar el delimitador de línea)
'[empleo de CRLF (Windows)]
oOutText.WriteString("Hola Mundo" & Chr(13) & Chr(10))
oOutText.WriteString("Línea 2" & Chr(13) & Chr(10))
'vaciado de los buffers y cierre
oOutText.flush
oOutText.closeOutput()
```

Creditos

Autor : Jean-François Nifenecker – jean-francois.nifenecker@laposte.net
Somos como enanos sentados sobre los hombros de gigantes. Si vemos más cosas y más lejanas que ellos, no es por la perspicacia de nuestra visión, ni por nuestra grandeza, sino porque son ellos los que nos elevan. (Bernard de Chartres [atribuido])

Historial

Versión	Fecha	Comentarios
1.01	01/10/2017	Primera versión.
1.14	14/12/2019	Correcciones menores.
	10/04/2021	Traducción y ajuste formato: B. Antonio Fernández

El documento original se puede obtener en la [Wiki francesa de publicaciones de L.O.](#)

Licencia

Esta guía de referencia está bajo licencia **Creative Commons BY-SA v3 (fr)**.
Información de la licencia : [en español](https://creativecommons.org/licenses/by-sa/3.0/fr/)
<https://creativecommons.org/licenses/by-sa/3.0/fr/>

