



LibreOffice RefCard

LibreOffice BASIC Files

v. 3.0 – 09/02/2025

Written using LibreOffice v.25.2 – Platform: All



File paths

LibreOffice is multi-platform, thus file names are often expressed in URL form:

file:///device/path/to/somefile.ext

From OS to URL `UrlFileName = ConvertToURL(OSFileName)`

From URL to OS `OSFileName = ConvertFromURL(URLFileName)`

BASIC Instructions

<code>Dir()</code>	Returns a file or dir name, or the names of all files and directories found on a drive or dir which correspond to the specified path.
<code>FileCopy()</code>	Copies a file.
<code>FileDateTime()</code>	Returns a date and time of a file creation or last mod, as a string.
<code>FileExists()</code>	Returns True if a file or directory exists.
<code>FileLen()</code>	Returns a file length (in bytes).
<code>GetAttr()</code>	Returns a file, volume or directory type.
<code>GetPathSeparator()</code>	Returns the path separation character for the operating system.
<code>Kill()</code>	Deletes a file from a drive.
<code>MkDir()</code>	Creates a new directory.
<code>Name()</code>	Renames an existing file or directory.
<code>RmDir()</code>	Deletes an existing directory.
<code>SetAttr()</code>	Defines a given file attributes.

Managing file contents using Instructions in BASIC

Process

1. **Get** an internal identifier on the file (FreeFile),
2. **Open** the file (Open),
3. **Write** to the file (Print, Put or Write) or **read** it (Get, Line Input# or Input#),
4. **Close** the file (Close)

Accessing a file contents using its handle (identifier).

<code>Close</code>	Closes a file that was opened using Open.
<code>Eof()</code>	Checks whether the file pointer is at the end.
<code>FileAttr()</code>	Returns the handle of a file that was opened using Open.
<code>FreeFile</code>	Returns an available handle number before opening a file.
<code>Get</code>	Reads a record from a file and inserts it into a variable.
<code>Input</code>	Sequentially reads an opened file data.
<code>Line Input</code>	Reads a line from a file.
<code>Loc()</code>	Returns the current position in an opened file.
<code>Lof()</code>	Returns an opened file size in bytes.
<code>Open</code>	Opens a data channel.
<code>Print</code>	Writes data into a sequential file (undelimited line).
<code>Put</code>	Writes a record into a file.
<code>Reset</code>	Close all open files and flushes the contents to the device of all file buff.
<code>Seek()</code>	Returns the next read/write pos in a file opened using Open For Random.
<code>Write</code>	Writes data into a sequential file (delimited line).

Print or Write?
Print saves the text as-is.
Write adds delimiters according to the information type (text: ", date or boolean: #).
These delimiters are ignored when read using Input.

Opening modes (Text files)

Sequential **write**: Open For Output Sequential **read**: Open For Input

Hint: For other access modes (direct or binary access), use the stream APIs.

Sequential read example (Text file)

Read a file known from its FileURL address.

```
Dim Handle As Integer, TheLine As String
Handle = FreeFile
Open FileURL For Input As #Handle
'line reads
Do While Not Eof(Handle)
    'we read the current line
    Line Input #Handle, TheLine
Loop
Close #Handle
```

Sequential write example (Text file)

Write to a file known from its FileURL address.

```
Dim Handle As Integer : Handle = FreeFile
Open FileURL For Output As #Handle
'line writes
Print #Handle, "Some text."
Print #Handle, "More..."
Print #Handle, "The end."
Close #Handle
```

Using a SimpleFileAccess object

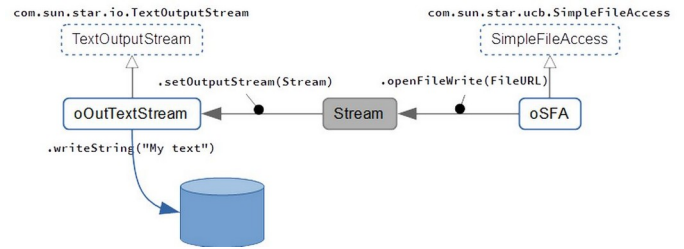
<code>oSFA = createUNOService("com.sun.star.ucb.SimpleFileAccess")</code>	
<code>copy</code>	Copies a file.
<code>createFolder</code>	Creates a new directory
<code>exists</code>	Checks whether a file or directory exists.
<code>getContentType</code>	Returns a file contents type.
<code>getDateTimeModified</code>	Returns a file last modification date.
<code>getFolderContents</code>	Returns a directory contents
<code>getSize</code>	Returns a file size.
<code>isFolder</code>	Checks whether an URL is a directory.
<code>isReadOnly</code>	Checks whether a file is set as read-only.
<code>kill</code>	Del a file or directory. A dir is always deleted empty or not.
<code>move</code>	Moves a file.
<code>setInteractionHandler</code>	Declares an interaction handler for ulterior operations.
<code>setReadOnly</code>	Sets a file read-only flag (rights necessary).

Managing file contents using the stream APIs

We call the LibreOffice API SimpleFileAccess and Stream services.

Overview

Example : writing to a text file (the actual code is shown in the example below).



Process

1. **Create the object** for accessing files,
`oSFA = createUNOService ("com.sun.star.ucb.SimpleFileAccess")`
2. **Connect** the stream that corresponds to the process (access type),
3. **Write** to the file or **read** it (according to the file type),
4. In write mode, **flush** the stream (.flush),
5. **Close** the file (.closeXxx).

Accessing file contents

SimpleFileAccess service (SFA)

<code>openFileRead</code>	Opens a file in read mode.
<code>openFileWrite</code>	Opens a file in write mode.
<code>openFileReadWrite</code>	Opens a file in write and read mode.

Stream services (InputStream, OutputStream and Stream)

These are the “active” parts.

Correspondance between the SFA service and streams:

SFA methods	Stream services
<code>openFileRead</code>	<code>com.sun.star.io.InputStream</code> <code>com.sun.star.io.TextInputStream</code>
<code>openFileWrite</code>	<code>com.sun.star.io.OutputStream</code> <code>com.sun.star.io.TextOutputStream</code>
<code>openFileReadWrite</code>	<code>com.sun.star.io.Stream</code>

Stream service

<code>getInputStream</code>	Returns the InputStream part of a read/write stream.
<code>getOutputStream</code>	Returns the OutputStream part of a read/write stream.

Closing these streams closes the OutputStream or InputStream respectively.

InputStream service

<code>readBytes</code>	Reads the specified number of bytes.
<code>readSomeBytes</code>	Reads the available number of bytes, with the maximum specified.
<code>skipBytes</code>	Skips the specified number of bytes (positive value).
<code>available</code>	Returns the number of bytes that can safely be read or skipped.
<code>closeInput</code>	Closes the stream.

TextInputStream service

Inherits from InputStream.

<code>readLine</code>	Reads the text until a line break (CR, LF, or CRLF) or EOF is met. Returns the string read (without CR or LF). The chars read are converted according to setEncoding. If EOF was reached before running the method, a 0-length string is returned.
<code>readString</code>	Reads the text until a delimiter or EOF is met. Returns the string read. CRLF is not the default delimiter! This means that, when no delimiter is specified or found, the stream is read to EOF. The characters read are converted according to the encoding set using setEncoding. If EOF was reached before running the method, a zero-length string is returned.
<code>isEOF</code>	Returns the EOF status. This status cannot be detected by a read attempt for an empty string because such a response may be valid when using readLine() (the line is empty) or readString() (two delimiters inline).

setEncoding Sets the char encoding (defaults to UTF-8). The available names are listed here: <http://www.iana.org/assignments/character-sets> (Name column). The current supported char sets depend upon the implementation.

OutputStream service

<code>writeBytes</code>	Writes a sequence of bytes in the stream (blocking call).
<code>flush</code>	Flushes the stream buffers.
<code>closeOutput</code>	Used to signal that all data have been written.

TextOutputStream service

Inherits from OutputStream.

<code>writeString</code>	Writes a string to the stream (encoding specified using setEncoding). Line breaks and delimiters that might be necessary, have to be manually added to the string.
--------------------------	---

setEncoding See above TextInputStream.setEncoding.

Example : Reading from a text file

```
Dim oSFA As Object, oInText As Object, FileURL As String, TheLine As String
oSFA = createUNOService("com.sun.star.ucb.SimpleFileAccess")
FileURL = ConvertToURL("C:\path\to\MyFile.txt")
oInText = createUNOService("com.sun.star.io.TextInputStream")
oInText.setInputStream(oSFA.openFileRead(FileURL))
TheLine = oInText.readLine()
oInText.closeInput()
```

Example : Writing to a text file

```
Dim oSFA As Object, oOutText As Object, FileURL As String
oSFA = createUNOService("com.sun.star.ucb.SimpleFileAccess")
FileURL = ConvertToURL("C:\path\to\MyFile.txt")
oOutText = createUNOService("com.sun.star.io.TextOutputStream")
oOutText.setOutputStream(oSFA.openFileWrite(FileURL))
'write (line delimiters must be specified)
oOutText.WriteString("Hello World" & Chr(13) & Chr(10)) 'CRLF (Windows)
oOutText.WriteString("Line #2" & Chr(13) & Chr(10))
'flush buffers and close
oOutText.flush
oOutText.closeOutput()
```

