

Installing and Loading Tesseract OCR with MBS Plugin

Everything you need to find Tesseract libraries for macOS, Windows and Linux and instructions to load and initialize them with the MBS plugin ([OCR.Load](#), [OCR.IsLoaded](#), [OCR.Initialize](#)).

Quick summary (tl;dr)

- **macOS:** Install via Homebrew (``brew install tesseract` + `brew install tesseract-lang`` for extra languages) or download prebuilt libs (MBS offers lib bundles). Then call `MBS("OCR.Load"; "/opt/homebrew/lib/libtesseract.4.dylib")` (or path where Homebrew installed libs).
- **Linux (Debian/Ubuntu):** Install libraries and language files with `apt: sudo apt-get install libtesseract5 tesseract-ocr-lang`. Then call `MBS("OCR.Load"; "libtesseract.so.5")` (and load leptonica first if required).
- **Windows:** Use a community build/installer (UB-Mannheim and similar). Place DLLs in program folder and call `MBS("Process.SetCurrentDirectory"; "C:\Program Files\Tesseract-OCR") & MBS("OCR.Load"; "libtesseract-4.dll")`.
- **Important:** Tesseract library version must match tessdata files (wrong combinations can crash). When using Tesseract 4.x+ with MBS, prefer `OCR.Load` to explicitly load newer runtimes.

Why explicit loading matters (MBS plugin)

The MBS FileMaker/Xojo plugin ships with its own built-in engine historically (older Tesseract 3.x). To use modern Tesseract 4.x or later you must explicitly load the newer native libraries using `MBS("OCR.Load"; Path)`. Use `MBS("OCR.IsLoaded")` to check whether a newer engine is active. This avoids accidentally using mismatched data files and improves accuracy and speed on modern versions.

Warning: Using tessdata files for a different major Tesseract version than the library can crash the process. Always match the library version and tessdata set.

Where to get the libraries (official & common sources)

macOS

Best route for most developers: **Homebrew**. Install the engine and optional language packages with:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
brew install tesseract  
brew install tesseract-lang # optional: many extra languages
```

Homebrew places libraries in the Cellar (and symlinks under `/opt/homebrew/lib` on Apple Silicon or `/usr/local/lib` on Intel macs). You can point `OCR.Load` to those dylib files.

Linux (Debian / Ubuntu style)

Use your distribution packages for the simplest setup. Example (Debian/Ubuntu):

```
sudo apt-get update  
sudo apt-get install libtesseract5 tesseract-ocr tesseract-ocr-
```

Package names differ by distro and Tesseract version — the important result is the shared libraries such as `libtesseract.so.5` and the leptonica library (e.g. `liblept.so.5` or similar).

Windows

There is no single official Windows installer maintained by upstream; common community builds are provided (UB-Mannheim is widely used). Download the installer or zip from community builds, install to `C:\Program Files\Tesseract-OCR` (or a folder you choose), and the folder will contain DLLs and a ``tessdata`` folder.

After installation, call `Process.SetCurrentDirectory` (or ensure your process can find the DLLs) and load the DLLs with `OCR.Load`.

Prebuilt bundles / MBS-provided libs

Monkeybread Software publishes a ready-to-download library bundle that includes leptonica and tesseract binaries prepared for use with the MBS plugin. These can be handy if you want a known-good pair of libraries for FileMaker/Xojo projects.

Loading Tesseract into the MBS plugin — practical examples

The MBS plugin provides three functions you'll use:

- `MBS("OCR.IsLoaded")` — returns 1 if a newer Tesseract (4.x+) is loaded.
- `MBS("OCR.Load"; Path)` — load native library by full native path or filename (depending on OS and working dir).
- `MBS("OCR.Initialize"; PathToTessdata; "eng")` — point tesseract to the `tessdata` folder and select language(s).

Linux example

```
# (install libs once)
sudo apt-get install libtesseract5 tesseract-ocr-eng

# In FileMaker script (pseudo)
# load leptonica first if not part of the same library file:
MBS("OCR.Load"; "liblept.so.5") &
MBS("OCR.Load"; "libtesseract.so.5")

# then initialize (point to directory which contains
tessdata):
MBS("OCR.Initialize"; "/usr/share/tesseract-ocr/4.00/
tessdata"; "eng")
```

macOS example (Homebrew)

```
# install via Homebrew (once)
brew install tesseract
brew install tesseract-lang # optional

# In FileMaker script:
MBS("OCR.Load"; "/opt/homebrew/lib/liblept.5.dylib") &
MBS("OCR.Load"; "/opt/homebrew/lib/libtesseract.4.dylib")

# initialize (point to tessdata root - common homebrew path
is /opt/homebrew/Cellar/tesseract//share/tessdata)
MBS("OCR.Initialize"; "/opt/homebrew/share/tessdata"; "eng")
```

Windows example (UB-Mannheim installer)

```
# After installing to C:\Program Files\Tesseract-OCR
MBS("Process.SetCurrentDirectory"; "C:\\Program Files\
\Tesseract-OCR") &
MBS("OCR.Load"; "libleptonica-6.dll") &
MBS("OCR.Load"; "libtesseract-5.dll")
```

```
# Initialize: pass native path to parent of tessdata (must
end with tessdata folder name)
```

```
MBS("OCR.Initialize"; "C:\\Program Files\\Tesseract-OCR\
\tessdata"; "eng+deu")
```

Notes:

- If the leptonica code is included in the same dynamic library file you may only need to load one file. If not, load leptonica first then tesseract (order matters).
- On macOS the full path to the `.dylib` is usually required; on Linux a soname like libtesseract.so.5 can work if the loader searches the standard system library paths.`
- On Windows you may need to set the current directory or ensure the DLLs are discoverable by the process loader (PATH or same folder).

Initializing OCR and language data (tessdata)

After the native library is loaded, call `MBS("OCR.Initialize"; PathToTessdata; Lang)`. The **PathToTessdata** parameter must be a native path that ends with the folder name `tessdata`. The language string is usually an ISO 639-3 code (for English use `"eng"`). For multiple languages use plus sign: `"eng+deu"`.

Important: data files are version-specific. Tesseract 4.x and 5.x use different tessdata models (`tessdata_fast` / `tessdata_best`); feeding 3.x data to a 4.x+ library or vice versa may crash the process.

MBS also provides a download for language files and an archive of ready-to-use tessdata that matches the plugin's expectations — this can simplify deployment, especially on FileMaker Server where permissions and file access can be restricted.

Common troubleshooting tips

- 1. Check OCR.IsLoaded** before relying on default bindings: `If [MBS("OCR.IsLoaded") = 0]`.
- 2. Library load errors:** If `MBS("OCR.Load")` reports errors, confirm the path, file name and loader search paths. On macOS Homebrew paths differ by CPU architecture (`/opt/homebrew` vs `/usr/local`).
- 3. Missing tessdata files:** If initialization fails or Tesseract aborts, check that tessdata contains the language model and any required auxiliary files. Some languages require additional traineddata or config files.
- 4. Version mismatch:** If OCR crashes unexpectedly, confirm that the tessdata files were built for the same major engine version you loaded.
- 5. Server environments:** On FileMaker Server, each server thread may need per-thread initialization: check `MBS("OCR.IsInitialized")` and initialize as needed for each execution thread.

Deployment checklist

- Ensure the correct native libraries (leptonica + libtesseract) are available on the target system.
- Ship or point to the matching tessdata folder; verify language files exist.
- Use `MBS("OCR.Load")` to load the native libraries when you require a newer engine (4.x or later).
- Initialize per-thread on server installations using `MBS("OCR.Initialize")`.
- Test OCR on representative documents for each language to surface missing model files early.
- If Windows can't find a module, you may use [Process.SetDllDirectory](#) function to tell Windows where to find the DLL files.