

# How to script the Esko PDF Export plug-in

---

## 1. Initialize the scripting in the PDF Export plug-in

Insert the paths where to look for the Esko PDF Export plug-in into the ExternalObject.searchFolders variable and then create an ExternalObject instance.

If there's more than one search folder, use ";" as the separator.e.g:

```
ExternalObject.searchFolders="..\..\..\Plug-ins;..\..\..\Plug-ins\myplugins;" + ExternalObject  
.searchFolder
```

Example:

```
ExternalObject.searchFolders="C:\Program Files\Adobe\Adobe Illustrator  
2020\Plug-ins\Esko\Data Exchange\PDF Export;" + ExternalObject.searchFolder  
var dw = new ExternalObject("lib:PDFExport_WAI24r.aip");
```

## 2. Open a document to work with:

Example:

```
fileName = "C:\sample.ai"  
var f = new File(fileName);  
doc = app.open(f);
```

## 3. Create an instance of the EskoPDFExport class

Example:

```
var scripter = new EskoPDFExport();
```

## 4. Specify the output file name

Example:

```
scripter.outputPath = "C:\output.pdf";
```

## 5. Set other attributes of the EskoPDFExport object

All attributes except "outputPath" are optional and have a default value.

The output file format (Normalized PDF or PDF+) is automatically determined and it depends on the document mode specified in Ink Manager.

*Attributes common to Normalized PDF and PDF+ files:*

Name	Type	Default	Description
borderMode	int	3	Trim Box fitting, one of the following numbers: 1 - Artwork bounding box 2 - Current artboard 3 - Trim Box from the document
fitMediaBoxToArtwork	bool	false	True to fit Media Box to artwork.
outlineText	bool	false	True to outline all text.
includeHiddenObjectsAndLayers	bool	false	True to include hidden objects and layers.
includeNotes	bool	true	True to include notes.

*Attributes specific to Normalized PDF files:*

Name	Type	Default	Description
embedImages	bool	false	True to use embed images, false to save links.
addPreview	bool	false	Only valid when embedImage is false.

copyImages	bool	false	True to add preview. Only valid when embedImage is false. True to copy linked files next to the output file, False to use links and apply server mapping.
copyImagesNotOnServers	bool	false	Only valid when copyImages is false. True to copy links if not on file servers.
expandPatterns	bool	false	True to expand patterns.
contourizeBitmaps	bool	false	True to contourize bitmaps.
blendResolution	int	600	Any integer value, 0 for off.

*Attributes specific to PDF+ files:*

Name	Type	Default	Description
copyStructuralDesignFilesNotOnServer	bool	false	True to copy CAD files next to the output when exporting to Automation Engine.

Example:

```

scripter.borderMode = 3;
scripter.fitMediaBoxToArtwork = false;
scripter.outlineText = false;
scripter.includeHiddenObjectsAndLayers = false;
scripter.includeNotes = true;
scripter.embedImages = false;
scripter.addPreview = false;
scripter.copyImages = false;
scripter.copyImagesNotOnServers = false;
scripter.expandPatterns = false;
scripter.contourizeBitmaps = false;
scripter.blendResolution = 600;
scripter.copyStructuralDesignFilesNotOnServer = false;

```

## 6. Call the exportPDF function to export the pdf

Example:

```
scripter.exportPDF();
```

Note: You can also specify the output file as a parameter, this value would overwrite the outputPath attribute.

Example:

```
scripter.exportPDF("C:\\output.pdf");
```

## 7. Get messages from the EskoPDFExport class

Example:

```

var ErrorCode    = scripter.errorCode;
var ErrorMessage = scripter.errorMessage;
var AllMessages  = scripter.formattedExportMessage;

if (ErrorCode == 0)
    // Get warning messages when file has been exported successfully
    $.writeln("Export Success.\n", "Messages: \n", AllMessages);
else
    // Get the error code and error message in case output fails
    $.writeln("Export Failed.\n", "Error Code: ", ErrorCode, " Error Message:",
        ErrorMessage);

```

## 8. Close the document when the export is done

Example:

```
doc.close();
```

**A very simple JSX can look like this:**

-----SAMPLE-----

```
ExternalObject.searchFolders="C:\\Program Files\\Adobe\\Adobe Illustrator
                                2020\\Plug-ins\\Data
                                Exchange\\Esko;" + ExternalObject.searchF
                                older

var dw = new ExternalObject("lib:PDFExport_WAI24r.aip");
fileName = "C:\\sample.ai"
var f = new File(fileName);
doc = app.open(f);
var scripter = new EskoPDFExport();
scripter.outputPath = "C:\\output.pdf";
scripter.borderMode = 3;
scripter.fitMediaBoxToArtwork = false;
scripter.outlineText = false;
scripter.includeHiddenObjectsAndLayers = false;
scripter.includeNotes = true;
scripter.embedImages = false;
scripter.addPreview = false;
scripter.copyImages = false;
scripter.copyImagesNotOnServers = false;
scripter.expandPatterns = false;
scripter.contourizeBitmaps = false;
scripter.blendResolution = 600;
scripter.copyStructuralDesignFilesNotOnServer = false;
scripter.exportPDF();

var ErrorCode = scripter.errorCode;
var ErrorMessage = scripter.errorMessage;
if (ErrorCode != 0)
    $.writeln("Error Code: ", ErrorCode, "    ErrorMessage:", ErrorMessage);
doc.close();
```

-----END-----

### How to run the sample script

1. Copy the script files into Adobe Illustrator/scripting folder
2. Open ExportPDFsampleScript.jsx and verify that the search folder is correctly set to the PDF export plugin folder.

3. On Windows: open ExportEskoPDF.vbs

On the Mac: open ExportEskoPDF.applescript

Specify the inputFiles and outputFolder in the last part of the code.

The default input is "Sample.ai" in the current folder.

The default output folder is the folder of the script file.

4. On Windows: double click the vbs to run the script.

On the Mac: click Run button from the AppleScript Editor to run the script.