

FinalWriter

Simon Ihmig

COLLABORATORS

	<i>TITLE :</i> FinalWriter		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Simon Ihmig	August 9, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	FinalWriter	1
1.1	MathScript & FinalWriter	1
1.2	einleitung	1
1.3	Installation	1
1.4	Macken von FinalWriter	2
1.5	Verwendung der ARexx-Skripte	2
1.6	Frequently Asked Questions	2

Chapter 1

FinalWriter

1.1 MathScript & FinalWriter

Benutzungshinweise und Tips zur Verwendung von MathScript mit FinalWriter.

[Einleitung](#)

[Installation](#)

[Macken von FinalWriter](#)

[Verwendung der ARexx-Skripte](#)

[FAQ](#)

FinalWriter ©Copyright SoftWood Inc.

1.2 einleitung

Um EPS-Dateien von MathScript in Textverarbeitungs- und DTP-Programmen zu nutzen, müssen diese in der Lage sein, solche EPS-Dateien einzuladen. Die meisten sollten das können. Um den PostScript Code jedoch auch anzeigen bzw ausdrucken zu können, muß dieser interpretiert werden, was normalerweise in einem PostScript-fähigen Drucker geschieht. Wenn man jedoch ein Vorschau-Abbild in die Datei integriert, kann sie auch angezeigt bzw auf Druckern, die nicht PostScript-fähig sind, ausgegeben werden. MathScript ist ab der Version 2.0 dazu in der Lage (Binary EPS). Wenn man beim Ausdruck jedoch hochfertige Ergebnisse erhalten möchte, muß man sehr hochauflösende Vorschau-Abbilder benutzen, was entsprechend viel Speicher verbraucht. Da FinalWriter jedoch auch in der Lage ist, PostScript selbst zu interpretieren, können von MathScript erstellt Binary EPS Dateien schnell in geringer Auflösung auf dem Monitor angezeigt werden und in bester Qualität durch Verwendung des PostScript Codes ausgedruckt werden.

Wenn Sie einen PostScript-Drucker besitzen (und auch als PostScript drucken), verändern Sie die Einstellungen von MathScript so, daß der mathematische Zeichensatz in die PostScript-Beschreibung integriert wird, da der PostScript-Interpreter im Drucker den Font nicht von der Festplatte nachladen kann. Allerdings entsteht damit ein kleiner Fehler (siehe [Macken von FinalWriter](#)).

Wenn Sie keinen solchen Drucker besitzen, gehen Sie sicher, daß der Font nicht eingebettet wird, damit dieser Fehler umgangen wird und die Dateien nicht zu groß werden.

1.3 Installation

Um das Zusammenspiel von MathScript und FinalWriter zu erleichtern, sollten sie folgendes tun:

- Definieren Sie in FinalWriter zwei neue Schalter in der Befehlsleiste, die beide ARexx-Skripte aufrufen. Als ARexx-Skript des einen Schalters wählen Sie MathScript:Rexx/OpenMathScript.ms, für den anderen nehmen Sie MathScript:Rexx/CloseMathScript.ms.

- Legen Sie das Makro InsertFormula.ms auf eine beliebige Funktionstaste in MathScript.
- Editieren Sie die Datei ../FinalWriter/FWFiles/init.ps . Dort müßte sich folgender Abschnitt befinden:

```
systemdict /findfont { % Comment out the next 3 lines if you don't have a PSFonts: directory % If you have a directory containing
Type-1 fonts to use for EPS clip % art, substitute the path name of the directory for "PSFonts:" below. -->% dup FontDirectory
exch known not % Not in FontDirectory? -->% { dup (PSFonts:) () loadfont % Try to load from PSFonts: -->% } if
```

Tun Sie, was im Kommentar steht, nämlich entfernen Sie die Prozentzeichen am Anfang der letzten drei Zeilen.

Das wär's auch schon:)

1.4 Macken von FinalWriter

FinalWriter scheint einen kleinen Bug im Umgang mit EPS-Dateien zu haben. Wenn man eine EPS-Datei, die einen Font integriert hat, mit dem Dokument abspeichert, ist diese beim erneuten Laden des Dokuments nicht mehr sichtbar (bei "Vollabbild")

Da dies nur in Verbindung mit einem PostScript-Drucker vonnöten ist, ist dies weiter nicht tragisch, da der Ausdruck als PostScript trotzdem korrekt ist, und die Formel durch die Vorschau (Binary EPS) auf dem Monitor angezeigt werden kann

Ohne PostScript-Drucker braucht man den Font eh nicht in die EPS-Datei integrieren, so daß dieser Fehler nicht auftaucht.

1.5 Verwendung der ARexx-Skripte

Um die Benutzung von MathScript zusammen mit FinalWriter zu erleichtern, werden drei Skripte benutzt: OpenMathScript.ms CloseMathScript.ms InsertFormula.ms Das erste wird von FinalWriter aus aufgerufen (aus dem Menu oder durch Benutzerdefinierte Schalter, siehe [Installation](#)). Die letzten beiden werden durch MathScript gestartet.

Die Funktionen im einzelnen:

OpenMathScript.ms: Dieses Skript holt den Screen von MathScript in den Vordergrund bzw startet das Programm, wenn es noch nicht läuft. Falls beim Start eine Formelgrafik in FW selektiert war, wird die zugehörige Formelbeschreibung im MathScript eingeladen, wodurch diese Formel nachträglich verändert werden kann.

InsertFormula.ms: Dieses Skript speichert die aktuelle Formel als Binary EPS Datei und lädt sie in FinalWriter. Falls eine Formelgrafik in FW selektiert ist, wird diese durch die neue ersetzt, andernfalls wird sie auf der aktuellen Seite eingefügt.

CloseMathScript: Dieses Makro darf erst aufgerufen werden, wenn das Dokument in FinalWriter abgespeichert und geschlossen wurde. Es löscht alle temporären Dateien, die durch InsertFormula.ms erstellt wurden.

Um nun ein Dokument mit mathematischen Formeln durch FinalWriter zu erstellen, tun Sie folgendes:

1. Starten Sie FinalWriter
2. Sobald Sie eine Formel benötigen bzw eine schon vorhandene editieren wollen, starten Sie OpenMathScript.ms
3. Geben Sie Ihre Formel ein.
4. Starten Sie von MathScript aus das Skript InsertFormula.ms.

Wiederholen Sie die Schritte 2-4, bis Sie alle Formeln haben.

6. Nachdem Sie ihr Dokument abgespeichert und geschlossen haben, starten Sie CloseMathScript.ms.

1.6 Frequently Asked Questions

Bei Benutzung der Option "Vollabbild" in FW oder beim Ausdruck ist keine Formel zu sehen. Überprüfen Sie folgende Punkte: In der Datei init.ps (im Verzeichnis FWFiles von FW) muß folgender Abschnitt enthalten sein (siehe auch [Installation](#)):

```
systemdict /findfont { % Comment out the next 3 lines if you don't have a PSFonts: directory % If you have a directory containing
Type-1 fonts to use for EPS clip % art, substitute the path name of the directory for "PSFonts:" below. dup FontDirectory
exch known not % Not in FontDirectory? { dup (PSFonts:) () loadfont % Try to load from PSFonts: } if
```

[...]

Bei Eingabe von "assign PSFonts: exists" muß u.a. der Pfad von MathScript ausgegeben werden. Dieser Assign sollte vom Install-Script erstellt worden sein.

Im Verzeichnis von MathScript muß die Datei PSFont enthalten sein.

Bei der Benutzung des Makros InsertFormula.ms wird der Fehler angezeigt, daß der Datenträger Ram Disk voll ist. Tun Sie folgendes: Editieren Sie die Dateien InsertFormula.ms und CloseMathScript.ms. Ändern Sie alle Stellen, an denen "T:" steht, in einen beliebigen Pfad um. Damit werden die Formeln anstatt ins RAM dorthin gespeichert (und danach selbstverständlich wieder gelöscht).