



Troi **Graphic** Plug-in

# **TROI GRAPHIC PLUG-IN™ 1.0 USER GUIDE**

**October 1998**

**Troi Automatisering**  
Vuurlaan 18  
2408 NB Alphen a/d Rijn  
The Netherlands  
Fax: +31-172-470539

You can also visit the Troi web site at: <<http://www.troi.com/>> for additional information.

Troi Graphic Plug-in is copyright 1998 of Troi Automatisering. All rights reserved (28-October-98).

# Table of Contents

Installing plug-ins..... 1

Summary of functions..... 2

Using external functions..... 2

Trgr-Version.....2

Trgr-RGBToClip.....3

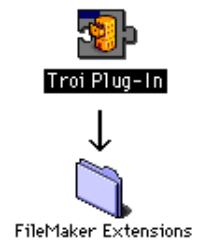
Trgr-ScreenToClip.....4

Creating Thumbnail Images ..... 5

## Installing plug-ins

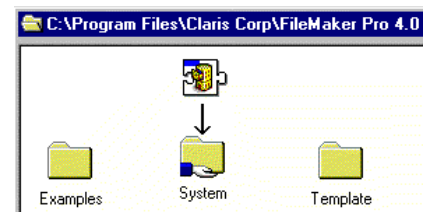
### For Macintosh:

- Quit FileMaker Pro.
- Put the file "Troi Graphic Plug-in" from the folder "Macintosh Plug-in" into the "FileMaker Extensions" folder in the FileMaker Pro 4.0 folder.
- If you have installed previous versions of this plug-in, you are asked: "An older item named "Troi Graphic Plug-In" already exists in this location. Do you want to replace it with the one you're moving?". Press the OK button.
- Start FileMaker Pro. The first time the Troi Graphic Plug-in is used it will display a dialog box, indicating that it is loading and showing the registration status.



### For Windows:

- Quit FileMaker Pro.
- Put the file "trgraph.fmx" from the directory "Windows Plug-in" into the "SYSTEM" subdirectory in the FileMaker Pro 4.0 directory.
- If you have installed previous versions of this plug-in, you are asked: "This folder already contains a file called 'trgraph.fmx'. Would you like to replace the existing file with this one?". Press the Yes button.
- Start FileMaker Pro. The Troi Graphic Plug-in will display a dialog box, indicating that it is loading and showing the registration status.



**TIP** You can check which plug-ins you have loaded by going to the plug-in preferences: Choose **Preferences** from the **Edit** menu, and then choose **Plug-ins**.

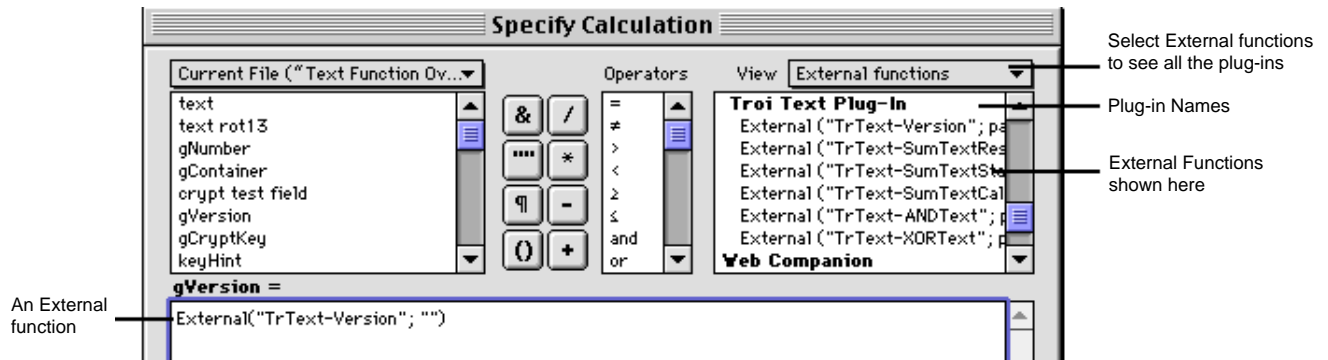
You can now open the file "Graphic Examples.fp3" to see how to use the plug-in's functions. There is also a Function overview in this file.

**IMPORTANT** There is a problem in FileMaker Pro 4.0v1. Please make sure that all plug-ins that are in the folder "FileMaker Extensions" are enabled in the preferences. (Under Edit/ Preferences/ Application/ Plug-ins). Make sure all plug-ins have a cross before their name. Remove plug-ins you don't use from the "FileMaker Extensions" folder.

NB: This bug is fixed in version 4.1 and 4.0v2.

## Summary of functions

Plug-ins add new functions to the standard functions that are available in FileMaker Pro. You can see those extra functions for all plug-ins at the top right of the Specify Calculation Box:



**IMPORTANT** In the United States, commas act as list separators in functions. In other countries semicolons might be used as list separators. The separator being used depends on the operating system your computer uses, as well as the one used when the file was created. All examples show the functions with semicolons.

The Troi Graphic Plug-in adds the following functions:

<u>function name</u>	<u>short description</u>
Trgr-Version	check for correct version of the plug-in
Trgr-RGBToClip	puts a RGB colour in the clipboard buffer
Trgr-ScreenToClip	puts (a part of) the computer screen in the clipboard buffer

## Using external functions

External functions for this plug-in can be used in a script step using a calculation. The functions Trgr-RGBToClip and Trgr-ScreenToClip should not be used in a define field calculation.

**IMPORTANT** The Balance functions have to be used in a specific way, to create the desired effect. See the section on Balance functions for the specifics on this.

## Trgr-Version

Example usage: External(Trgr-Version; "") will return "Troi Graphic Plug-in 1.0".

**IMPORTANT** always use this function to determine if the plug-in is loaded. If the plug-in is not loaded use of external functions may result in data loss, as FileMaker will return an empty field to any external function that is not loaded.

## Trgr-RGBToClip

**Syntax**    `External("Trgr-RGBToClip" , "switches/ red / green / blue")`

Puts a small coloured square (8x8) on the clipboard. The colour of the square is determined by the RGB (Red, Green, Blue) parameters.

### Parameters

*switches*: not used, reserved for future use. Leave blank or put "-unused"

*red*: the value of the red component of the colour, use a number between 0 and 255

*green*: the value of the green component of the colour, use a number between 0 and 255

*blue*: the value of the blue component of the colour, use a number between 0 and 255

255 means full intensity of that colour and 0 means least intensity.

Returned result: an errorcode:

0	no error
\$\$-108	memFullErr    Ran out of memory

Other errors might be returned.

**NOTE** The original contents of the clipboard is lost.

Example usage:

```
External("Trgr-RGBToClip", "-unused" & gRed & "|" & gGreen & "|" & gBlue)
```

### Example

Define the following fields:

container	Container	
gErrorCode	Global	Number
gRed	Global	Number
gGreen	Global	Number
gBlue	Global	Number

In ScriptMaker define a script "Colour to Container" as follows:

```
Set Field [gErrorCode,  
    External("Trgr-RGBToClip", "-unused" & gRed & "|" & gGreen & "|" & gBlue)]  
Paste [Select, container]
```

This script will put a square on the clipboard with the colours specified by the global fields, and then paste them into the container field.

## Trgr-ScreenToClip

**Syntax** External("Trgr-ScreenToClip" , "*switches/ left / top/ right /bottom*")

Captures (a part of) the computer screen and puts it on the clipboard. The part that is captured is determined by the parameters.

### Parameters

*switches*: not used, reserved for future use. Leave blank or put "-unused"

*left*: the left coordinate (in pixels) of the area to capture

*top*: the top coordinate of the area to capture

*right*: the right coordinate of the area to capture

*bottom*: the bottom coordinate of the area to capture

Returned result: an errorcode:

0	no error
\$\$-108	memFullErr     Ran out of memory

**NOTE** The original contents of the clipboard is lost.

Example usage:

```
Set Field[gErrorCode, External("Trgr-ScreenToClip", "-unused|" &"0|0|20|20")]
```

will return a the a part of the screen on the clipboard.

### Example

Define the following fields:

screenImage	Container	
gErrorCode	Global	Number
gLeft	Global	Number
gTop	Global	Number
gRight	Global	Number
gBottom	Global	Number

In ScriptMaker define a script "Screen to Container" as follows:

```
Set Field [gErrorCode, External("Trgr-ScreenToClip", "-unused "&
                                "|0|0|" & Status(CurrentScreenWidth) & "|" & Status(CurrentScreenHeigth)]
Paste [Select, screenImage]
```

This script will capture the entire screen into the screenImage container. This part in the calculation specifies the rectangle to capture:

```
"|0|0|" & Status(CurrentScreenWidth) & "|" & Status(CurrentScreenHeigth)
```

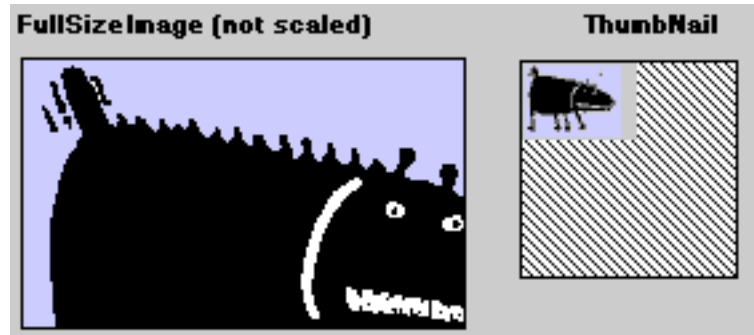
If this script is run on a 640x480 screen the rectangle to capture would be equivalent to:

```
"|0|0|640|480"
```

## Creating Thumbnail Images

One of the great possibilities of the ScreenToClip function is to make thumbnail images of large images in the database. If you have large images already in a FileMaker Pro database there was no way to easily make small preview images. But with the Graphic plug-in you can create small thumbnails, for example if you want to put your database on the web:

### Example



### Steps for creating thumbnails

These are the main steps to create them:

- 1 - create a thumbnail container field and some assisting global fields
- 2 - create a new layout with the thumbnail field and the original image on it.
- 3 - create a script to make a thumbnail of the image.
- 4 - if wanted you can do this in a loop for all your records.

### 1- Define Fields

Define the following fields:

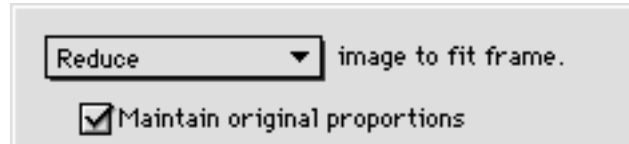
thumbnail	Container	
gErrorCode	Global	Number
gGlobalFieldBounds	Global	Text
gLocalFieldBounds	Global	Text
gLeft	Global	Number
gTop	Global	Number
gRight	Global	Number
gBottom	Global	Number
gLeftOffset Mac	Global	Number
gTopOffset Mac	Global	Number
gLeftOffset Win	Global	Number
gTopOffset Win	Global	Number

We assume that there is already this field:

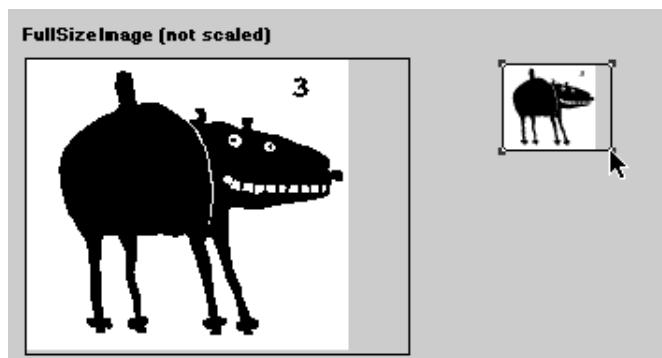
imageField Container

## 2- Create a thumbnailer Layout

Create a layout or modify an existing layout. Put both the **thumbnail** field and the original image field **imageField** on it. Set the Graphic formatting of **imageField** to Reduce image to fit frame: In layout Mode , select **imageField**, and then choose **Graphic** from the **Format** menu, and then choose **Reduce**. Also check **Maintain original proportions**.



Now change the dimensions of the frame of **imageField** to the size you want the thumbnail to be. In Browse mode you can see the image scaled down. This scaled image is going to be captured.



## 3- Create thumbnailer Scripts

In ScriptMaker define a script "Prepare for thumbnailing". This will get the global screen coordinates of the scaled image field. This script is not needed if you fill this **gGlobalFieldBounds** field manually.

Define "Prepare for thumbnailing" as follows:

```
Comment [Get the local coordinates of the field (relative to the window)]
Set Field [gLocalFieldBounds,
    FieldBounds(Status(CurrentFileName), Status(CurrentLayoutName), "ImageField") ]
Comment [Get rid of last number (which indicates rotation)]
Set Field [gLocalFieldBounds,
    Left(gLocalFieldBounds , Left(gLocalFieldBounds, Position(gLocalFieldBounds, " ",
        64000 , -1) -1) ) ]
Comment [Convert to global (screen) coordinates]
Comment [split the coordinates]
Set Field [gLeft, MiddleWords(gLocalFieldBounds, 1 , 1)]
Set Field [gTop, MiddleWords(gLocalFieldBounds, 2 , 1)]
Set Field [gRight, MiddleWords(gLocalFieldBounds, 3 , 1)]
Set Field [gBottom, MiddleWords(gLocalFieldBounds, 4 , 1)]
...
```



```

...
Comment [create the coordinate with offsets]
If [Status( CurrentPlatform)= 1]
    Set Field [gGlobalFieldBounds, NumToText(gLeft + gLeftOffset Mac)  &"|" &
        NumToText(gTop + gTopOffset Mac)  &"|" &
        NumToText(gRight + gLeftOffset Mac)  &"|" &
        NumToText(gBottom + gTopOffset Mac)  ]
Else
    Set Field [gGlobalFieldBounds, NumToText(gLeft + gLeftOffset Win)  &"|" &
        NumToText(gTop + gTopOffset Win)  &"|" &
        NumToText(gRight + gLeftOffset Win)  &"|" &
        NumToText(gBottom + gTopOffset Win)  ]
End If
Toggle Window [Maximize]
Toggle Status Area [Show]
Set Zoom Level [100%]

```

Now we have the coordinates captured, define a script "One record: Fill Thumbnail Field" as follows:

```

Comment [Get the screenshot to the clipboard]
Set Field [gErrorCode, External("Trgr-ScreenToClip", "-unused|" & gGlobalFieldBounds)]
If [gErrorCode = 0]
    Comment [Now paste it into the Thumbnail field]
    Paste [Select, ThumbNail]
Else
    Halt Script
End If

```

This script will capture the part of the screen as specified by **gGlobalFieldBounds** into the thumbnail field

#### 4- Create Thumbnails in a loop

Define a script "Loop: Make Thumbnails" as follows:

```

Perform Script [Sub-scripts, "Check Graph plug-in"]
Perform Script [Sub-scripts, "Prepare for thumbnailing"]
Go to Record/Request/Page [First]
Loop
    Perform Script [Sub-scripts, "One record: Fill Thumbnail Field"]
    Go to Record/Request/Page [Exit after last, Next]
End Loop
Exit Record/Request
Beep
Show Message ["Ready!"]

```

This will create the thumbnails for the current found set.