



# TROI GRAPHIC PLUG-IN 1.1 USER GUIDE

September 2001



**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 1998-2001 of Troi Automatisering. All rights reserved (25 September 2001).

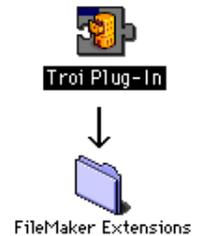
# Table of Contents

<b>Installing plug-ins.....</b>	<b>1</b>
<b>If You Have Problems.....</b>	<b>1</b>
<b>What can this plug-in do?.....</b>	<b>2</b>
<b>Getting started</b>	<b>2</b>
<b>Using external functions.....</b>	<b>2</b>
<b>Where to add the External Functions?.....</b>	<b>2</b>
<b>Simple example.....</b>	<b>3</b>
<b>Summary of functions.....</b>	<b>3</b>
<b>Creating Thumbnail Images</b>	<b>4</b>
<b>Steps for creating thumbnails.....</b>	<b>4</b>
<b>1-Define fields.....</b>	<b>4</b>
<b>2-Create a thumbnail layout .....</b>	<b>5</b>
<b>3-Create thumbnailer scripts .....</b>	<b>5</b>
<b>4-Create thumbnails in a loop .....</b>	<b>6</b>
<b>Function Reference</b>	<b>7</b>
<b>Trgr-RGBToClip</b>	<b>7</b>
<b>Trgr-ScreenToClip</b>	<b>9</b>
<b>Trgr-Version</b>	<b>11</b>

## Installing plug-ins

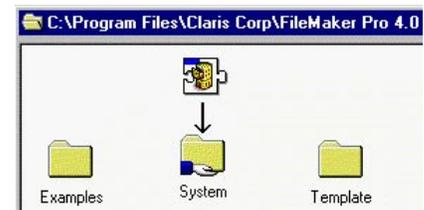
### For Macintosh:

- Quit FileMaker Pro.
- Put the file "Troj Graphic Plug-in" from the folder "MacOS Plug-in" into the "FileMaker Extensions" folder in the FileMaker Pro folder.
- If you have installed previous versions of this plug-in, you are asked: "An older item named "Troj 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 Troj 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 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 Troj 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 "All Graphic Examples.fp5" to see how to use the plug-in's functions. There is also a Function overview available.

## If You Have Problems

This user guide tries to give you all the information necessary to use this plug-in. So if you have a problem please read this user guide first. If that doesn't help you can get free support by email. Send your questions to [support@troj.com](mailto:support@troj.com) with a full explanation of the problem. Also give as much relevant information (version of the plug-in, which platform, version of the operating system, version of FileMaker Pro) as possible.

If you find any mistake in this manual or have a suggestion please let us know. We appreciate your feedback!

**TIP** You can get more information on returned error codes from our OSErrrs database on our web site: <http://www.troj.com/software/oserrrs.html>. This free FileMaker database lists all error codes for Windows and Mac OS!

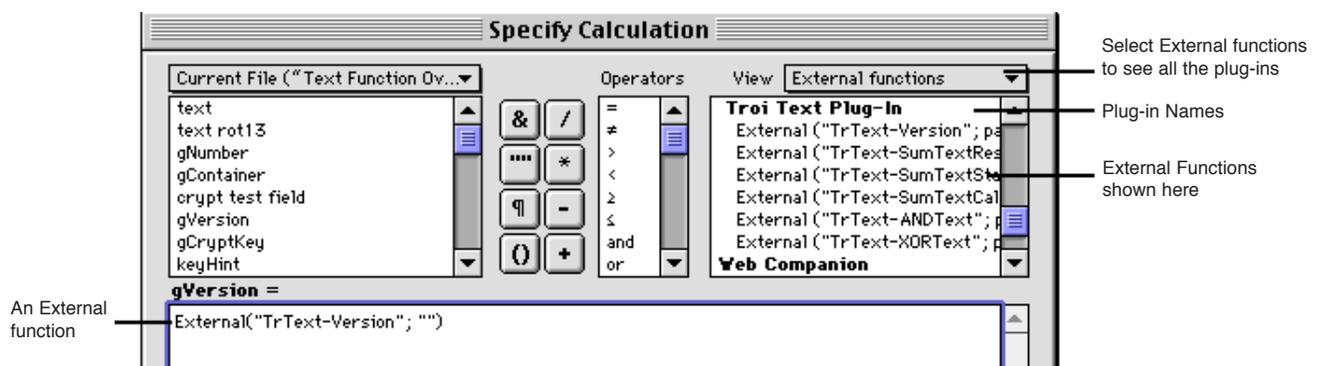
# What can this plug-in do?

The Troi Graphic Plug-in adds graphic functions to FileMaker Pro. With this plug-in you can fill container fields with any RGB color you want, capture screenshots and make thumbnails of images in the database.

## Getting started

### Using external functions

The Troi Graphic Plug-in adds new functions to the standard functions that are available in FileMaker Pro. The functions added by a plug-in are called external functions. You can see those extra functions for all plug-ins at the top right of the Specify Calculation Box:



You use special syntax with external functions: `External("function name", parameter)` where function name is the name of an external function. The parameter is required, even if it's only "". Plug-ins don't work directly after installation. To access a plug-in function, you need to add the calls to the function in a calculation for example in a text calculation in Define Fields or in a ScriptMaker Script.

**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 separator used when the file was created. All examples show the functions with commas. For example: `External("Trgr-Version", "")` will become `External("Trgr-Version"; "")` in such a file.

### Where to add the External Functions?

External functions for this plug-in are intended to be used in a script step using a calculation. For most functions of this plug-in it makes no sense to add them to a define field calculation, as the functions will have side effects.

## Simple example

We start with a simple example to get you started. This example can also be found in the folder "Simple Example from User Guide". It will fill a container field with a blue image. Create a new database myRGB.fp5, with a global text field called gErrorCode and a container field called myContainer, in which you are going to store the IP address of the computer. Create a new ScriptMaker Script called "Simply Blue". Delete all steps and then add the following script steps:

```
Set Field [ gErrorCode, External("Trgr-RGBToClip", "-unused|0|0|255")]
Paste [myContainer]
```

Performing this script will put a small blue rectangle on the clipboard and put it into the container field. You can modify the container field to enlarge the rectangle, so you can use it as a fill color.

**NOTE** Function names, like Trgr-RGBToClip are case sensitive. Be sure to spell them right, or get them from the External Functions list at the top right of the "Specify Calculation" dialog.

Please take a close look at the included example files, as they provide a great starting point. From there you can move on, using the functions of the plug-in as building blocks. Together they give you great new tools!

## Summary of functions

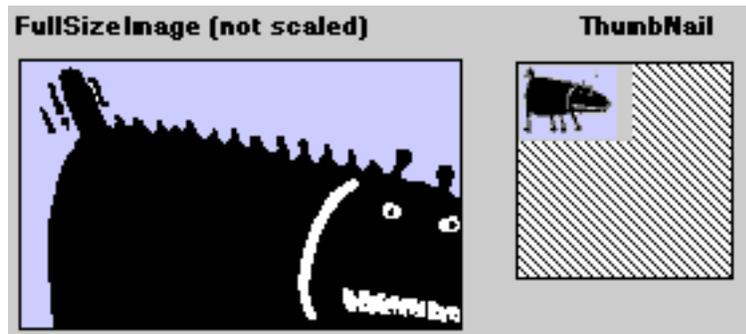
The Troi Graphic Plug-in adds the following functions:

<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

## 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 already have large images in a FileMaker Pro database there is no easy way to 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 thumbnail images:

- 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:

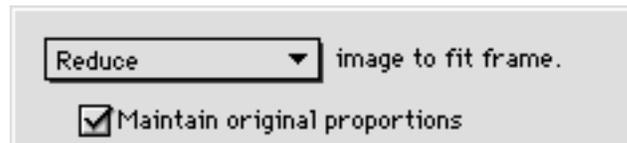
thumbnail	Container	
gErrorCode	Global	Number
gGlobalFieldBounds	Global	Text
gLocalFieldBounds	Global	Text
gLeft	Global	Number
gTop	Global	Number
gRight	Global	Number
gBottom	Global	Number
gLeftOffset Mac	Global	Number
gTopOffset Mac	Global	Number
gLeftOffset Win	Global	Number
gTopOffset Win	Global	Number

We assume that this field is already present in the database:

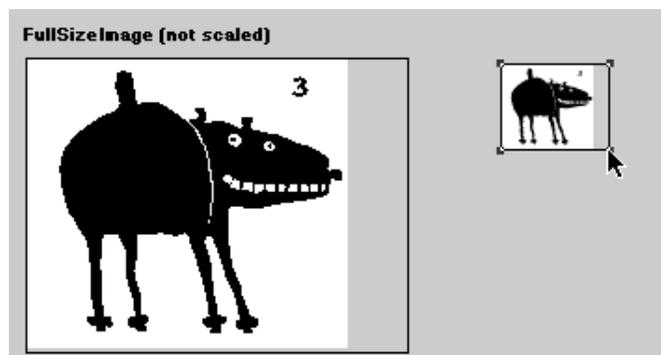
```
imageField Container
```

## 2- Create a thumbnail 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 [Abs(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|" & gGlobalField-
Bounds)]
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 all records in the current found set.

# Function Reference

## Trgr-RGBToClip

**Syntax**      Set Field [ result, External("Trgr-RGBToClip", "switches| red | green | blue" )]

Puts a small square (8x8) of the specified color 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 color, use a number between 0 and 255*  
*green: the value of the green component of the color, use a number between 0 and 255*  
*blue: the value of the blue component of the color, use a number between 0 and 255*

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

### Returned result

If successful it returns 0. If unsuccessful it returns an error code starting with \$\$ and the error code.

Possible error codes are:

0	no error
\$\$-108	memFullErr, ran out of memory

Other errors may be returned.

### Special considerations

The original contents of the clipboard is lost.

### Example usage

```
Set Field[gErrorCode, External("Trgr-RGBToClip","-unused| 255| 0| 0")]
```

This will put a red square on the clipboard.

### Example 2

Define the following fields:

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

## Trgr-RGBToClip

Create a ScriptMaker Script "Color to Container" and define these 2 steps:

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

This will put a square on the clipboard with the colors specified by the global fields.

If you fill the numbers with red = 150, green = 0 and blue = 150 is the result will be a purple color.

# Trgr-ScreenToClip

**Syntax**      Set Field [ result, External("Trgr-ScreenToClip", "switches| left | top | right | bottom" )]

Puts the specified part of the screen on the clipboard.

## Parameters

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

*left: the left co-ordinate of the area to capture*

*top: the top co-ordinate of the area to capture*

*right: the right co-ordinate of the area to capture*

*bottom: the bottom co-ordinate of the area to capture*

## Returned result

If successful it returns 0. If unsuccessful it returns an error code starting with \$\$ and the error code.

Possible error codes are:

0	no error
\$\$-108	memFullErr, ran out of memory

Other errors may be returned.

## Special considerations

The original contents of the clipboard is lost.

## Example usage

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

This will return the upper left portion of the screen on the clipboard.

## Example 2

Define the following fields:

gLeft	Global, Number	screenImage	Container
gTop	Global, Number	gErrorCode	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]
```

## Trgr-ScreenToClip

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(CurrentScreenHeighth)
```

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

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

# Trgr-Version

**Syntax**      Set Field [ result, External("Trgr-Version", "switches") ]

Use this function to see which version of the plug-in is loaded.

Note: This function is also used to register the plug-in.

## Parameters

*determine the behaviour of the function*

*switches can be one of this:*

- GetVersionString      the version string is returned (default)*
- GetVersionNumber      Returns the version number of the plug-in*
- ShowFlashDialog      Shows the Flash Dialog of the plug-in (returns 0)*

*If you leave the parameter empty the version string is returned.*

## Returned result

The function returns "" if this plug-in is not loaded. If the plug-in is loaded the result depends on the input parameter. It is either a:

VersionString:

If you asked for the version string it will return for example "Troj Graphic Plug-in 1.1"

VersionNumber:

If you asked for the version number it returns the version number of the plug-in x1000. For example version 1.3 will return number 1300.

ShowFlashDialogResult:

This will show the flash dialog and then return the error code 0.

## Special considerations

**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.

## Example usage

External(Trgr-Version, "") will for example return "Troj Graphic Plug-in 1.1".

## Example 2

External("Trgr-Version", "-GetVersionNumber") will return 1100 for version 1.1.

External("Trgr-Version", "-GetVersionNumber") will return 1101 for version 1.1b1

External("Trgr-Version", "-GetVersionNumber") will return 2130 for version 2.1.3

So for example to use a feature introduced with version 1.1 test if the result is equal or greater than 1100.