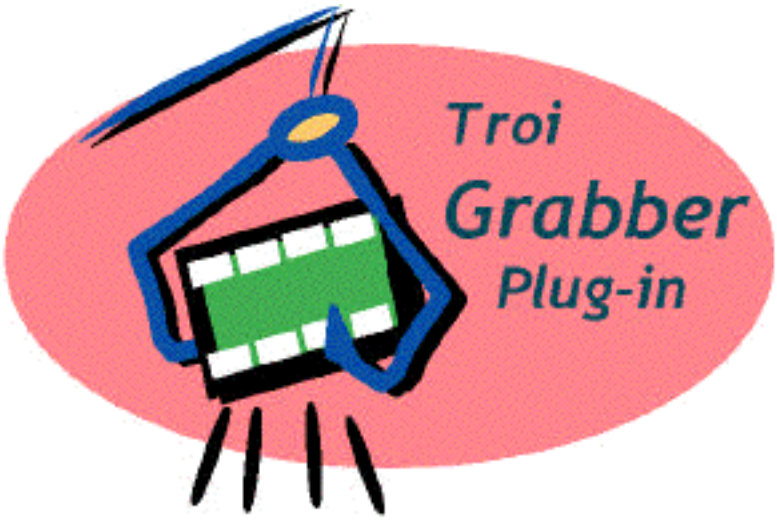


Troi Grabber Plug-in 1.5 USER GUIDE

April 2002

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.

Table of Contents

Click on a line to jump to that subject

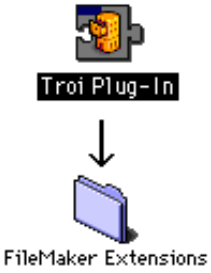
Installing plug-ins..... 1
If You Have Problems..... 1
What can this plug-in do?..... 2
Getting started 2
Using external functions..... 2
Where to add the External Functions?..... 2
Simple example..... 3
Summary of functions..... 3
Implementing Picture Grabbing 4
Plug-in Limitations..... 5
Steps for creating a picture grabbing database (Mac OS) 6
Steps for creating a picture grabbing database (Windows) 8
Rotation and Cropping (Mac only) 9
Implementing Rotation and Cropping 9
Getting a cropped image into a container..... 11
Recording Movies (Mac only) 13
Inserting Movies into FileMaker (Mac only) 14

Function Reference 15
Grab-AquireToClip 15
Grab-DisplayCropRect 16
Grab-DoSettingsDialog 17
Grab-GetSettings 18
Grab-GetTimecode 19
Grab-ImageToClip 21
Grab-Initialise 22
Grab-RecordMovie 23
Grab-SetRotation 24
Grab-SetSettings 25
Grab-StartPreview 26
Grab-Stop 27
Grab-Version 28

Installing plug-ins

For MacOS:

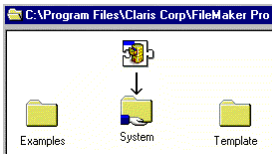
- Quit FileMaker Pro.
- Put the file “Troi Grabber Plug-in” from the folder “Mac OS Plug-in” into the “FileMaker Extensions” folder in the FileMaker Pro folder.



- If you have installed previous versions of this plug-in, you are asked: “An older item named “Troi Grabber 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 Grabber Plug-in is used it will display a Grabber box, indicating that it is loading and showing the registration status.

For Windows:

- Quit FileMaker Pro.
- Put the file "grabber.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 'grabber.fmx'. Would you like to replace the existing file with this one?’. Press the Yes button.
- Start FileMaker Pro. The Troi Grabber 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 Grabber Examples.fp5" to see how to use the plug-in's functions. There is also a function overview.

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. Also you might visit our support web page:

<http://www.troi.com/support/>

This page contains FAQ's (Frequently Asked Questions), help on registration and much more. If that doesn't help you can get free support by email. Send your questions to **support@troi.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.troi.com/software/oserrrs.html>. This free FileMaker database lists all error codes for Windows and Mac OS!

What can this plug-in do?

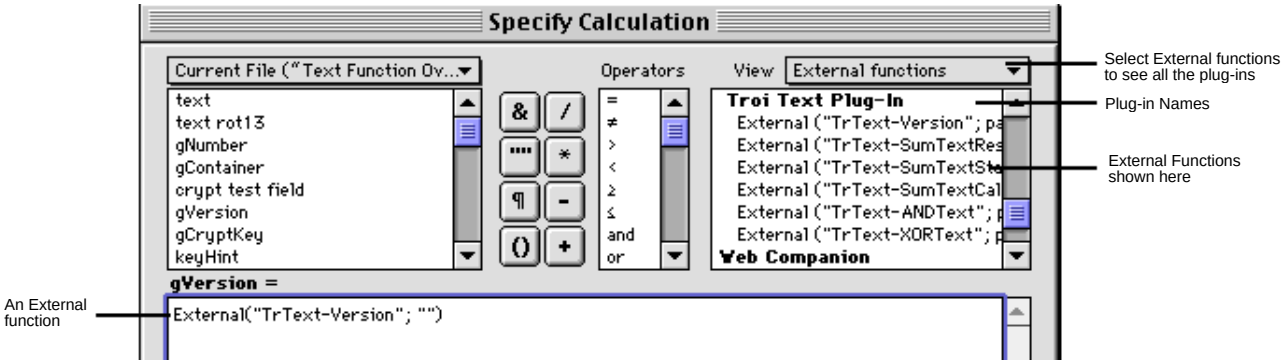
The Troi Grabber Plug-in is a very powerful tool which adds Video Image Grabbing functions to FileMaker Pro. With it you can take a picture from a video camera and put it into a container field. On Mac OS X you can also record a movie from a video source.

IMPORTANT On Mac you need QuickTime (version 2.5 or higher) installed and a video input source that is compatible with QuickTime. Examples of these inputs are the built in video-in port on AV Macs and the (color) Quickcam from Connectix. On Windows you need an input device with a TWAIN driver.

Getting started

Using external functions

The Troi Grabber 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("Grab-Version", "")` will become `External("Grab-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

This example shows how you begin using the Troi Grabber Plug-in. You need to work with more functions in a real database, but this is shown later.

Say you have a database myGrabTest.fp5, with a global text field called gDimensions. In ScriptMaker create a script "Initialise". Add the following script step to this script:

```
Set Field[gDimensions, External("Grab-Initialise", "") ]
```

This will initialise the Grabber Plug-in. On Windows this will return 0 or an error code. On Mac this function will return the width and height of the video source, or an error code. For example it might return "640|480". In the script use this as the basis for determining to how to proceed.

IMPORTANT Function names, like `Grab-Initialise` 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" Grabber. Also make sure you don't use a capital i instead of a lowercase l in the word `Initialise`.

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 all the tools you need to grab images and movies directly from FileMaker Pro.

Summary of functions

The Troi Grabber Plug-in adds the following functions:

function name	short description
Grab-Version	Use this function to see which version of the plug-in is loaded. This function is also used to register the plug-in.
Grab-Initialise	Initialises the plug-in and checks for available input sources.
Grab-StartPreview	(Mac only) Begins showing the preview of the image to be captured.
Grab-ImageToClip	(Mac only) Captures a picture and puts it in the clipboard buffer.
Grab-DoSettingsDialog	Shows a video settings dialog.
Grab-GetSettings	(Mac only) Retrieves the current video settings from the plug-in.
Grab-SetSettings	(Mac only) Sets the video settings to the passed parameter.
Grab-Stop	Stops the video preview.
Grab-AquireToClip	(Win only) Aquires a picture and puts it in the clipboard buffer.
Grab-SetRotation	(Mac only) Rotates the preview.
Grab-DisplayCropRect	(Mac only) Displays a cropping rectangle on the preview.
Grab-RecordMovie	This will record the movie and write it to the specified file.
Grab-GetTimecode	This will return timecodes of a movie.

Implementing Picture Grabbing

Implementing video grabbing it not difficult, but the developer of the database must be aware of some limitations to get optimum results.

Plug-in Limitations

The Grabber plug-in has some limitations. Please be aware of the following:

General Limitations

- **Platform Differences:** The plug-in implementation is different for Mac and Windows. The commands for the Windows version are slightly different from the Mac OS version. The Mac version is based on QuickTime, while the Windows version is based on TWAIN drivers.

Limitations for Mac OS

- **Preview only when window is on top**: The video preview is only shown and updated when FileMaker is the frontmost application. Also the window where the preview is shown must be frontmost. If you switch to a different file or application the video preview is no longer updated.

- **FileMaker is unaware of the video preview**: When you start a video preview, the plug-in shows the image stream in the window. The FileMaker application is not aware that this is happening. There is no fixed relation to the preview rectangle and the other FileMaker objects on the screen. So when a user hides the status bar the FileMaker objects move to the left but the preview rectangle is still shown on the same coordinates. This is not serious, but looks very strange and may confuse a user. Therefore before starting a video preview, be sure to lock the window down. Use these script steps:

```
Toggle Status Area [Hide, Lock]
Set Zoom Level [100%, Lock]
```

It doesn't matter if the status area is hidden or shown, but make sure to lock it. The same holds true for the Zoom level; choose a fixed zoom level and lock it.

- **Video Preview is not aware of the layout**: Make sure to stop the video preview when the user leaves the layout. The preview rectangle is still shown on the same coordinates if you don't do this. This is also not serious, but looks very strange and may confuse a user. Also note that when switching to Layout mode the preview can continue to be shown. At the moment we don't have a way to prevent this surprising but harmless behaviour.

- **Video Preview must fit on the window**: Make sure the video preview is completely visible on the screen. Otherwise the capture is only of the visible part of the rectangle.

Limitations for Windows

- **No movie recording on Windows**: At the moment the grabbing on Windows is limited to pictures only. You can only grab a movie on Mac OS and Mac OS X.

- **TWAIN drivers may be problematic**: Working with some TWAIN drivers has proved quite difficult, so the Windows plug-in may not work in your situation. Always test very careful, save other work first and test this software using a test database. Use at your own risk!

- **No Preview in Window**: Some TWAIN drivers don't allow to show a preview in the FileMaker window. Therefore this had to be implemented different with the AquireToClip command.

- **Aquire Image Window stays black**: With some TWAIN drivers the preview image stays black or is not updated after moving the Aquire window.

- **Aquire Image Window can be put behind the FileMaker window**: The Aquire window may be put behind the FileMaker window..

- **Camera needs to be on**: Badly written TWAIN drivers may not report the fact that the camera is not connected. This might even crash the system. The plug-in can't detect this. Therefore be sure the camera is connected properly.

- **No way to set or get settings**. At this moment we don't have a way to get or set the TWAIN settings.

Steps for creating a picture grabbing database (Mac OS)

These are the main steps to create a grabbing database:

1 - create a picture container field and some assisting global fields.
2 - create a new layout with room for the video input rectangle and the picture field on it. You can also modify an existing layout.
3 - create 3 scripts: to start the preview, to capture an image and to stop the preview.
4 - if wanted you can do this in a loop so you grab pictures in sequence.

1- Define Fields

In your database define the following fields:

```
image           Container field
gErrorCode      Global, Text
gWidthHeight    Global, Text
gWidth          Global, Text
gHeight         Global, Text
```

2- Create a Grabber layout

Create a layout or modify an existing layout. Make room for the video preview image. You can do this by trying out the preview and adjusting the scripts and layout.

3- Create two Grabber scripts

In ScriptMaker define a script "Start Preview (Mac)". This script will get the width and height from the video source and starts the video preview:

Define the script "Start Preview (Mac)" as follows:

```
Comment [Get the maximum dimensions of the video camera]
Set Field [gWidthHeight, External("Grab-Initialise", "")]
If [Left(gWidthHeight , 2) <> "$$"]
     Comment [there was no error; split the 2 dimensions]
     Set Field [gWidth, Left(gWidthHeight , Position( gWidthHeight , "|" , 1 , 1 ) - 1)
     Set Field [gHeight, Right(gWidthHeight , Position( gWidthHeight , "|" , 1 , 1 ) +
     Comment [start the preview 10 pixels from the left and 100 pixels from the top...]
     Comment [... with the width and height being half of the maximum of the camera]
     Set Field [gErrorCode,
              External("Grab-StartPreview", "10|100|" & gWidth /2 & "|" & gHeight / 2)]
Else
     Comment [An error occurred. put the errorcode into the gErrorCode global.]
     Set Field [gErrorCode, gWidthHeight]
     If [gErrorCode = "$$-9405"]
         Beep
         Show Message [No video component was found on this computer.]
     Else
         Beep
         Show Message [An error occured while initializing the video input.]
     End If
End If
```

In ScriptMaker also define a script "New record and Grab Image To Clipboard" as follows:

```
Set Field [gErrorCode, External("Grab-ImageToClip", "")]
If [gErrorCode <> 0]
     Beep
     Show Message [An error occurred. Make sure you start the preview]
     Halt Script
End If
New Record/Request
Paste [Select, image]
Go to Field []
```

This script will capture the current preview image in a container in a new record. You can of course change this to paste it in an existing container.

4- Grab Pictures in a loop

If you want to grab images in a loop define a script "Loop: Grab Images into records" as follows:

```
Go to Layout [Image grabbing]
Loop
     Perform Script [Sub-scripts, New record and Grab Image To Clipboard]
     Pause/Resume Script [0:00:10]
     Exit Loop If [gErrorCode <> 0]
End Loop
```

This will grab a picture every 10 seconds.

Steps for creating a picture grabbing database (Windows)

Implementing video grabbing on Windows is slightly different.
These are the main steps to create a grabbing database on windows:

1 - create a picture container field and some assisting global fields.
2 - create a new layout with the picture field on it. You can also modify an existing layout. (On windows the space for the video input rectangle is not needed).
3 - create 2 scripts: to start the image aquiring and to stop it again.

1- Define Fields

In your database define the following fields:

```
image           Container field
gErrorCode      Global, Text
```

2- Create a Grabber layout

Create a layout or modify an existing layout. Put the the picture field image on it.

3- Create two Grabber scripts

In ScriptMaker define a script "Aquire Image (Windows)". This will initialise the video source and show the aquire image preview:

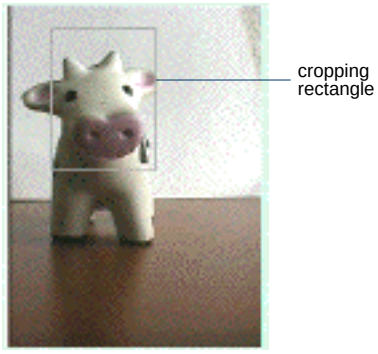
Define the script "Aquire Image (Windows)" as follows:

```
Comment [Initialise the TWAIN driver]
Set Field [gErrorCode, External("Grab-Initialise", "")]
If [Left(gErrorCode , 2) = "$$"]
     Comment [An error occurred.]
     Beep
     Show Message [An error occurred while initializing the video input.]
     Halt Script
Else
     Comment [there was no error]
     Comment [show the Aquire image window]
     Set Field [gErrorCode, External("Grab-AquireToClip", "")]
End If
If [Left(gErrorCode , 2) = "$$"]
     Comment [An error occurred.]
     Beep
     Show Message [An error occurred while with AquireToClip.]
     Halt Script
Else
     Comment [there was no error]
     New Record/Request
     Paste [Select, image]
     Go to Field []
End If
```

This script will show the Aquire dialog box and the user can then grab a picture. This picture is then pasted in the container field of a new record. You can of course change this script to paste it in an existing container.

Rotation and Cropping (Mac only)

The following features of Troi Grabber plug-in are only available for Mac OS, as they make use of Quicktime. You need version 1.1 or later of the Grabber plug-in for these advanced features. In the picture to the right a sample preview is showing a video preview that is rotated 90° and it shows a cropping rectangle.



Implementing Rotation and Cropping

The Troi Grabber Plug-in for Mac OS can rotate the video preview and crop images. Implementing these features require that you add some special steps to your ScriptMaker scripts.

These are the main steps to take to enable these features:

1 - Initializing the video preview using a "GWorld"
2 - Set the rotation
3 - Initialize the cropping rectangle

NOTE You don't need to implement both step 2 and 3. You can also use rotation or cropping seperately.

What is a GWorld?

The cropping and rotation functions are implemented with the help of a GWorld. A GWorld is a technical term: it is an offscreen part of the computer memory that makes it possible to manipulate the grabbing process.

Initializing the video preview

You don't need to remember what a GWorld is. The main thing to remember is that for Rotation and Cropping you need to initialise the Grabber Plug-in using the `useGWorld` tag, like this:

```
Set Field [gWidthHeight, External("Grab-Initialise", "useGWorld"))]
```

TIP A GWorld uses extra memory, so for large video images you might need to give FileMaker Pro more memory.

Setting the rotation

If you want to rotate the video image generated you use the function `Grab-SetRotation`. Here is a sample script `Set rotation 90`:

```
Comment [Set rotation to 90 degrees]
Set Field [gErrorCode, External("Grab-SetRotation", "90")]
```

Initializing the cropping rectangle

If you want to show a cropping rectangle use the function `Grab-DisplayCropRec`t. Here is a sample script `Show Crop Rectangle`:

```
Comment [Show a rectangle at the specified coordinates]
Set Field [gErrorCode, External("Grab-DisplayCropRect", "100|50|240|320")]
```

The parameters for the `Grab-DisplayCropRect` function are the 4 coordinates of the cropping rectangle in this format: `"left | top | width | height "`. The coordinates are relative to the video source: left and top are the coordinates (in pixels) in the source of the upper left part of the rectangle. width and height give the dimension (in pixels) of the rectangle.

Example script

The following script `Start Preview Mac RotCrop` shows the elements together:

```
Comment [Start up the preview of the image to be grabbed]
Set Field [gWidthHeight, External("Grab-Initialise", "useGWorld"]
If [Left(gWidthHeight , 2) <> "$$"]
       Set Field [gWidth, Left(gWidthHeight , Position(gWidthHeight , "|" , 1 , 1 ) - 1
       Set Field [gHeight, Right(gWidthHeight , Position(gWidthHeight , "|" , 1 , 1 ) +
       Set Field [gErrorCode,
                 External("Grab-StartPreview", "10|100|" & gHeight / 2 & "|" & gWidth /
       Perform Script [Sub-scripts, Set Grabber Settings]
       Perform Script [Sub-scripts, Set rotation 90]
       Perform Script [Sub-scripts, Show Crop Rectangle]
Else
       Comment [An error occurred. put the errorcode into the global.]
       Set Field [gErrorCode, gWidthHeight]
End If
```

After this script has run the preview should be visible, rotated 90° and with a cropping rectangle.

Getting a cropped image into a container

When you are previewing you can grab the complete image, as well as a part of the image into a container field.
There are 2 methods to crop images:

1 - Crop the image on the clipboard
2 - Get a cropped image from the video preview

Use the first method if it is important that the cropped image is exactly the same as (the part of) the larger image. If you use the second way the cropped image is grabbed freshly from the video source and will be the image that is then showing.

Scenario 1

Storing 2 identical images: 1 full size and 1 cropped

If you want the grabber to grab an image and also store a part of that same image, use this script.

"New record + store 2 identical images: 1 Full + 1 Cropped":

```
Comment [# 1- get the full size image from the grabber into the clipboard...]
Set Field [gErrorCode, External("Grab-ImageToClip", "")]
If [gErrorCode <> 0]
  Beep
  Show Message [An error occurred. Make sure you start the preview before
                grabbing an image.]
  Halt Script
End If
New Record/Request
Comment [... paste it into the 1st container field.]
Paste [Select, image]
Comment [# 2- crop the image on the clipboard....]
Perform Script [Sub-scripts, CropClipboard]
Comment [... then paste it into the 2nd container field.]
Paste [Select, image Detail]
Comment [Leave the record to save it to disk:]
Go to Field []
```

This is the script `CropClipboard`:

```
Comment [Crop an image on the clipboard to the requested size.]
Set Field [gErrorCode, External("Grab-CropClip", "100|50|240|320")]
If [gErrorCode <> 0]
  Beep
  If [gErrorCode = "$$-4207"]
        Show Message [An error occurred. There was no picture in the clipboard.]
  Else
        Show Message [An error occurred. There was not enough memory or you have
                      specified incorrect parameters in this script.]
  End If
  Halt Script
End If
```

TIP In this example the parameters for Grab-CropClip are hardcoded into the script, but it would be more flexible to store this into global fields.

Scenario 2

Getting full size image and cropped image after each other

If you want the grabber to grab a full size image first then get a cropped image after that use this script.

"New record and Grab 2 Images To Clipboard":

```
Comment [# 1- get the full size image from the grabber into the clipboard...]
Set Field [gErrorCode, External("Grab-ImageToClip", "")]
If [gErrorCode <> 0]
  Beep
  Show Message [An error occurred. Make sure you start the preview before grabbing an
               image.  ]
  Halt Script
End If
New Record/Request
Comment [... paste it into the 1st container field.]
Paste [Select, image]
Comment [# 2- get a cropped image from the grabber on the clipboard....]
Set Field [gErrorCode, External("Grab-ImageToClip", "cropped")]
Comment [... then paste it into the 2nd container field.]
Paste [Select, image Detail]
Go to Field []
```

Recording Movies (Mac only)

The function `Grab-RecordMovie` can record a video stream to a movie file. This is only available on Mac OS and Mac OS X, as this uses of Quicktime. You need version 1.4 or later of the Grabber plug-in for these advanced features.

- Recording only works if no GWorld is used. So make sure you don't use the switch `useGWorld` when you intialise the grabber with the function `Grab-Initialise`.

- The Troi Grabber plug-in can not select or manipulate files. Our Troi File plug-in makes it easy to work with movie files on disk. You can download a demo copy from our web site on this page:

<http://www.troi.com/software/fileplugin.html>

These are the main steps to create a database that can record movies:

1 - create a picture container field and some assisting global fields.
2 - create a new layout with room for the video input rectangle and the picture field on it. You can also modify an existing layout.
3 - create 3 scripts: to start the preview, to record a movie and to stop the preview.

Steps 1 to 3 are the same as the steps for creating a picture grabbing database with the exception of the script that records a movie. See "Steps for creating a picture grabbing database (Mac OS)" on page 6 for all these steps.

NOTE if you implement all steps of "Steps for creating a picture grabbing database (Mac OS)" and the script below you can make a database that can grab single images and also record a movie too.

Script variant 1: Record Movie until mouse click

This script will record until the user clicks on the mouse. In addition to the existing field in your database define a global text field `gInfoText`. This field will indicate to the user what to do. ScriptMaker also define a script "Record Movie until mouse click" as follows:

```
#This will create movie file and record into it until the user clicks on the mouse.
Enter Browse Mode []
Set Field [gInfoText, "Recording movie. Click the mouse to stop recording."]
Refresh Window []
#The next step is to make sure that on Mac OS X the screen is updated correctly:
Pause/Resume Script [0:00:00]
Set Field [gErrorCode, External("Grab-RecordMovie", "-RecordUntilMouseClick " &
                                gOverwriteExistingFiles & " |" & gPathToMovieFile)]
Set Field [gInfoText, ""]
If [gErrorCode = 0]
     Show Message [The movie has been recorded and written to disk.]
Else
     Beep
     Show Message [An error occurred.]
End If
```

Script variant 2 : Record Movie during a specified time

This script will record for a specified number of seconds. In addition to the existing field in your database define a global text field `gInfoText`. This field will indicate to the user what to do. ScriptMaker also define a script "Record Movie until mouse click" as follows:

```
#This will create movie file and record 10 seconds of movie.
Enter Browse Mode []
Set Field [gInfoText, "Recording 10 secs. of movie until: "  &
             TimeToText(Status(CurrentTime) + 10) &". Press ESC to stop earlier."]
Refresh Window []
Pause/Resume Script [0:00:01]
Set Field [gErrorCode, External("Grab-RecordMovie", "-RecordSpecifiedTime" &
                    gOverwriteExistingFiles & "|" & gPathToMovieFile & "|10")]
Set Field [gInfoText, ""]
If [gErrorCode = 0]
     Show Message [The movie has been recorded and written to disk.]
Else
     Beep
     Show Message [An error occurred.]
End If
```

Inserting Movies into FileMaker (Mac only)

If you want to insert the movies you created by the Grabber plug-in into a FileMaker database you need the help of the Troi File Plug-in. It can insert multiple movie files into FileMaker Pro with a script. You can download a demo copy from our web site on this page:

<http://www.troi.com/software/fileplugin.html>

NOTE FileMaker Pro can only insert movies as a reference.

To get more information of the movie use the Troi Grabber plug-in. It can get timecodes of a movie file. See the example file "Insert1Movie.fp5" to see how to implement this.

Function Reference
Grab-AquireToClip

Syntax Set Field [result, External("Grab-AquireToClip", "")]

This shows the aquire dialog box. The live video image preview is shown in a separate window. There you can click on the “Aquire image” button. The image is put on the clipboard.

Parameters

no parameters *leave empty for future use.*

Returned result

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

Special considerations

This function is only available on Windows.

Example usage

External("Grab-AquireToClip", "")

Grab-DisplayCropRect

Syntax Set Field [result, External("Grab-DisplayCropRect", "left | top | width | height")]

This tells the grabber show a cropping rectangle on the video source.

Parameters

coordinates are relative to the video source:
left and top are the coordinates (in pixels) in the source of the upper left part of the rectangle.
width and heigth give the dimension (in pixels) of the rectangle.

Returned result

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

Special considerations

This function is currently only available on Mac OS and Mac OS X.

Example usage

External("Grab-DisplayCropRect", "100|50|240|320")

Grab-DoSettingsDialog

Syntax Set Field [result, External("Grab-DoSettingsDialog", "switch")]

This shows the video settings dialog. With this dialog the user can change the settings of the video source.

Parameters

switch *set to "-video". No other switches are supported at this moment.*

Returned result

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

Special considerations

After the user clicks on the OK button the settings are changed. You can retreive the current settings with the function "Grab-GetSettings".

Example usage

External("Grab-DoSettingsDialog", "-video")

This will show the Video Settings dialog.

Grab-GetSettings

Syntax Set Field [result, External("Grab-GetSettings", "")]

This returns the current video settings.
Important: The video preview must be running first.

Parameters

none *please leave blank for future use*

Returned result

If successful this function returns the current settings for the video input. If unsuccessful it returns an error code starting with $$ and the error code.

The setting is coded and can't be read. Store the result for later use in (for example) a global field.

Special considerations

Do not change the returned settings. Only store them in a field so you can reset them later.

Example usage

Set Field[gGrabberSettings, External("Grab-GetSettings", "")]

Grab-GetTimecode

Syntax Set Field [result, External("Grab-GetTimecode", "switches |FileSpec")]

This will return a timecode of the movie.

Parameters

switches	*these determine which timecode is returned*
FileSpec	*the FileSpec or path to the movie file to create*

switches must be one of the following:

-StartOfMovie	*get the timecode of the beginning of the movie*
-EndOfMovie	*get the timecode of the end of the movie*
-CurrentTime	*get the timecode at the current point in the movie*
-Duration	*get the duration of the movie in timecode*

Returned result

If successful it returns a timecode. If unsuccessful it returns an error code starting with $$ and the error code. Possible error codes are:

$$-2020	QuickTime not available
$$-2032	No timecode track found
$$-43	The file was not found
$$-37	The name of the file is not correct

Other errors may be returned.

Special considerations

This function is currently only available on Mac OS and Mac OS X.
If the movie does not contain a timecode track only the duration timecode can be returned.

Example usage

Set Field [result, External("Grab-GetTimecode", "-StartOfMovie |Mac HD:projects:test.mov")]

This will get the timecode of the start of the specified movie, for example "01:23:59:29"

Example 2

This example will put the timecodes into fields. We assume that in your FileMaker file the following fields are defined:

TimeCode Start	Text
TimeCode End	Text
TimeCode Duration	Text
gFilePath	Global, text

gFilePath should contain the path to the movie file, for example "Mac HD:movies:shrek.mov". In ScriptMaker add the following script steps:

```
Set Field[TimeCode Start, External(""Grab-GetTimecode", "-StartOfMovie | & gFilePath]
If[ Timecode Start = "$$-2032"]
```

Grab-GetTimecode

```
    # no time code found
     Exit Script
  Else
    Set Field[TimeCode End, External(""Grab-GetTimecode", "-EndOfMovie | & gFilePath]
    Set Field[TimeCode Duration, External(""Grab-GetTimecode", "-Duration | & gFilePath]
  End if
```

This will get the timecodes for the movie specified by gFilePath. If the movie does not have a timecode track an error code of "$$-2032" will be returned.

Grab-ImageToClip

Syntax Set Field [result, External("Grab-ImageToClip", "")]

This puts current image from the video preview on the clipboard.

Parameters

no parameters *leave empty for future use.*

Returned result

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

Example usage

External("Grab-ImageToClip", "")

Grab-Initialise

Syntax Set Field [result, External("Grab-Initialise", "switches")]

Use this function to test if there if a video input source available and initialise it.

Parameters

switches *leave blank or see Rotation/Cropping in this overview*

Returned result

Win: If successful it returns 0.
Mac: If unsuccessful it returns an error code starting with $$ and the error code. Possible codes are:
$$-2020 QuickTime not available
$$-230 No video input device found

Example usage

Set Field [result , External("Grab-Initialise", "")]

This initialises the plug-in. On Windows this will return 0 or an error code. On Mac this will return the width and height of the video source, or an error code. For example this might return "640|480".

Example 2

External("Grab-Initialise", "useGWorld useAltBounds")

Grab-RecordMovie

Syntax Set Field [result, External("Grab-RecordMovie", "switches |FileSpec|Seconds")]

This will record the movie and write it to the specified file.

Parameters

switches	*these determine how this function works*
FileSpec	*the FileSpec or path to the movie file to create*
Seconds	*(optional) the number of seconds of movie that will be recorded.*

switches can be one of the following:

-RecordUntilMouseClick	*(default) record until the user clicks the mouse*
-RecordSpecifiedTime	*record for the specified number of seconds*

You can also add this switch:

-OverwriteExisting	*overwrite an existing file without generating an error*

Returned result

If successful it returns 0. If unsuccessful it returns an error code starting with $$ and the error code. Possible error codes are:

$$-2020	QuickTime not available
$$-2042	dataNotOpenForRead , start the preview first
$$-48	The file already exists

Other errors may be returned.

Special considerations

This function is currently only available on Mac OS and Mac OS X.

Example usage

Set Field [result, External("Grab-RecordMovie"; "-RecordUntilMouseClick |Mac HD:test.mov")]

This will record a movie "test.mov" on the harddisk. The movie will be recorded until a mouse is clicked.

Example 2

We assume that in your FileMaker file the following fields are defined:

gFilePath	Global, text	gRecordTime	Global, time

gFilePath should contain the path to the movie file, for example "Mac HD:movies:test.mov". gRecordTime should contain the number of seconds to record. In ScriptMaker add the following script step:

Set Field[gErrorCode, External("Grab-RecordMovie"; "-RecordSpecifiedTime |"& gFilePath & "|" & gRecordTime)]

This will record a movie at the place specified by gFilePath. The movie will be record until a mouse for the number of seconds.

Grab-SetRotation

Syntax Set Field [result, External("Grab-SetRotation", "rotationangle")]

This tells the grabber how to rotate the video preview.

Parameters

rotationangle *the number of degrees the image must be rotated. The rotation can only be 90, -90 and 180.*

Returned result

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

Special considerations

This function is currently only available on Mac OS and Mac OS X.

Grab-SetSettings

Syntax Set Field [result, External("Grab-SetSettings", "videoSettings")]

This sets the video settings to the given video settings.

Parameters

videoSettings *Use only a video setting data you got as a result from "Grab-GetSettings"*

Important: Do not use changed settings.

Returned result

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

Special considerations

Important: Use only the video setting data you got as a result from "Grab-GetSettings". This data is in a special format and can't be changed.

Example usage

External("Grab-SetSettings", gGrabberSettings)

Grab-StartPreview

Syntax Set Field [result, External("Grab-StartPreview", "left|top|width|height|switches")]

This starts the video preview stream at the specified coordinates and dimensions.

Parameters

left + top *the coordinates (in pixels) on the window of the upper left part of the video rectangle.*
width + height *give the dimension (in pixels) of the video rectangle.*
switches *(optional) these determine how this function works*

switches can be left empty or can be this:
-EnableHighQuality *this will enable a high quality preview and recording on some video sources.*

Returned result

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

Special considerations

Important: Use the function "Grab-Initialise" first to test if there if an input source available and to initialise it.
The results of "Grab-Initialise" are the maximum width and height. To get a proportional smaller image divide the width and height by the same factor.

Example usage

Set Field [result, External("Grab-StartPreview", "20|120|80|60")]

This will start a preview at position left 20 and top 120, and 80 pixels wide and 60 high.

Example 2

We want to start to display a preview at half the size of the video source. We assume that in your FileMaker file the following fields are defined:

gErrorCode	Global, text	gWidth	Global, number
gWidthHeight	Global, text	gHeight	Global, number

In ScriptMaker add the following script steps:

```
Set Field [ gWidthHeight, External("Grab-Initialise", "")]
If [Left(_WidthHeight , 2)  =  "$$"]
  # An Error occurred. Save the errorcode into the global "gErrorcode".
  Set Field [ gErrorcode, _WidthHeight]
Else
  # No Error; get the width and height:
  Set Field [ gWidth,  Trim(Left(  _WidthHeight , Position( _WidthHeight , "|" , 1 , 1 ) - 1))]
  Set Field [ gHeight, Trim(Middle(_WidthHeight , Position( _WidthHeight , "|" , 1 , 1 ) + 1 , 64000))]
  # Start the preview at half of the size of the maximum.
  Set Field [ gErrorCode, External("Grab-StartPreview", "10|100|" & _Width /2  & "|" & _Height / 2) ]
End if
```

Grab-Stop

Syntax Set Field [result, External("Grab-Stop", "")]

This stops the video preview stream.

Parameters

no parameters *leave empty for future use.*

Returned result

This function always returns 0.

Example usage

Set Field [result, External("Grab-Stop", "")]

This will stop showing the preview.

Grab-Version

Syntax Set Field [result, External("Grab-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

switches *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 "Grabber Plug-in 1.2"

VersionNumber:
If you asked for the version number it returns the version number of the plug-in x1000. For example version 1.2 will return number 1200.

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

Set Field [result, External(Grab-Version, "")]

This will return the current version of the plug-in for example: "Grabber Plug-in 1.4".

Example 2

External("Grab-Version", "-GetVersionNumber") will return 1100 for version 1.1.
External("Grab-Version", "-GetVersionNumber") will return 1101 for version 1.1b1
External("Grab-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.