

InsideScan[™] 3

Reference Manual

**Mac OS X and Microsoft Windows plug-in v3.5
for FileMaker Pro 14 or newer**

© 2008-2019 PowerSolutions S.r.l.

NOTE: documentation for new or modified features in version 3 has been written in red

InsideScan™ 3

Copyright © 2008-2019 PowerSolutions S.r.l.
All rights reserved

SOFTWARE LICENSE AGREEMENT

PowerSolutions S.r.l. grants you a non-transferable, non-exclusive license to use this copy of the program and accompanying materials according to the following terms:

You may:

- a) use the program on only one computer at a time;
- b) make one (1) copy of the program in machine readable form solely for backup purposes;
- c) physically transfer the program from one computer to another, provided that the program is used on only one computer at a time.

You may not:

- a) use the program in a network unless you pay for a separate license for each terminal or workstation from which the program will be accessed;
- b) modify, translate, reverse engineer, decompile, disassemble, or copy (except for the backup copy) the program or accompanying materials.

This license is not a sale. Title and copyrights to the program, accompanying materials and any copy made by you remain with PowerSolutions S.r.l.

LIMITED WARRANTY AND DISCLAIMER

THE PROGRAM AND ACCOMPANYING MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

PowerSolutions S.r.l. does not warrant that the functions contained in the program will meet your requirements or that the operation will be uninterrupted or error free.

The entire risk as to the use, quality, and performance of the program is with you. Should the program prove defective, you, and not the Author, assume the entire cost of any necessary repair.

LIMITATION OF LIABILITY

IN NO EVENT WILL POWERSOLUTIONS S.R.L. BE LIABLE FOR ANY DAMAGES, INCLUDING LOSS OF DATA, LOST PROFITS, COST OF COVER OR OTHER SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES ARISING FROM THE USE OF THE PROGRAM OR ACCOMPANYING MATERIALS, HOWEVER CAUSED, EVEN IF THE AUTHOR IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY YOU OR ANY THIRD PARTY.

Introduction

Welcome and thanks for your interest in InsideScan.

InsideScan, for both Mac OS X and Microsoft Windows, is a file and image/document management plug-in that lets you handle your images in a FileMaker Pro database.

Through a set of external functions, InsideScan expands FileMaker Pro functional aspects to a considerable extent by letting you:

- Scan images and documents directly from TWAIN and Apple Image Capture (Mac OS X only) compliant acquisition sources, and choose to save them on disk or into your container fields. It is also possible to scan and save multi images and documents to TIFF and PDF files.
- Import images in your database, by supplying their pathname or URL. Batch images import capability is available too, as well as importing by reference.
- Export images from container fields, in a specified format (i.e. JPEG, TIFF, PICT, BMP, PSD, single frames, etc.).
- Create a compressed thumbnail of the image to save memory space or to make presentations.
- Apply a set of transformations to an image, such as resize, depth, resolution, rotate, invert, crop, convert, flip, scale and more.
- Apply a watermark to images, with opacity, scaling and rotation control.
- Get and set information for images. For example, it is possible to obtain type, width, height, depth, resolution and size. It is also possible to retrieve meta data compliant information, such as IPTC and EXIF.
- Print images with just one command.
- Handle files and folders. For instance you can delete, rename and copy files, create and list folders, get and set dates, OS X permissions and more.

All of this makes InsideScan use ideal for creating databases that electronically manage documentation, for filing images, for creating sales catalogs (even for the Web), for enriching personal data cards and for all those situations where using images is a necessity.

Registration, technical support and contacts

To register your copy of InsideScan and take advantage of our technical support services, visit the PowerSolutions web site at <https://www.powersolutions.it/>

If you would like to get in touch with us personally, please write to us at one of the following electronic mail addresses:

support@powersolutions.it (technical support)

info@powersolutions.it (general queries)

Information about this guide

You will find all of the information you need to correctly install and use InsideScan in this guide. Information in this guide, however, is given assuming that you are familiar with the general concepts of the supported operating systems and FileMaker Pro (especially in regards to external functions use).

Hardware and software requirements

InsideScan 3.x is a 64-bit only plug-in and requires FileMaker Pro version 14 or newer in order to run. The minimum operating systems are Mac OS X 10.10 and Windows 7 SP 1.

Installation

Before you start installing InsideScan, be sure you have obtained its registration code.

If you don't have the code, InsideScan will operate just the same, but only in demo mode (it will run for a half hour period, without any limitation; to test the plug-in for a further half hour period, just quit FileMaker Pro and launch it again).

To install InsideScan on your hard drive, just copy the plug-in into the FileMaker "Extensions" folder (the procedure is the same for both Mac OS X and Windows).

Starting InsideScan

After having installed InsideScan, you can run FileMaker Pro and open the test database included with the package to take a look at what the plug-in can do for you.

Please also use the section below for function reference.

Enjoy!

InS_Acquire

[modified in version 3.0]

Scans one or more images from the currently selected acquisition source.

Syntax:

InS_Acquire([format] [; path] [; show UI] [; count] [; settings] [**; interface**])

Parameters:

Format [optional]

String specifying the graphic format to be used for the scanned images. If the parameter is missing, InsideScan will use JPEG as default, unless the file path's extension specifies a different format.

Default value: *JPEG*

Path [optional]

String specifying the full path of the file where the scanned image will be saved. If destination format has been specified, the file extension (if any) must match that format, otherwise an error will be returned.

Default value: *none*

Show UI [optional]

If 0, the plugin will scan directly from the device without showing the driver's interface; otherwise, the driver's interface will be shown as usual (default behavior). Notice that this feature is not mandatory: TWAIN devices doesn't need to support this feature, so it isn't guaranteed to work as expected.

Default value: *1*

Count [optional]

Maximum number of images to be scanned, -1 if you want to acquire all of the available sheets.

Default value: *1*

Settings [optional]

List of pairs *key=value*, separated by space, to allow the user specify additional settings related to either the scanning or the destination image. Currently supported settings are:

jpegQuality=N quality level to be used for the JPEG format encoding. *N* is an integer in range [1..100].

startIndex=N first valid index to be used in multiple images acquisition. *N* is an integer greater than zero.

<i>indexDigits=N</i>	number of digits for the index to be appended to the image file name in multiple images acquisition. <i>N is an integer greater than zero.</i>
<i>allowMultiPage=M</i>	this parameter lets you save multi-page scans to separate files even if you're using a multi-page capable file format like TIFF or PDF. <i>M can be either "yes" or "no".</i>
<i>inverse=M</i>	this parameter lets you fix a wrong digitalization behavior that causes images being returned with wrong colors. <i>M can be either "yes" or "no": default value is "no", so if you're getting wrong images back you'll have to specify 'inverse=yes'.</i>
<i>duplex=M</i>	this parameter lets you enable duplex scanning, if supported by your scanner. <i>M can be either "yes" or "no": default value is "no", i.e. duplex scanning is disabled by default.</i>

Default value: *none*

Interface [optional]

Software interface to be used to acquire the image from the hardware source. There are currently two possible values:

<i>TWAIN</i>	image will be acquired by using previously selected TWAIN source. If none, the source selection dialog will be shown.
<i>IC</i>	image will be acquired by using previously selected scanner or camera from Apple Image Capture technology. If none, the camera list will be shown [Mac OS X only]

Default value: *TWAIN*

Returned value

The function returns the full path of the file where the image has been saved, a list of path if more than a file was created, or an image if a file path has not been specified.

Remarks

This function acts differently based on the availability of multiple images and a valid output path:

- A single image is available

If the file path is valid, the image will be saved using this path; otherwise it will be returned to the FileMaker Pro script, to be placed into a container field.

- Multiple images available

If the file path is valid, the images will be saved using this path; if the format requested

is a multi-page or multi-frame format, all of the acquired pictures will be saved within that file. For example, the call:

```
InS_Acquire(""); "/Path/To/File.pdf"; 1; 4)
```

will create a file named "File.pdf", with up to four pages (one for each scanned image) and return the text:

```
/Path/To/File.pdf
```

Otherwise, if the format to be used doesn't support multiple pages or layers, or if you've specified the parameter "allowMultiPage=no", the specified path will be used as a base name for a set of images. For example, the call:

```
InS_Acquire(""); "/Path/To/File.jpg"; 1; 4)
```

will create up to four different files named "File_001.jpg" ... "File_004.jpg" and return the following text:

```
/Path/To/File_001.jpg  
/Path/To/File_002.jpg  
/Path/To/File_003.jpg  
/Path/To/File_004.jpg
```

Similarly, if you call:

```
InS_Acquire(""); "/Path/To/File.pdf"; 1; 3; "allowMultiPage=no")
```

you'll get the following result:

```
/Path/To/File_001.pdf  
/Path/To/File_002.pdf  
/Path/To/File_003.pdf
```

If the file extension is omitted, it will be added based on the file format specified as first call parameter. This means that the call:

```
InS_Acquire("JPEG"; "/Path/To/File"; 1; 4)
```

will create the very same files as the one described above:

```
/Path/To/File_001.jpg  
/Path/To/File_002.jpg  
/Path/To/File_003.jpg  
/Path/To/File_004.jpg
```

Finally, if a full path has not been specified, a multi-page/layer image will be built in memory and returned to the FileMaker Pro script. If the format specified doesn't support multiple pages or layers, an error will be returned.

InS_ChoseFolder

Lets the user choose a folder using the folder selection dialog.

Syntax:

InS_ChoseFolder

Parameters:

None

Returned value

The function returns the full path of the chosen folder, or an error code if it was unable to complete the operation. If the user cancels the selection, an empty string is being returned.

InS_ ChooseFile

Lets the user choose a file using the standard file selection dialog.

Syntax:

InS_ChooseFile([type1 type2 ... type N])

Parameters:

Type list [optional]

A list of the image types to be shown in the dialog; the types are separated by spaces. If the parameter is omitted or empty, only readable image files will be shown.

A special "*****" type can be used to force the dialog show any file type.

Returned value

The full path of the file chosen by the user

InS_CreateFolder

Creates the folder with specified full path.

Syntax:

InS_CreateFolder(path)

Parameters:

Folder path

Full path of the folder to be created.

Returned value

Full path of the folder just created, or an error message if there were problems.

InS_Copy

Copies a file to another location.

Syntax:

InS_Copy(source path; destination path)

Parameters:*Source path*

Full path of the file to be copied

Destination path

Full path of the file to be created

Returned value

Full path of the newly created file, or an error message if there were problems during copy.

InS_Delete

Removes the file specified.

Syntax:

InS_Delete(path)

Parameters:

Path

String specifying the full path of the file (or folder) to be removed.

Returned value

The function returns 0 if the file (or folder) has been successfully deleted, otherwise an error code is returned.

Remarks

-

InS_Effects

Applies the specific effects to the image and returns the resulting picture.

Syntax:

InS_Effects(container, path or URL; effect list)

Parameters:

Container, Path or URL

The container field containing the source picture, or a string specifying the full path or the absolute URL of the source image file.

Effect List

String containing a list of effects to be applied in the same order. Effects are separated each other by a space. Supported effects are:

contrast=Value	Value in range [-100..100]
brightness=Value	Value in range [-100..100]
gamma=Value	Value in open range]0..N]

Please note: duplicate commands won't be ignored. For example, if you specify the following effects:

```
InS_Effects(Table::Image, "contrast=15 brightness=20 contrast=5")
```

InsideScan will perform both contrast operations, so the contrast value 15 will be applied, the brightness set to 20, then again the contrast value 5 will be applied.

Returned value

The function returns the new image obtained by applying supplied effects to a copy of the source image. The original picture won't be modified.

Remarks

-

InS_Exists

Checks if the file specified exists.

Syntax:

InS_Exists(path)

Parameters:

Path

String specifying the full path of the file (or folder) to be verified.

Returned value

The function returns:

- 1 if the file exists
- 2 if the folder exists
- 0 if it doesn't exist
- "ERROR n" if an error occurred while trying to access the file.

Remarks

-

InS_Export

Exports the image from the container field to File Path, converting it to the specified format (if any).

Syntax:

```
InS_Export(container; path [; format])
```

Parameters:

Container

The container field containing the picture to be exported on disk.

Path

String specifying the full path of the file to be created.

Format [optional]

Image format to use when exporting the image.

Returned value

The function returns the full path of the exported image, otherwise an error code if the plug-in has been unable to export it.

Remarks

You can specify the output format by either the file extension in File Path, or the Format parameter. If you specify both of them, and they're incompatible, an error will be returned.

For example, the following will give you an error:

```
InS_Export(db::Image; "/Path/File.jpg"; "PNG")
```

as the *.jpg* extension is not compatible with *PNG* format.

Examples

The following script:

```
Set Field [db::Result; InS_Export(db::Image;  
    "/Volumes/ExtDisk/Imgs/Img1.jpg";  
    "JPEG") ]
```

exports the image from *db::Image* container field to a *JPEG* file in the folder *Imgs* on the volume *ExtDisk*. The text field *db::Result* will contain either the full path of the image created by InsideScan, or an error message in the form "ERROR X", where X is the error code.

InS_GetSource

Returns currently selected source for image acquisition.

Syntax:

InS_GetSource()

Parameters:

-

Returned value

The function returns the name of currently selected source along with related image acquisition interface name. Source name and interface name are separated by a semicolon character.

Remarks

InS_GetSourceList*[new in version 3.0]*

Returns the list of available sources for image acquisition.

Syntax:

InS_GetSourceList([interface])

Parameters:

Interface [optional]

Software interface to be used to list the available hardware sources. There are currently two possible values:

TWAIN will return a list of available TWAIN sources, if any.

IC will return a list of available devices compatible with Apple Image Capture technology, if any [Mac OS X only]

Default value: **TWAIN**

Returned value

The function returns the names of currently available acquisition sources along with related image acquisition interface name.

Returned sources are separated by a *carriage return* (ASCII 13).

Remarks

- You can use one of the returned names to select a source programmatically, by calling InS_SelectSource

InS_GetTwainSource*[no more available in version 3.0]**Returns currently selected TWAIN source.***Syntax:**

InS_GetSource

Parameters:*None***Returned value**

The function returns the name of currently selected TWAIN source

Remarks

- This function is no more available starting from version 3.0. It's been replaced by **InS_GetSource**

InS_GetTwainSourceList*[no more available in version 3.0]*

Returns the list of available TWAIN sources.

Syntax:

InS_GetTwainSourceList

Parameters:

None

Returned value

The function returns the names of currently available TWAIN sources, separated by a *carriage return* (ASCII 13)

Remarks

- You can use one of the returned names to select a source programmatically, by calling InS_SelectTwainSource

- This function is no more available starting from version 3.0. It's been replaced by **InS_GetSourceList**

InS_GetFileInfo

Returns a list of details about the specified file.

Syntax:

InS_GetFileInfo(path; format)

Parameters:

Path

Full path of the file to be examined.

Format [optional]

Text to be used to return custom-formatted info. This text can be anything you like, and will be processed by the external function to replace special parameters with corresponding values. Special parameters are:

<code>%crdate</code>	Replaced with creation date (format: YYYY-MM-DD)
<code>%crttime</code>	Replaced with creation time (format: HH:MM:SS)
<code>%moddate</code>	Replaced with modification date (format: YYYY-MM-DD)
<code>%modtime</code>	Replaced with modification time (format: HH:MM:SS)
<code>%ownerid</code>	Replaced with file owner ID (number, Mac OS X only)
<code>%ownername</code>	Replaced with file owner name (Mac OS X only)
<code>%groupid</code>	Replaced with file owner group ID (number, Mac OS X only)
<code>%groupname</code>	Replaced with file owner group Name (Mac OS X only)
<code>%hfscreator</code>	Replaced with HFS file creator code (four chars, Mac OS X only)
<code>%hfstype</code>	Replaced with HFS file type code (four chars, Mac OS X only)
<code>%permissions</code>	Replaced with Posix file permissions (string, Mac OS X only)
<code>%size</code>	Replaced with file size (in bytes)

This allows you not only to automatically format complex file descriptions, but also request just a specific file information. If this parameter is omitted or empty, the default format will be used.

Returned value

The function returns a list of values separated by CR. The following labels are used:

Size: 	File size in bytes.
Created: <DT>	Date and time of creation, in the following format: YYYY-MM-DD HH-MM-SS
Modified: <DT>	Date and time of last modificaton, in the following format: YYYY-MM-DD HH-MM-SS
Owner: <O>	Owner user name and ID, in the following format: name (ID)
Group: <G>	Owner group name and ID, in the following format: group (ID)
HFS Type: <T>	Four char code identifying the file type (Mac OS X)

HFS Creator: <C> Four char code identifying the application (Mac OS X)
Permissions: <P> Posix file permissions, in the following format:
perm. string (number)
for example:

Remarks

-

Example

The following external function call:

```
InS_GetFileInfo("/Path/To/File.jpg")
```

could return something like the following:

```
Size: 315052  
Created: 2007-07-21 19:06:37  
Modified: 2007-07-21 19:06:38  
Owner: testuser (501)  
Group: testgroup (501)  
HFS Type: 'JPEG'  
HFS Creator: 'prvw'  
Permissions: ---rw-r--r-- (644)
```

Otherwise, if you need a more descriptive description:

```
InS_GetFileInfo("/Path/To/File.jpg";"The file was created on %crdate at  
%crttime, modified on %moddate at %modtime. It is owned by %ownername  
(ID: %ownerid) and group %groupname (ID: %groupid). HFS creator is  
%hfscreator, HFS type is %hfstype. The file size is %size bytes, with  
permissions: %permissions.")
```

will return something like the following:

```
The file was created on 2007-07-23 at 09:56:21, modified on 2007-07-23  
at 09:56:21. It is owned by testuser (ID: 501) and group testgroup  
(ID: 501). HFS creator is 'prvw', HFS type is 'JPEG'. The file size is  
54259 bytes, with permissions: ---rw-r--r--.
```

InS_GetImageInfo

Returns a list of details on the image.

Syntax:

InS_GetImageInfo(container, path or URL; format)

Parameters:

Container, Path or URL

The container field containing the picture to be examined, or a string specifying the full path or the absolute URL of the file to be examined.

Format [optional]

Text to be used to return custom-formatted info. This text can be anything you like, and will be processed by the external function to replace special parameters with corresponding values. Special parameters are:

<code>%width</code>	Replaced with image width (in pixels)
<code>%height</code>	Replaced with image height (in pixels)
<code>%depth</code>	Replaced with image color depth (in bits per pixel)
<code>%resolution</code>	Replaced with image resolution (in dots per inch) NOTE: if horizontal and vertical resolution differ, the parameter will be replaced with the formatted string: “(<HRes>x<VRes>)”
<code>%grayscale</code>	Replaced with either “grayscale” or “color”, depending on the image colors
<code>%image_format</code>	Replaced with image file format (TIFF, JPEG, etc.)
<code>%meta_format</code>	Replaced with metadata model name, if metadata are available within the image (EXIF, IPTC, etc.)
<code>%frame_count</code>	Replaced with the number of frames stored into the image
<code>%image_size</code>	Replaced with the size of the image (in bytes)
<code>%image_path</code>	Replaced with the file path of the image
<code>%is_reference</code>	Replaced with “1” if the image in field is a reference to an image file; replaced with “0” if the image in field is not a reference to a file.

This allows you not only to automatically format complex image descriptions, but also request just a specific image information. If this parameter is omitted or empty, the default format will be used.

Returned value

The function returns a string containing details on image depth, resolution, size, format and (if available) metadata information format. Each information is on a separate line, and the lines are separated by CR (ASCII 13)

Remarks

When using the default information format, missing information are omitted: for example, metadata information are missing if the file doesn't contain metadata.

Example

The following external function call:

```
InS_GetImageInfo("/Path/To/File.jpg")
```

will return default-formatted info, like the following:

```
Width:1600 pixel
Height:1200 pixel
Depth:24 bit/pixel
Resolution:72 dpi
Format:JPEG
Size:445931 bytes
MetaData:COMMENTS,EXIF,EXIF_MAIN
Path:/Path/To/File.jpg
```

Otherwise, if you need a more descriptive description:

```
InS_GetImageInfo("/Path/To/File.jpg";"The image '%image_path' is
%width pixels wide and %height pixels tall. Its depth is %depth bits/
pixel, with a resolution of %resolution dpi (%grayscale)")
```

will return something like the following:

```
The image '/Path/To/File.jpg' is 1100 pixels wide and 1700 pixels
tall. Its depth is 24 bits/pixel, with a resolution of 200 dpi (color)
```


InS_GetMetadata

Extracts the specified metadata format from the file.

Syntax:

InS_GetMetadata(container, path or URL; format)

Parameters:

Container, Path or URL

The container field containing the picture to be examined, or a string specifying the full path or an absolute URL of the file to be examined.

Metadata format

String specifying the format of the metadata to be retrieved. Recognized formats are:

COMMENTS	Comments or keywords about the file
EXIF_MAIN	Metadata common to TIFF and Exif files
EXIF	Exif specific metadata
EXIF_GPS	Exif GPS metadata that are part of the Exif standard
EXIF_MAKERNOTE	Exif maker notes are metadata that are added by camera constructors. The following makers are currently supported: Asahi, Canon, Casio (type 1 and type 2), Fujifilm, Kyocera, Minolta, Nikon (type 1, type 2 and type 3), Olympus / Epson / Agfa, Panasonic, Pentax and Sony
EXIF_INTEROP	Exif interoperability metadata
IPTC	
XMP	

Returned value

The function returns the full path of the exported image, otherwise an error code if the plug-in has been unable to export it.

Remarks

-

InS_Import

Imports the image specified, by content or by reference.

Syntax:

InS_Import(path or URL [; as reference])

Parameters:

Path or URL

String specifying the full path or the absolute URL of the file to be imported.

As Reference [optional]

Integer number treated as a boolean flag: 0 is false, 1 is true. If the parameter is omitted or empty, the image is imported by content.

Returned value

The function returns the imported image, to be stored into a container field.

Remarks

-

InS_ImportFrame

Imports a layer, frame or page from the specified image file.

Syntax:

InS_ImportFrame(path or URL; frame number)

Parameters:

Path or URL

String specifying the full path or the absolute URL of the source file.

Frame Number

Number (zero-based) of the layer, frame or page to be imported from the file.

Returned value

The function returns the extracted image, to be stored into a container field.

Remarks

Currently supported multi-page/frame/layer file formats are:

- TIFF
- PSD (Photoshop)
- GIF
- ICO (Windows icon file)

InS_Launch

Opens the file specified with an external application.

Syntax:

InS_Launch(path, [application path])

Parameters:

Path

String specifying the full path of the file to be opened.

Application Path [optional]

String specifying the full path of the application to be used to open the file. If the parameter is omitted or empty, the file is opened using the default application for the specific file type.

Returned value

The function returns 0 if the file was open with no problems, otherwise an error code is returned.

Remarks

-

InS_ListFolder

Gets a list of the contents of the specified folder.

Syntax:

InS_ListFolder(path [; depth] [; list hidden] [; list files] [; list folders] [; filter extensions])

Parameters:

Path

String specifying the full path of the folder whose contents are to be listed.

Depth [optional]

Number specifying the "depth" levels to be listed, if the folder contains subfolders. Use 0 to list just first level, -1 to list all of the contents up to any depth, or a number greater than zero to set the maximum depth the function should reach. If the parameter is omitted or empty, only elements at first level are listed.

List Hidden [optional]

Number specifying if the hidden files should be listed. If 1, the resulting list will include hidden files; if 0 (or a number different than 1) the resulting list won't include hidden files. If the parameter is omitted, hidden files won't be listed.

List Files [optional]

Number specifying if the regular files should be listed. If 1, the resulting list will include files; if 0 (or a number different than 1) the resulting list won't include files. If the parameter is omitted, files will be listed.

List Folders [optional]

Number specifying if the folders should be listed. If 1, the resulting list will include folders; if 0 (or a number different than 1) the resulting list won't include folders. If the parameter is omitted, folders will be listed.

Filter Extensions [optional]

String containing a list of file extensions to be filtered, separated by spaces. If the parameter is omitted, any file will be listed.

Returned value

The function returns a text with a list of file and/or folder paths relative to the Folder Path; these paths are separated by a carriage return. An empty string is returned if the listing fails.

Remarks

-

InS_Move

Moves a file or folder to a new location, and/or renames it.

Syntax:

InS_Move(source path; destination path)

Parameters:

Source Path

String specifying the full path of the file or folder to be moved.

Destination Path

String specifying the full path of the newly created file or folder.

Returned value

The function returns 0 if the move was successful, an error code otherwise.

Remarks

The destination file or folder name should be specified explicitly: this function doesn't adopt original file name as default.

InS_Preferences

Shows the InsideScan Preferences window.

Syntax:

InS_Preferences

Parameters:

None

Returned value

None

Remarks

The Preferences window is shown to let the user modify InsideScan settings.

InS_Print

Prints the file or field using specified printer, or the default one if a printer name has not been specified.

Syntax:

InS_Print(container, path or URL [; show page setup?] [; show printing dialog?] [; printer name])

Parameters:

Container, Path or URL

The container field containing the picture to be printed, or a string specifying the full path or the absolute URL of the file to be printed.

Show Page Setup [optional]

Specifies if the page setup dialog should be used. If the parameter is omitted or empty, the dialog is not used.

Show Printing Dialog [optional]

Specifies if the printing dialog should be used. If the parameter is omitted or empty, the dialog is not used.

Printer Name [optional]

Name of the printer to be used. If the parameter is omitted or empty, the default printer is used.

Returned value

The function returns an error code, or 0 if the image was successfully printed.

Remarks

-

InS_Register

Function to be called to use the plug-in in registered mode.

Syntax:

InS_Register(user name; license key)

Parameters:

User Name

String specifying the name of the licensee.

License Key

String specifying the license key received after purchasing InsideScan

Returned value

The function returns 1 if the plug-in has been registered correctly, 0 otherwise.

Remarks

-

InS_SelectSource

[new in version 3.0]

Selects a source to be used as default in following scans.

Syntax:

InS_SelectSource([source][, interface])

Parameters:

Source [optional]

String specifying the name of the source to be selected, as returned by InS_GetSource or InS_GetSourceList: *source_name;interface_name*

If the source name is omitted, the hardware source selection window will be displayed for the selected interface, to let the user choose the source. If interface name is omitted, too, a list of TWAIN sources will be shown.

Returned value

The function returns the name of the selected source, or an error code if an error occurred.

Remarks

This function replaces `InS_SelectTwainSource`, available in InsideScan up to version 2.6

Example

1. The following external calls:

```
InS_SelectSource("")  
InS_SelectSource(";TWAIN")
```

will both display the TWAIN source list to let the user choose the default one to be used when scanning through TWAIN technology.

2. The following external call:

```
InS_SelectSource(";IC")
```

will display a list of the cameras available to Image Capture technology (Mac OS X only).

3. The following external call:

```
InS_SelectSource("CanoScan LIDE 200;IC")
```

will select the source **CanoScan LIDE 200** to be used as default when scanings through Image Capture technology (Mac OS X only).

InS_SelectTwainSource*[no more available in version 3.0]*

Selects a TWAIN source to be used as default in following scans.

Syntax:

InS_SelectTwainSource(source)

Parameters:

Source [optional]

String specifying the exact name of the source to be selected. If the parameter is omitted, the standard TWAIN source selection window will be displayed, to let the user choose the source.

Returned value

The function returns the name of the selected source, or an error code if an error occurred.

Remarks

- This function is no more available starting from version 3.0. It's been replaced by **InS_SelectSource**

InS_SetFileInfo

Sets the file info based on parameter values.

Syntax:

InS_SetFileInfo(path; info)

Parameters:

Path

String specifying the full path of the file to be changed.

Info

List of couples "PARAMETER=VALUE", where PARAMETER is one of the following:

crdate=<D>	Creation date, with <D> formatted YYYY-MM-DD
crttime=<T>	Creation time, with <T> formatted HH:MM:SS
moddate=<D>	Modification date, with <D> formatted YYYY-MM-DD
modtime=<T>	Modification time, with <T> formatted HH:MM:SS
ownerid=<UID>	File owner ID, where <UID> is an integer number (existing user ID)
ownername=<O>	File owner name
groupid=<GID>	File owner group ID, where <GID> is an integer number (existing group ID)
groupname=<N>	File owner group name
hfscreator=<C>	HFS file creator code (Mac OS X), where <C> is a four character sequence
hfstype=<T>	HFS file type code (Mac OS X), where <T> is a four character sequence
permissions=<S>	Posix file permissions (Mac OS X), where <S> is a character representation of the Posix permissions

Returned value

The function returns the full path of the file, or an error code if an error occurred.

Remarks

-

InS_Transform

Applies the specific transformations to the image and returns the resulting picture.

Syntax:

InS_Transform(container, path or URL; transformation list[; frame number])

Parameters:

Container, Path or URL

The container field containing the source picture, or a string specifying the full path or the absolute URL of the source image file.

Transformation List

String containing a list of transformations to be applied in the same order. Transforms are separated each other by a space. Supported transforms are:

resize=WxH	Exact resize, non-proportional
depth=BitPerPixel	BitPerPixel in [1,4,8,16,24,32]
resolution=DPI	DPI is an integer value (number of dots per inch)
grayscale	Final result will be a grayscale image
rotate=Angle	Angle = degrees. Positive values rotate image counterclockwise, negative values rotate it clockwise
scale=Percent%	
scale=WxH	Fit to box size having width = W and height = H
flip=[H V]	Flip the image horizontally (H) or vertically (V)
invert	Invert image colors
crop=N	Number of predefined crop setting, stored in preferences
crop=WxH	Crop centered rect having width = W and height = H
crop=X,Y,X2,Y2	Crop exact rect having left,top,right,bottom as specified
crop=X,Y,WxH	Crop exact rect having left,top,width,height as specified
format=F	Resulting image file format: F = "TIFF" (or "PNG", "JPEG", "GIF", etc).
quality=N	Resulting image quality (for JPEG images only): N is a number in range [1..100]

Please note: duplicate commands won't be ignored. For example, if you specify the following transforms:

```
InS_Transform(Table::Image, "rotate=50 grayscale rotate=15")
```

InsideScan will perform both rotate operations, so the image will be rotated by 50 degrees, converted to grayscale, then rotated again by 15 degrees.

Frame Number [optional]

Number (one-based) of the layer, frame or page the transformation should be applied on. If the parameter is omitted, the first frame is used.

Returned value

The function returns the new image obtained by applying supplied transformations to a copy of the source image. The original picture won't be modified.

Remarks

-

Example

1. The following external call:

```
InS_Transform(Table::Image, "crop=100x100 rotate=15 grayscale")
```

will return a portion of the original image (centered rect sized 100x100 pixel), rotated by 15 degrees and converted to grayscale.

2. The following external call:

```
InS_Transform(Table::Image, "format=JPEG quality=50")
```

will return a JPEG picture with image quality set to 50%.

InS_TransformJPEG

Applies the specific lossless transformations to the JPEG image file and returns the file path.

Syntax:

InS_TransformJPEG(path; transformations[; perfect])

Parameters:

Path

A string specifying the full path of the JPEG file to be transformed.

Transformation List

String containing a list of transformations to be applied in the same order. Transforms are separated each other by a space. Supported transforms are:

rotate=Angle	Angle = degrees. Positive values rotate image counterclockwise, negative values rotate it clockwise. Allowed values are: 90, 180, 270
flip=[H V]	Flip the image horizontally (H) or vertically (V)

Please note: duplicate commands won't be ignored. For example, if you specify the following transforms:

```
InS_TransformJPEG(Table::Image, "rotate=90 flip=H rotate=180")
```

InsideScan will perform bot rotate operations, so the JPEG image will be rotated by 90 degrees, flipped horizontally, then rotated again by 180 degrees.

Perfect [optional]

Number acting as a boolean value, indicates if the image size can be altered in order to allow the requested transformation (value = 1) or if the image size should be preserved regardless of the function success (value = 0). If the parameter is omitted, the image size is preserved, and the function may fail to be applied.

The reason of the above is that the algorithm used to apply the transforms needs that the image horizontal and vertical sizes are multiple to a specific value, usually 8 or 16. If you specify 0 to ensure the transformation will be applied, the image **could** be trimmed down to conform to the above prerequisite, if needed.

Returned value

The function returns the path of the source JPEG image, modified by applying requested lossless transformations. The source picture **will be modified**.

Remarks

-

Example

The following external call:

```
InS_TransformJPEG(Table::ImagePath, "rotate=180")
```

will losslessly rotate the original image by 180 degrees.

InS_TranslatePath

Translates the absolute path from system notation to FileMaker notation and back.

Syntax:

InS_TranslatePath(path; notation)

Parameters:

Path

String specifying the absolute path to be translated.

Notation

String specifying the notation which the absolute path should be translated to:

FMP	Translate the path to FileMaker Pro notation
OS	Translate the path to the current Operating System notation

Returned value

The function returns the translated path, or the string "ERROR N" where N is an error code.

Remarks

-

Example

On Mac OS X the following external call:

```
InS_TranslatePath("/Users/username", "FMP")
```

will return the string:

```
"/Macintosh HD/Users/username"
```

where "Macintosh HD" is the actual name of the startup disk. Likewise, the call:

```
InS_TranslatePath("/Macintosh HD/Users/username", "OS")
```

will return the string:

```
"/Users/username"
```

InS_Version

Returns the version accordingly to the request.

Syntax:

InS_Version([format])

Parameters:

Format [optional]

String specifying the type of information you need. Accepted parameters are "short", "long" and "platform"; if the parameter is omitted, "short" version is returned.

Returned value

The function returns a string with the requested information.

Remarks

-

InS_Watermark

Overlays the second image on the first one with given settings.

Syntax:

InS_Watermark(source container, path or URL; watermark container, path or URL [; settings])

Parameters:

Source container, Path or URL

The container field containing the source picture, or a string specifying the full path or the absolute URL of the source image file.

Watermark container, Path or URL

The container field containing the watermark picture, or a string specifying the full path or the absolute URL of the watermark image file.

Settings [optional]

A list of settings in the form "*keyword=value*" to specify how the watermark should be handled. Supported settings are:

opacity=Value	Opacity of the watermark image. Value in range [0..100] where 1 means fully transparent and 100 means fully opaque.
scale=[X Y XY]	Scale [proportional] watermark image to fit within the destination image box: X=horizontally, Y=vertically, XY=both sizes.
resize	Resize [not proportional] watermark image to fit within the destination image box.
format=F	Resulting image file format: F = "TIFF", "PNG", "JPEG", "GIF", etc.
center	Watermark image is centered on source image.
rotate=D	Watermark image is rotated by D degrees before applying transformations.

Note: "scale" settings have greater priority than "resize", so if you specify both parameters "resize" will be ignored.

Returned value

The function returns the new image obtained by overlaying the watermark image to a copy of the source image. The original picture won't be modified.

Remarks

-

Appendix I

InsideScan Error Codes

Error Name	Code	Description
err_DemoTimeout	-100	The demo timeout limit has been reached: quit FileMaker and reopen it.
err_UserCanceled	-1	The operation was canceled by the user
err_None	0	The operation was successfully performed
err_Generic	1	An unknown error occurred
err_InitPreferences	2	An error occurred while loading plug-in preferences: the preferences file could be corrupted
err_ImportOldPreferences	3	An error occurred while importing old version (1.8.x) preferences
err_NotEnoughMemory	4	There isn't enough memory available to complete requested operation
err_UnimplementedFunction	5	Function is not implemented yet
err_BadParameters	100	Parameters error: the external function call contains too many, too few or wrong parameters.
err_BadParameter_First	101	First parameter of the external function call has a wrong/unacceptable value. The same meaning applies to codes in range [101..199].
err_FileNotFound	200	The file was not found at given path.
err_AppNotFound	201	The application was not found at given path.
err_FileExists	202	A file already exists at given path.
err_DirectoryExists	203	A directory already exists at given path.
err_BadFilePath	204	The file path is wrong: either a folder, a disk or a mount point is not found.
err_DirectoryNotFound	205	Given directory doesn't exist
err_CantLoadImage	300	The image specified can't be loaded in memory: it could be due to a memory related problem, or corrupted data.
err_CantUnderstandImage	301	InsideScan isn't able to interpret the image data.
err_CantResizeImage	302	The image specified can't be resized, most likely due to some memory error.
err_CantImportFormat	303	The given format isn't readable by InsideScan.
err_CantExportFormat	304	The given format isn't writable by InsideScan.
err_CantExportImageToFormat	305	The given format doesn't support image in this format. For example, the JPEG format doesn't handle 1-bit images.
err_UnknownFormat	306	The given format is unknown.
err_ImageConversion	307	The image can't be converted as requested.
err_UnknownTransform	308	The given transformation doesn't exist.

err_Conversion	309	
err_BadResizeValues	310	The external function call contains invalid resize values, for example an empty width and/or height.
err_BadFormat	311	
err_BadScaleValues	312	The external function call contains invalid scale values.
err_BadCropValue	313	The external function call contains invalid crop values.
err_BadImageData	314	The image contains invalid data
err_TwainNotAvailable	400	TWAIN is not installed on the computer.
err_CantSelectSource	401	An error occurred while trying to select the TWAIN source specified.
err_CantOpenSource	402	An error occurred while trying to open the TWAIN source specified.
err_CantAcquireImage	403	An error occurred while trying to acquire an image from the TWAIN source specified.
err_CantControlSourceUI	404	The TWAIN source specified doesn't allow to scan without showing its User Interface (InS_Acquire external call with "Show UI" parameter set to 0).
err_MultiImageNoPath	405	You asked InsideScan to perform a multi-page acquisition to field (no file path specified) using a "single-page format" like JPEG, PNG and so on.
err_DuplexNotSupported	406	Current TWAIN source doesn't support duplex scanning
err_ICDeviceNotAvailable	407	No device available to scan through Apple Image Capture (Mac OS X only)
err_ICDeviceBusy	408	Image Capture device is currently in use and can't be used for scanning (Mac OS X only)
err_ICDeviceCantInit	409	Image Capture device returned an error while trying to initialize scan (Mac OS X only)
err_QuickTimeNotAvailable	500	QuickTime is missing and was needed to perform requested operation
err_Internal_First	1000	Internal error codes start from 1000

Appendix II

Supported graphics formats (builtin)

Format Description	Name	Notes
Windows or OS/2 Bitmap	BMP	
Dr. Halo	CUT	Read only - Grayscale
DirectX Surface	DDS	Read only
Raw fax format CCITT G.3	G3	Read only
GIF	GIF	
High Dynamic Range Image	HDR	
Windows Icon	ICO	
IFF Interleaved Bitmap	IFF	
JPEG Network Graphics	JNG	Read only
JPEG/JIF	JPEG	
KOALA	KOALA	Read only
IFF Interleaved Bitmap	LBM	Read only
Multiple Network Graphics	MNG	Read only
Portable Network Media	PBM	
Kodak PhotoCD	PCD	Read only
Zsoft Paintbrush	PCX	Read only
Portable Network Media	PGM	
Portable Network Graphics	PNG	
Portable Network Media	PNM	
Portable Network Media	PPM	
Photoshop	PSD	Read only
Sun Raster Image	RAS	Read only
SGI Image Format	SGI	Read only
Truevision Targa	TARGA	
Tagged Image File Format	TIFF	
Wireless Bitmap	WBMP	
X11 Bitmap Format	XBM	Read only
X11 Pixmap Format	XPM	

Appendix III

Multimedia formats supported by latest QuickTime version

Format Description	Notes
3DMF	Mac OS 9 & Windows
3GPP	
3GPP2	
AIFF	
AMC	
AMR	
Animated GIF	
AU	
Audio CD Data	Mac OS 9
AVI	
BMP	
CAF	Mac OS X
Cubic VR	
DLS	
DV	
FLC	
GIF	
GSM	
JPEG/JFIF	
Karaoke	
MacPaint	
MIDI	
MPEG-1	
MP3 (MPEG-1, Layer 3)	
M3U (MP3 Playlist files)	
MPEG-2	Playback of program and elementary streams available via QuickTime MPEG-2 Playbck Component, sold separately in the Apple Store online.
MPEG-4	
MOV	
M4A	
M4B	
M4P (iTunes 4 audio)	