1. General

2. Understanding FileTrain

3. Configuration

4. Workflow examples and tips

5. Macros

6. Macro tags for FileTrain

1. General

Laidback Solutions recommend the user of FileTrain to read this guide thouroughly in purpose to fully understand the complex features and possibilities of FileTrain.

Images shown in this manual are from the Mac OSX version of FileTrain. All images may be clicked to be viewed in a bigger size.

For more information, updated FAQ, version updates, plug-ins etcetera, please visit www.laidbacksolutions.se

Mulltiple instances of FileTrain

If more than one FileTrain is running with a complete or partly equal setup there is inevitable that the two instances will be in conflict with each other and unexpexted behaviour willl be the result.

Make sure that only ONE instance of FileTrain is running!
If you are running FileTrain as a Windows service and starts the regular application, make sure to stop all the stations to prevent such unexpected behaviour.

1.1 What is FileTrain

FileTrain is a powerful tool created to simplify and automate static workflow handling of digital material. FileTrain comes with a set of advanced routing options and the default features will cover most workflows. Through plugins it is possible to extend the functionality for very advanced customization such as complex database conections and third party system integrations.

The default FileTrain can read both local and mounted network drives as well as FTP folders and email servers. The filtering of the existing objects on these locations enables more sophisticated handling depending on for example file name or content such as IPTC or XMP content. The detected files and folders may be handled in many different ways including moving, copying, sending email and put to external FTP servers.

FileTrain has many dynamic features such as changing file names dynamically (e.g. handling advanced math or date conversions to produce a new file or folder name). The advanced features also include conditions and file content parsing.

FileTrain is able to handle connections to FTP servers with regular FTP, SFTP and FTPS.

1.2 Requirements

FileTrain runs on Macintosh OSX and Windows. A java runtime, minimum version 1.5, is required for FileTrain to run. For windows platform there is a special version of FileTrain with a built-in Java runtime which will work only with FileTrain. This special version does not require any pre-installed java runtime.

Windows

Most Windows systems have no default Java runtime installed. Make sure that;
(i) a Java runtime installed and
(ii) the version of the Java runtime is at least 1.5.

To get help about your Java version, download your free Java and more, please visit http://www.java.com

Macintosh

A Java runtime is installed in the operating system on Macintosh OSX.

To get help about your Java version, upgrade your free Java and more, please visit

http://www.java.com

1.3 Plugins

The default set of features in FileTrain will cover most regular and advanced workflows. To enable special features such as searching databases etc. it is possible to implement plugins.

There are three different kinds of plug-ins in FileTrain, sources, filters and actions.

Sources are the modules which handle the searching of files or other objects. The default set of sources includes;

1. Files and folders located on a local or remote network drive
2. Files located on a FTP server
3. Emails.
4. Timers (with no need for actual objects to invoke handling in FileTrain)

Other sources that can be used via extended plug-ins may be databases or advanced folder handling where a file is dependent on another.

The **filters** will select which objects to handle. The default set of filters will cover most situations however it is possible to implement special filters.

The **action** sets are the actual handling of an object. In a basic workflow this is merely moving or copying a file or folder but the default set of actions in FileTrain also includes handling such as sending emails, uploading the files to various FTP sites and also expanding files. Special handling may involve IPTC-routing etc.

The sources and actions may fail in different ways while they are working and for this reason a special kind of action was introduced in FileTrain version 5. These error handling actions will be invoked when an error occur and this action may involve sending an email to an administrator or move the file that caused the problem to a certain error folder for example.

Even though FileTrain is shipped with a large set of components enabling very advance workflows the extended feature to create plugins separately makes FileTrain a complete tool for any possible digital workflow.

Failure loading plugin

If a plugin is failing to load at startup this plugin will be removed and not loaded the next time FileTrain starts.

1.4 Windows service

A service is a non-graphical process that will run in the background without any graphical user interface. Before running FileTrain as a service it is needed to do a setup of FileTrain. This is done by opening the regular FileTrain application. After the setup is done the service may be installed (use the File menu in FileTrain application) and started. The installed service will be in automatic mode which means it will automatically be started when the computer is restarted (this may be changed by the user in the systems service control dialog). However the service does not automatically start when it is installed, to start the service you finish the setup and close the normal FileTrain, then start the service in the system service control dialog.

FileTrain running both as service and desktop application
If FileTrain is running as a service AND as a regular desktop application the two instances might be in conflict with each other and unexpexted behaviour may occur in FileTrain.
Make sure that only ONE instance of FileTrain is running!
If you are running FileTrain as a Windows service and starts the regular application, make sure

to stop all the stations to prevent such unexpected behaviour.

1.5 Files

The log files contains information about the FileTrain processes. These log files are vital in case you have a problem with your FileTrain, you may send a support case directly from FileTrain which automatically will include all necessary log files in the support case.

The preferences contains the current settings for FileTrain.

The cache files are used internally in FileTrain to remember it's state when restarted.

The files are default kept in the following locations. Note that you may change the folder for log and cache via the preferences dialog in FileTrain.

Log: *user home*/Library/Logs/FileTrain

Pref: *user home*/Library/Preferences/FileTrain/FileTrainSetup.xml
user home/Library/Preferences/FileTrain/FileTrainCore.xml (this file will be written if there is not enough rights to use the system local preferences node)

Cache: *user home*/Library/Caches/FileTrain

Log: *user home*/AppData/Local/FileTrain/Logs

Pref: *user home*/AppData/Local/FileTrain/Preferences/FileTrainSetup.xml
user home/AppData/Local/FileTrain/*Preferences*/FileTrainCore.xml (this file will be written if there is not enough rights to use the system local preferences node)

Cache: *user home*/AppData/Local/FileTrain/*Caches*

Backup files

Auto backup

Every 12th hour a safety backup of the preference file is copied to the folder **-autoBackup** found in the general preference folder. These safety backups are saved 30 days before automatically deleted by FileTrain.

Version backup

If a new version of FileTrain is installed and started the old preferences will be saved as a safety backup in case you want to start the old FileTrain later. These version backup files are found in the folder **-versionBackup** found in the preferences folder.

1.6 Help/Support

From within FileTrain you may send a support case directly to Laidback. FileTrain will automatically collect all relevant log files and setup information about your FileTrain and send these as a new support arrend.

To be able to get as much information as possible in a support case it is recommended to do the following in case of an error in Filetrain;

1. Turn on ALL as log level (done in preferences dialog)
2. Run FileTrain so the error occurs
3. Go to the support tab in FileTrain main window and create a new support case (please give as much detailed information as possible when describing the problem)

Note that it is not recommended to run FileTrain on log level ALL for a longer period of time since this will cause a lot of log files which may effect FileTrain performance in a bad way.

Default the support case is uploading log files to ane xternal FTP and then sends a support email via Laidback SMTP servers. If Filetrain is running from behind a firewall which is not allowing this SMTP to be used you may in the preferences dialog select another SMTP server. The list of SMTP servers is taken from the SMTP configuration within FileTrain.

2. Understanding FileTrain

This section will give you a more in-depth understanding on how FileTrain's individual parts work and how they all, when put together, will work as a powerful tool for your digital workflow.

FileTrain is built with **stations**. A station is a holder for the building components **sources**, **filters** and **actions**. These individual components are put together within the station to work together. The three core components (source, filter and action) are explained in more detail below. The components can be re-used in different stations to make a complex workflow more understandable and simplify the maintainance. Each individual component (source, filter or action) executes it's own task within the framework of a station. Without the station the individual components can not work.

The core components are as stated unaware of other components. This feature gives the user possibillities to build very advanced workflows with little effort but it requires that the user know how the individual components work.

Reusable components - pros and cons

When setting up FileTrain you will use these individual components (sources, filters and actions) many times and it is important for you to really understand the power but also the pitfalls that this model has.

By using these atomic pieces FileTrain will become more dynamic in it's setup and less setup is actually needed when a station is built with single components compared to if a station was one component on its own.

By reusing the single components less setup is needed but there is a huge pitfall that you must be aware of. If you change a single component the changes will have effect on all places where this component exists. This is of course very useful when you like to change a single setting in many places but if you are not cautious this may lead to unwanted behavior in FileTrain.

Example: You create a filter which you name 'Jpeg images' and the filter is setup to look for files with name '*.jpg'. This filter is then used in many various stations in the setup. At a later stage you realize that in one station you are not only interested in jpeg images but indeed images that are larger than 1 MB. The intuition says that you just open that filter and add a new restriction for this size but if you do this you have to remember that all the places where this filter is used will have the same changes.

This singular component feature is a very powerful feature but as seen above, be aware of it's drawbacks when setting up big workflows.

2.1 Sources

A source is a component that will search for objects. The most common type of source is a folder on the local drive or on a mounted network drive. Other sources may for example be a folder on a remote FTP server, an email account on a remote email server or a database.

FileTrain has four different built-in sources (see below).

Other sources such as database lookups, synchronization sources or other special sources may be developed as plugins to FileTrain. For more information about an individual source type, see the configuration section.

Folder source

A folder source will look for objects in a folder on a local or remote network drive. The folder source is able to look down in a sub folder structure to detect objects in a sub level of the root folder.

FTP source

The FTP source will use the FTP protocol to look for objects on a remote FTP server.

Email source

The email source is scanning an email server for emails. Special email filters will apply to such a source.

Timer source

The timer source is a special type of source which is not actually looking for any files, folders or emails. This source will work as a alarm clock. When the alarm goes off this source will notify the station and the station may perform special tasks which do not require any objects to execute action upon. This may for example be used to send a daily report at a given time.

Note that the special timer filter is required for useful operations when using this timer source.

2.2 Filters

A filters is a component responsible for filtering the objects registered in a source. Different kind of objects may therefore be handled differently.

There are five built-in filter types in FileTrain, some of which will work with any kind of source and some are useful only on some sources. More advanced filters may be developed as plug-ins.

For more detailed coverage of the different filters, see the configuration section.

Email filter

The email filter will apply to emails from an email source. Note that individual files will not be handled separately. The email will be treated as one object and handled as one. Do not use this filter if you wish to execute actions on attachments in an email. The actions following this filter will only have the actual email in scope (email body, sender etc) when they execute. This may be useful if you like to save text files with the email content.

If you like to execute actions on attachments you should use the file filter instead.

File filter

The file filter is filtering regular files, located locally or remotely on FTP or email servers.

With this filter you may set restrictions such as file name, file size, certain internal properties on images, PDF files or MP3 files and more.

Folder filter

Similar to the file filter but this filter will only handle folders.

Timer filter

This filter is a special filter which is needed if the station is holding a timer source. This filter is not a regular filter in terms of restrictions. A file filter may for example have a restriction to only let files with file suffix 'jpg' go through the filter. The timer filter will not have any restrictions but will only be passed if the source is a timer source that fires.

2.3 Actions

The actions are the components in FileTrain which will invoke a special handling upon an object, e.g. *moving* a file. In Many different kinds of actions are shipped with FileTrain. Other actions such as advanced database handling may be developed as plugins.

For more detailed information about each individual action, see the configuration section.

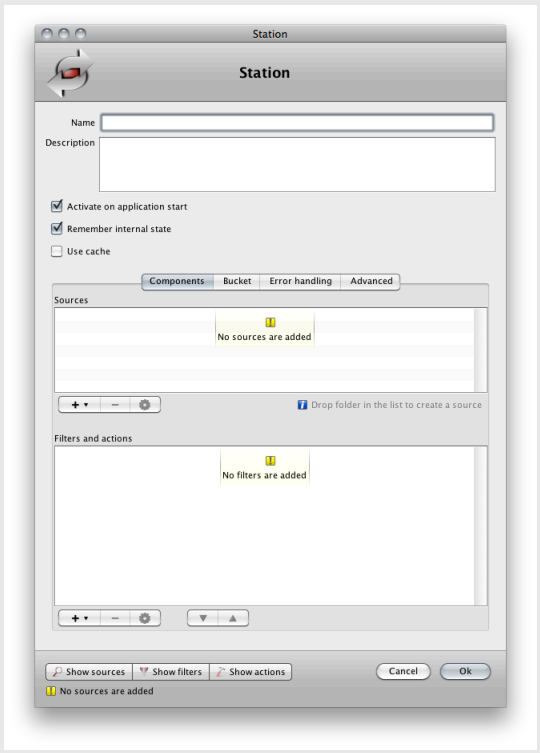
Error handling action

The three main component types; source, filter and actions, are put together in a station. The sources and actions may, depending on their types, experience problems when they are working, e.g. a folder source might not find the root folder or a FTP action may not be able to connect to the FTP server. When such an error occur the user may setup special actions for these events. These special actions are called error handling actions.

2.4 How to setup a workflow

All configuration parts in FileTrain are collected under the configuration node in the main window of FileTrain. Here you can see all the individual components as well as the stations. To edit a certain component you may double click on the item and the corresponding setup window is opened. In the setup window for stations there are lists with the individual components, these components are also clickable for quick editing.

When a station is created or edited the station setup dialog will appear, see image below.



Station setup window, component tab

In the station setup window all the sources, filters and actions are shown. There are also a special tab for error actions (the error list depends on the type of sources and actions in the station).

The name and description of a station is only for your personal use and will have no impact on the workflow.

Activate on application start should be selected if you would like this station to start automatically when FileTrain starts. There will be a initial built-in delay when FileTrain starts up before the station actually starts in case you want to stop one or more stations that are set to automatically start when FileTrain starts.

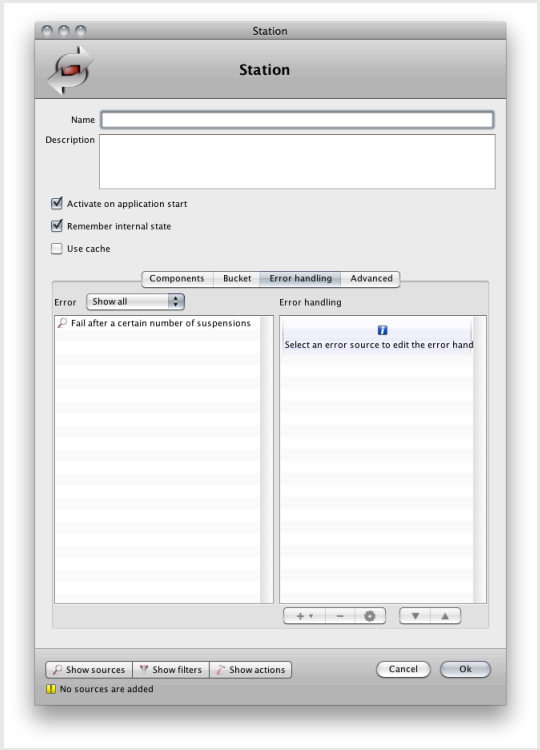
For more information about the **internal state** and **cache**, please read the section below about cache.

The source list shows all the sources that are present in this station. You may see a list of all existing sources by clicking the 'Show sources' button. You may drag and drop sources from this list to the source list in the station. You may also add a new source with the add button under the source list. If you haven't yet created any source you may create a new source from the add button.

The filter/action tree can be populated with filters and corresponding actions for that filter. To add a filter, drag a filter from the shown filter list (click the 'Show filters' to show this list) or by clicking the add button under the tree. To add actions to a specific filter either drag and drop actions from the action list (click the 'Show actions' to show this list) or by first selecting a filter and then click the add button under the tree.

> **Note**
>
> The individual order of the filters is very important.
>
> If an object being filtered is passing a filter, that particular filter's actions are invoked. If the object remains at the same location after the actions the object is continued in the next filter and may thus be handled by actions in many different filters.
>
> The action order is also very important since the actions are invoked in order. If one action for example is moving the object to a new location the rest of the actions are not going to handle the object since it has been removed.
>
> Use the up and down arrows below the filter and action tree to change the order of the filters and actions.

When all the sources, filters and actions are in place the station is theoretically ready to run. The individual components in a station may however cause errors and these errors can be taken care of in different ways. The station setup dialog has a tab for error handling.



Station setup dialog, error handling

In the error list to the left all the possible errors are seen (with the error popup above the list the errors from individual components may be viewed separately). Selecting an error in the error list will present the error handling action list for that particular error. With the add button under the error handling list an existing action may be added or a new error handling may be created.

The error shown at the top is special error which may occur if actions fails over and over in a station. In the advanced tab you may setup a maximum number of failures that you accept in this station. When this number of failures occurs for a certain object this object will be handled in this special error action.

> **Tip**
>
> If more than one error type is selected in the list on the left, all the changes of the error handling actions are affected by all the selected errors. If you for example want a certain error handling for all the errors, just select all the errors and add the specific error handling.
> Note that any individual error handling setting is lost if adding an error handling to multiple errors.

In the advanced tab you may change how objects are being handled in the station.

Station setup dialog, advanced setup

The max concurrent handling is default set to one (1). This is the number of objects that may be handled in parallell in this station. If you for example use a source that detects 100 files in one scan, each file is handled separately in the station and only one file is handled at a time. If you change the concurrent handling to for example 5, 5 files will be handled at the same time. Be careful when changing this number. If you set a high number of concurrent handling the CPU usage will increase and the handling may in fact get slower if this number is too high and the sources/actions uses a lot of network communication such as FTP server connection.

Suspension will occur when an action in this station fails its handling. The object that are being handled are being put to suspension for the specified suspension timeout. When this occurs the next object in the queue will be handled in the station. When the suspension time is over the object is being handled again but actions that have already been invoked on the object will not be executed again. An object will be suspended and the action will try to handle the object over and over again. If you like to have a maximum number of suspensions before the object will be handled different (for example you may want to move the object after 10 failures to another folder) you may specify the number in this advanced setup.

2.5 Station and object buckets

The bucket is a holder for keys and values and may be very useful if its capabilities are fully understood.

There are two different buckets. Each station has their own global bucket where the values are kept even if FileTrain is stopped. There is also a temporary bucket which is created for each object that is handled.

Object bucket

This bucket feature was introduced in FileTrain version 4.0.7 to enable different actions to share and use data between eachother. The object bucket feature is very powerful and may for example be useful when one action wants to use information from another action.

An object's bucket lifetime starts when an object is detected in a source and started to be handled by a station and ends when all the actions have been executed. This means that for each object that is handled in a station a new bucket is created. The bucket's data is shared among the actions which are invoked on the object. Values that are put into the bucket can be used by any action that is performed after the value is put into the bucket.

A common way to get values from this bucket is with the macro **%BUCKET[...]%** (see the macro section for more details). Filters may also filter on a bucket value and there is also a special action to set the values for a bucket.

Before an object is being handled by the actions the station itself puts some core information into the bucket. Information about these is found in the table below;

Common bucket values set by all stations

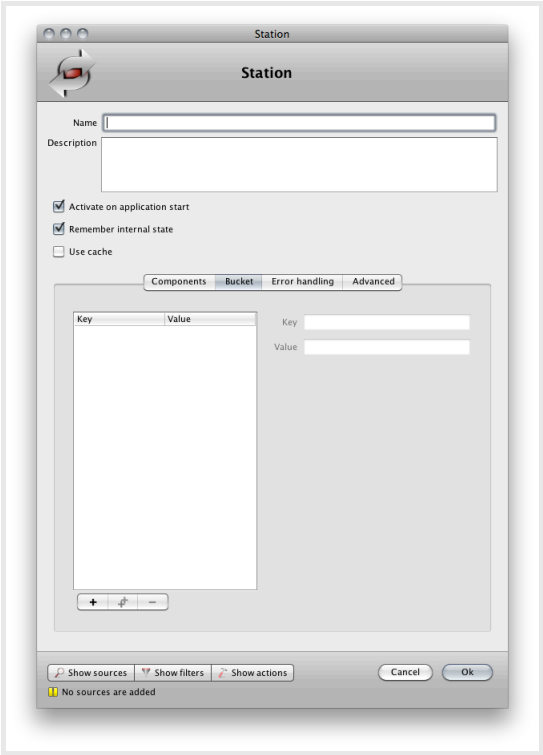
Bucket key	**Value description**
source.email	If the object being handled is an email the value contains the data about this email. More information for developers can be found in the plug-in API.
source.file	If the object being handled is a file the value contains a reference to this file. More information for developers can be found in the plug-in API.
source.exif.<file path>	If the object being handled contains exif information, this bucket value contains that data. Note that all data is loaded lazily, see the bucket key exifLoaded.
exifLoaded.<file path>	A boolean value to see if the exif data has been loaded or not.
source.imageData.<file path>	If the object being handled contains iamge information, this bucket value contains that data. Note that all data is loaded lazily, see the bucket key imageDataLoaded.
imageDataLoaded.<file path>	A boolean value to see if the image datahas been loaded or not.
source.iptc.<file path>	If the object being handled contains IPTC information, this bucket value contains the IPTC data. Note that all data is loaded lazily, see the bucket key iptcLoaded.
iptcLoaded.<file path>	A boolean value to see if the iptc has been loaded or not.
source.mp3.<file path>	If the object being handled contains MP3 tags, this bucket value contains that data. Note that all data is loaded lazily, see the bucket key mp3Loaded.

mp3Loaded.<file path>	A boolean value to see if the mp3 data has been loaded or not.
source.xmp.<file path>	If the object being handled contains XMP information, this bucket value contains that data. Note that all data is loaded lazily, see the bucket key xmpLoaded.
xmpLoaded.<file path>	A boolean value to see if the XMP has been loaded or not.
sourceFileName	If the currently handled object is a regular file the value will be the name of this file. Normally this is accessible through the macro %F%%E% but some actions may for example work on external files where %F% will refer to this external file name and thus getting the source file name is easily done using this bucket key.
sourceFilePath	If the currently handled object is a regular file the value will be the path to this file (excluding the file name).

Station bucket

As an addition to the object bucket a more global bucket was introduced in FileTrain 6.2. This station bucket is a more static bucket where the values will be saved with the station even if FileTrain is shut down. This station bucket may be used for example to have a running number or an internal log .

The station bucket may be seen in the setup of a station, see image below.



Station setup dialog, bucket setup

A common way to get values from this bucket is with the macro **%STATION_BUCKET[...]%** (see the macro section for more details). Filters may also filter on this bucket value and there is also a special action to set the values for the bucket.

Example with running number

If you like to have a running number which may for example be used to rename fiels passing a station you may use a combination of an external text file and macros reading this text file and writing new values to this text file. With the use of a station bucket this is now a much more simplier task.

You start with creating a new station bucket key. This is done in the bucket tab in the staiton setup dialog. We may for example create a new bucket item with key **Number** and let's assume we give it an initial value of **1**.

If we want all files that are detected in this station to be moved to a server location with this number attached we simply create a move action and in the name change field we can use the following text (to append a dash and the number)

%F%-%STATION_BUCKET[Number]%%E%

If we now want the number to increment for each file we will need to add an action after the move action which will be of type **bucket**. This action may set a bucket value. The key will be **Number** and the value will be the following (to add 1 to the number)

%MATH[+,1,%STATION_BUCKET[Number]%,0]%

Using the two actions described above each file will get their own unique number. If the station is edited you can always in the bucket tab see the current value for the key Number (and you may also manually edit the value.

For more advanced setups you may want to use a special timer source and filter in this station to reset the Number for example each hour or day.

2.6 Station cache and internal state

Each station has a very advanced internal cache system. This cache feature is used to prevent FileTrain to handle objects several times. The cache system will use resources and should be avoided when possible. The cache system consist of two parts, a **cache** and an **internal state**.

The **internal state** is a memory where the station will remember which actions have been executed on each object. The **cache** is a memory where the station will remember if an object has been handled or not. If you have a station where objects in the source location will be moved or deleted from the source location when FileTrain handles the objects you do not need to use any cache.

Example with cache

Imagine a station where all files will be handled and you have three actions.

The first action is copying the file to a server (server A), the second action is sending the file to a FTP server (server B) and the third is moving the file to another file server (server C).

A file is detected in the source folder and the station starts to handle the file. The first action copies the file successfully, the second action fails sending the file to the FTP and the station will therefor stop the handling and put the file on suspension. When the suspension is over the station starts the handling again.

If the station is set to remember the internal state, the station will not perform the first action again since that action is already executed. The station will start with the second action. If the station is not using internal state the first action will be executed each time.

In an environment where FileTrain is scanning a source location where the objects will not be moved or deleted you will need to use the cache to avoid that FileTrain handles the objects over and over again.

Each object in a source location is compared with the station's internal cache before it is handled. If the object name is the same but the modification time is changed, the object is handled as a new object in

FileTrain. The object is also handled as new if the object size is changed.

Workflow recommendation

We strongly recommend not to use FileTrain scanning source locations where you have a lot of objects and the objects will not be moved or deleted. This will cause FileTrain to keep a large cache and each object will be compared to this cache before handling may be started. If the source location contains a large number of objects this may cause FileTrain to be slower.

To avoid this, we recommend that FileTrain will be handling the objects before they are placed to the static storage.

2.7 Macro

Throughout all the setup of FileTrain you will find fields which are yellow instead of white. These fields are macro fields and may contain special macro text. The macro text is special dynamic text. The macro concept is explained in more detail in the macro section. The macro text enables you to setup FileTrain to very advanced workflows.

2.8 Troubleshooting

Symptom

FileTrain starts in demo mode even though I have entered my license key.

FileTrain shows error that preferences can not be written.

FileTrain acts strange and is slow (mainly FileTrain version 6.0.15 and earlier)

Cause

All the above symptoms may be related to lack of write permissions on some folders where FileTrain writes information.

Solution

Change the user permission on the following folders. The user that is running FileTrain must have write permission in the folder(s).

On Macintosh the following folders should have write permission:

/Library/Application Support

/Library/Preferences

<User>/Library/Preferences

<User>/Library/Caches

On Windows platform the following folders should have write permission:

<FileTrain installation folder>

3. Configuration

All the setup of stations and it's components are done directly from the FileTrain main application window in the configuration part. The different setup views when setting up stations and individual components are described more in detail in this section.

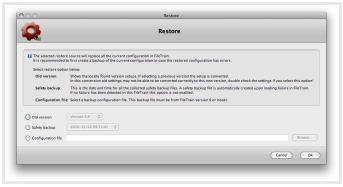
Configuration in FileTrain

Most individual parts of the configuration has export and import buttons. With these you may export/import individual parts of your configuration (compared with backup/restore which will save or load a full setup).

3.1 Backup and restore your setup

From the main configuration item you may create a backup of your setup at any time with the backup button.

You may also import a saved setup or a setup from a previous release of FileTrian with the restore button.



Restore setup in FileTrain

When you use the restore button the restore window will show three options. If old preferences are found on the local machine you will see the list of versions available. If there are any safety backups they will be available in the safety backup part and you may also select an individual backup file (created by the backup option).

> **Important when getting old version settings**
>
> If you restore setup from a previous version of FileTrain it is very important to overlook the setup after the restore is done. Older versions of FileTrain may be more or less incompatible with the current version of FileTrain. Therefor it is important that you manually look through the setup after such a restore to make sure that the settings are correct.

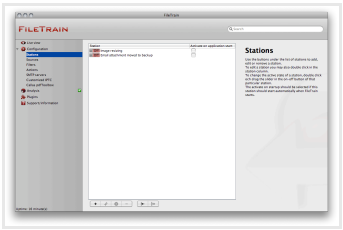
Version backup

From FileTrain version 6.4 old version setups can manually be found in the subfolder **-versionBackup** found in the general FileTrain preferences folder. These backup files are created automatically when you install a newer version of FileTrain. When you start the new version, FileTrain will automatically detect that your old setup is from another version and put this file in the backup.

If you for some reason needs to re-install your fileTrain to an earlier version the preferences may not be able to be read if you have run a newer version of FileTrain. In this case you can restore

3.2 Stations

In the stations configuration you may create new, edit or delete existing stations.



Station configuration

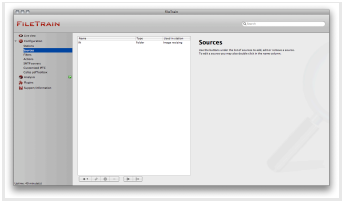
When editing a station the station setup window is shown. The setup of the station is explained in detail in the section 2.4 How to setup a workflow.

The stations have a little on/off slider attached next to their names. With this slider you control the current active state for the station. If the station is set to automatically start on startup the station will be active when FileTrain starts.

3.3 Sources

A source is a location where FileTrain will search for objects. The most common source is a folder on the local or remote network drive. Different sources may have different settings. FileTrain has four built-in sources, *Email*, *Folder*, *FTP folder* and *Timer*.

Sources may be developed as plugins to FileTrain. This enables other advanced sources such as different database sources, web loading etc. There are examples of very complex plugins developed for FileTrain where filtering and actions are included within the source which is working completely on it's own.



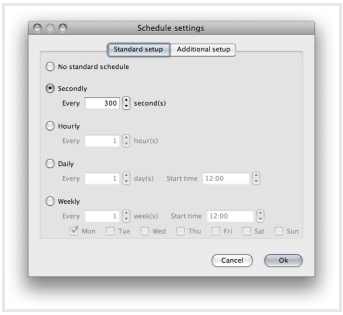
Source configuration

In the source configuration you may see in which station/stations the source is used. You may therefor quickly spot and delete sources that are not in use in any station. Most sources will have a setup for how often they will scan their source. This source schedule setup is described below.

Source schedule

A common setting in the sources are the scan interval which has soem advanced features. This schedule setup is used to decide when the source should make a scan (e.g. a folder source looks for new files in it's source folder).

The schedule is divided into two parts, one standard schedule setup and one additional setup.



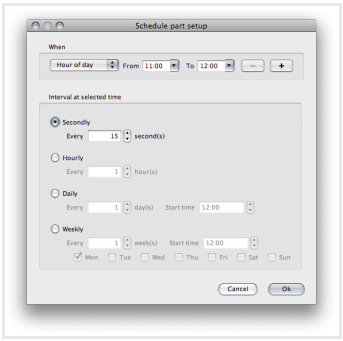
Standard schedule setup

There is an option in the standard schedule view to not use a standard schedule. If this option is selected only additional schedule will be invoked. The other options for the standard schedule are **secondly**, **hourly**, **daily** and **weekly**. Select the appropriate option for the source.

Secondly and hourly will scan every X second or hour.

The daily option has a time of the day when the scan will be done and an option to do this every day or every X day. If you for example use every 3 day(s) and the station starts at a monday, the next scan will be done on thursday (3 days ahead of monday).

The weekly option let you set a time of the day when to invoke the source and select which days of the week that should be included. Use this option if you for example want the source to be invoked mon-fri but not sat and sun.



Additional schedule setup

The additional schedule is a list of additions. The items in the additional setup will work complimentary to the standard setup.

When adding a new additional schedule part you first have to decide when this part should be active. You can add several 'when' settings and the schedule part will only be invoked when all the 'when' setup is ok.

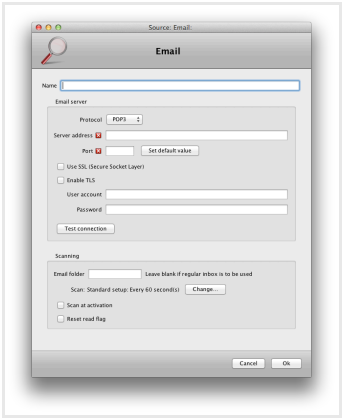
Below the when setup you select how often the schedule will be invoked during this when period.

> **Important note**
>
> If you use a setup which is impossible to achieve the source will not work at all

> **Combined setup for 'when'**
>
> If you for example want the additional schedule part to take place at monday and wednesday between 1 pm and 3 pm you select hour of day and put 13:00 and 15:00 in the from and to fields, then you add a new row with the add button on the right and you select day of week and select monday and wednesday.

Email source

The email source is a special source that is able to scan an email server account and handle the emails on this account. The protocols supported are POP3 and IMAP. See table below for more information about the settings.



Email source preferences

Settings for email source

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Protocol	Selector for the protocol to use, either POP3 or IMAP.
Server address	The email server name, either as IP number or DNS name, e.g. 192.168.40.1 or mail.yourserver.com.
Port	The email server port. If the default port should be used, click the button 'Set default value' after first selecting the correct protocol.
Use SSL	If this check box is selected FileTrain will use the secure socket layer whilst communicating with the email server.
Enable TLS	Enabling TLS communication with the email server
User account	The user login for the email account which should be monitored.
Password	The password for the user account above.
Email folder	Use this field to enter the name of the email folder holding the emails. Leave this field blank if you are using the regular inbox folder for the email account.
Scan	This describes how often the source will be scanned. To change the setting, click the 'Change' button. More information about the scan setup is seen in the source schedule section.
Scan at activation	If this check box is selected this source will do an initial scan when the source is started.
Reset read flag	If this check box is selected the email's read flag on the email server will be reset after FileTrain has handled them.

Folder source

A folder source is scanning a folder on a local or remote network drive. The monitored folder is accessed through the regular file system. See table below for more information about the individual settings.

Folder source preferences

Settings for folder source

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Folder	The folder to monitor. Click the 'Browse...' button to change the location.
Scan sub folders	If this check box is selected the sub folders in the folder will also be monitored and handled in the same way as the root folder. If the folder structure for found objects should be kept there is an option in relevant actions to keep the folder structure.

Scan	This describes how often the source will be scanned. To change the setting, click the 'Change' button. More information about the scan setup is seen in the source schedule section.
Safety margin	The safety margin is an interval during which objects need to stay intact. If an object is not changing during a safety interval it is considered to be ready for handling by FileTrain. By 'changing' FileTrain will consider the file size for regular files and the total file count and file size if the object is a folder.
Scan at activation	If this check box is selected the source will do an initial scan when the source is started.
Ignore invisible objects	If this check box is selected FileTrain will ignore objects that are considered to be invisible by the operating system.
Ignore zero sized objects	If this check box is selected FileTrain will ignore files that are zero (0) byte in size and empty folders.

Multi Folder source

The multi folder source is a more effecient source to use if many different source folders should be used in the same station.

Multi Folder source preferences

Settings for Multi Folder source

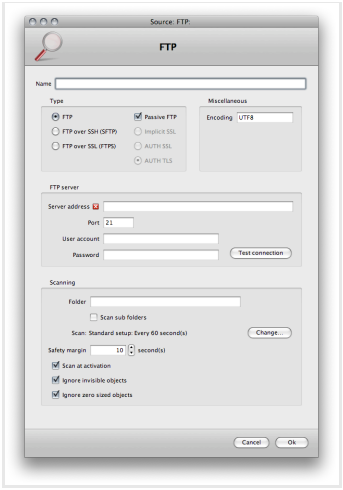
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Folder	The list of folders to monitor. Click the '+' and '-' buttons to add and remove folders. You may also drag and drop folders from the file system directly on the list to add folders.
Scan sub folders	If this check box is selected the sub folders in the folder will also be monitored and handled in the same way as the root folder. If the folder structure for found objects should be kept there is an option in relevant actions to keep the folder structure.
Scan	This describes how often the source will be scanned. To change the setting, click the 'Change' button. More information about the scan setup is seen in the source schedule section.
Safety margin	The safety margin is an interval during which objects need to stay intact. If an object is not changing during a safety interval it is considered to be ready for handling by FileTrain. By 'changing' FileTrain will consider the file size for regular files and the total file count and file size if the object is a folder.
Scan at activation	If this check box is selected the source will do an initial scan when the source is started.
Ignore invisible objects	If this check box is selected FileTrain will ignore objects that are considered to be invisible by the operating system.
Ignore zero sized objects	If this check box is selected FileTrain will ignore files that are zero (0) byte in size and empty folders.

FTP source

FileTrain may scan a remote folder located on a FTP server using the File Transfer Protocol. See table below or more information about the individual settings.

When objects are to be handled from the FTP server the objects are lazily downloaded to a temporary folder before actions are invoked on them. This makes FileTrain very effecient if many actions are to be invoked upon an object. The lazily download means that the objects are not downloaded unless they are really needed to be.

If the local temporary object is removed during action handling (for example if a move action is invoked) the server object will be deleted by FileTrain after handling is done.



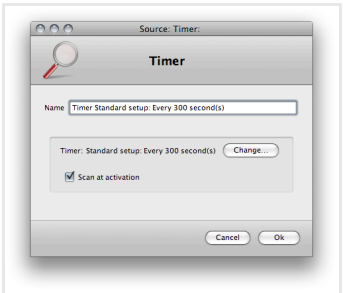
FTP source preferences

Settings for FTP source

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Type	The type settings enables you to select what type of FTP connection to use: **FTP:** A regular FTP connection, this is the most commonly used connection. **SFTP**: Use this if the FTP server is supporting FTP over SSH. **FTPS**: Use this if the FTP server is supporting FTP over SSL. Depending on the type selected various other selections are available such as passive FTP transfer etc.
Encoding	The default encoding is UTF8 when transfering data to and from the FTP server. If your FTP server is using another set of encoding you may specify that encoding here.
Server address	The FTP server name, either as IP number or DNS name, e.g. 192.168.40.2 or ftp.yourserver.com.
Port	The FTP server port. The default FTP port is 21.
User account	The user login for the account which should be monitored.
Password	The password for the user account above.
Folder	The folder to monitor. This is the path to the folder seen from the login location on the FTP server. Note that different user account may have different root folders on the FTP server. The folder specified in this field is the path from the root folder when logging in with the user account above. Be aware that a forward slash (/) may have to precede the folder path (depending on the FTP server).
Scan sub folders	If this check box is selected the sub folders in the folder will also be monitored and handled in the same way as the root folder. If the folder structure for found objects should be kept there is an option in relevant actions to keep the folder structure.
Scan	This describes how often the source will be scanned. To change the setting, click the 'Change' button. More information about the scan setup is seen in the source schedule section.
Safety margin	The safety margin is an interval during which objects need to stay intact. If an object is not changing during a safety interval it is considered to be ready for handling by FileTrain.
Scan at activation	If this check box is selected the source will do an initial scan when the source is started.
Ignore invisible objects	If this check box is selected FileTrain will ignore objects that are considered to be invisible by the system.
Ignore zero sized objects	If this check box is selected FileTrain will ignore files that are zero (0) byte in size and empty folders.

Timer source

The timer source is a special source which will not scan any server or folder but only work similar to an alarm clock. When the 'alarm' is due this source will tell the station to execute. The station must be setup with the special timer filter in order to actually do anything. The actions contained in the timer filter will be executed.



Timer source preferences

Settings for timer source

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Timer	This describes how often the source will 'set off'. To change the setting, click the 'Change' button. More information about this setup is seen in the source schedule section.
Scan at activation	If this check box is selected the source will 'set off' initially when the source is started.

3.4 Filters

Filters are filtering the objects detected in the sources. This enables advanced workflow where different kind of objects may be routed to different workflows.

There are five default filter types, **email**, **file**, **folder** and the more special **misc filter** and **timer filter**. When FileTrain is started without any filters present the user will be asked if default filters should be created. These default filters are some commonly used filters and their names will be preceded by three asterisks (***). The default filters are editable and may be changed by the user.

All filters have a name. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component. Furthermore the filters have a set of restrictions. To add a new restriction the button with a plus is clicked, to remove a restriction the button with a minus next to that restriction is clicked. Some restrictions have a check box called 'invert'. If this invert check box is selected the restriction settings for that particular row will be inverted.

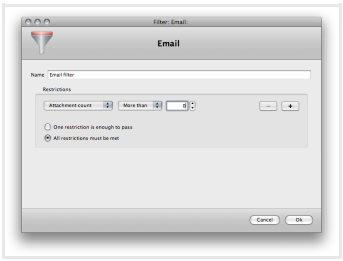
> **The invert check box**
>
> Using a file filter with the name restriction
>
> ***.pdf**
>
> will let any file with the file extension pdf pass the filter.
>
> If the invert check box is selected only files that does NOT have the file extension pdf will pass.

Below the restriction set there is a selection whether one of the restrictions is sufficient or if all the restrictions must be met.

Email filter

The email filter is a special filter being able to filter emails on a remote email server. An email source must have an email filter to be able to handle the email itself. If attachments of the email is to be handled a regular file filer should be used. Even if a file filter is used the macros to get certain email values such as subject or sender are availlable when actions handle the attachment file. If you use the email filter the actions for this filter will only have the email itself to work with, not the attachment files.

> **Attachments**
>
> To filter on individual attachments, use a file filter instead.



Email filter preferences

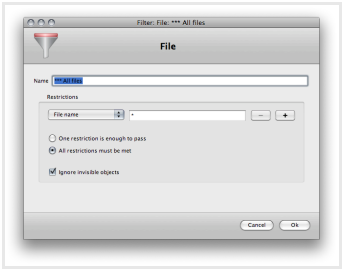
Restrictions for email filter

Restriction	Description
Attachment name	This will filter on any of the attachment's name. Asterisk (*) may be used as a wild card meaning any character (0, 1 or more characters). Question mark (?) may be used as a wild card where there must be a character but that character may be any character. If different names are wanted you separate them with commas. For example, if files that ends with '.eps' and files that ends with '.ps' should be filtered you enter '*.eps, *.ps' in the file name field. Note that it is enough that one attachment is passing this filter for the email to be handled. Also note that if the attachment itself is wanted to be handled, use a regular file filter instead.
Attachment count	This will compare the email's attachment count and if equal, less than or greater than the specified number the email will be handled.
Email sender	This will compare the email sender with the text entered. The same rules with asterisk, question marks and commas applies as for attachment name and email body.
Email body	This will compare the email body with the text entered. The same rules with asterisk, question marks and commas applies as for attachment name and email sender.
Email sent	This is the time when the email was sent. If the send time compared to current time is larger than the specified time, the email is handled.
Time from discovery	This is the time from the moment FileTrain detects the object. When the specified time has elapsed the object is handled.
Bucket	This filter will check a specific bucket key. The filtering may check if the bucket value is equal to a certain value or if the bucket value contains a text or if the bucket value at all exists. You must select if the bucket key that is checked is from a station bucket or object bucket, read more about the different buckets in chapter 2.

File exists	This filter may check if an external file with the path specified exists or not.

File filter

The file filter is filtering regular files as well as FTP files and attachments in emails.



File filter preferences

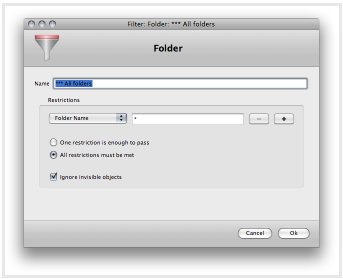
Restrictions for file filter

Restriction	Description
File name	This will filter on the file name. Asterisk (*) may be used as a wild card meaning any character (0, 1 or more characters). Question mark (?) may be used as a wild card where there must be a character but that character may be any character. If different names are wanted you separate them with commas. Example, if files that ends with '.eps' and files that ends with '.ps' should be filtered you enter '*.eps, *.ps' in the file name field.
Size	This denotes the file size. You may select that the file size should be equal, less than or bigger than a certain byte count.
Modification time	This is the file modification time. If the modification time compared to current time is larger than the specified time, the file is handled.
Time from discovery	This is the time from the moment FileTrain detects the file. When the specified time has elapsed the file is handled.
IPTC	This feature is reading the actual IPTC information in the file (works only with jpg and tiff files). Select the IPTC field in the popup and select one of the following; *Equals*: The IPTC field value should be exact as specified. *Contains*: The IPTC field should contain the value specified. *Has value*: The IPTC field should contain something.
PDF format	This feature is only working on PDF files. You may choose to check any of the following boxes; *Media box, Crop box, Bleed box, Trim box or Art box* and you may check either width or height or both whether they are less than, bigger than or equals a certain number in millimeters.
PDF version	This feature is only working on PDF files. Specify the version number, e.g. 1.3 and select whether the version on the files has to be this version, must be less than or greater than this version.
XMP	This will filter the file on the XMP content. This is only applicable to JPG, PDF and TIFF files. Select the XMP property in the popup and select one of the following; *Equals*: The XMP value should be exact as specified. *Contains*: The XMP value should contain the value specified. *Has value*: The XMP value should be present in the file.
MP3 album	This will look at the ID3 information of the file and filter on the album tag. (works only with MP3 files). The same special characters as for the *file name* restriction can be used.
MP3 artist	This will look at the ID3 information of the file and filter on the artist tag. (works only with MP3 files). The same special characters as for the *file name* restriction can be used.
MP3 comment	This will look at the ID3 information of the file and filter on the comment tag. (works only with MP3 files). The same special characters as for the *file name* restriction can be used.
MP3 genre	This will look at the ID3 information of the file and filter on the genre tag. (works only with MP3 files). The same special characters as for the *file name* restriction can be used.
MP3 title	This will look at the ID3 information of the file and filter on the title tag. (works only with MP3 files). The same special characters as for the *file name* restriction can be used.
MP3 track	This will look at the ID3 information of the file and filter on the track tag. (works only with MP3 files). Put the number in the input area and select one of the following; *Equals*: The ID3 value should be exact as specified. *Less than*: The ID3 value should should be less than the value specified. *More than*: The ID3 value should should be morethan the value specified.
MP3 track count	This will look at the ID3 information of the file and filter on the track count tag. (works only with MP3 files). Put the number in the input area and select one of the following; *Equals*: The ID3 value should be exact as specified. *Less than*: The ID3 value should should be less than the value specified. *More than*: The ID3 value should should be morethan the value specified.
MP3 year	This will look at the ID3 information of the file and filter on the year tag. (works only with MP3 files). Put the year in the input area and select one of the following; *Equals*: The ID3 value should be exact as specified. *Less than*: The ID3 value should should be less than the value specified. *More than*: The ID3 value should should be morethan the value specified.
Color space	This restriction will pass files that has the specified color space.
Image resolution	Checks the x and/or the y resolution in DPI.

Image dimensions	Filter on the actual dimensions of the image.
Bucket	This filter will check a specific bucket key. The filtering may check if the bucket value is equal to a certain value or if the bucket value contains a text or if the bucket value at all exists. You must select if the bucket key that is checked is from a station bucket or object bucket, read more about the different buckets in chapter 2.
File exists	This filter may check if an external file with the path specified exists or not.

Folder filter

The folder filter handles folders. Please note that some of the restrictions may not work for remotely located folders. See table 3.3-5 for more information about the restrictions for this filter.



Folder filter preferences.

Restrictions for folder filter

Restriction	Description
Folder name	This will filter on the folder name. Asterisk (*) may be used as a wild card meaning any character (0, 1 or more characters). Question mark (?) may be used as a wild card where there must be a character but that character may be any character. If different names are wanted you separate them with commas.
Size	This denotes the number of items within the folder (both files and folders). You may select that the count should be equal, less than or bigger than a certain number.
Modification time	This is the folder modification time. If the modification time compared to current time is larger than the specified time, the folder is handled.
Time from discovery	This is the time from the moment FileTrain detects the folder. When the specified time has elapsed the folder is handled.
Bucket	This filter will check a specific bucket key. The filtering may check if the bucket value is equal to a certain value or if the bucket value contains a text or if the bucket value at all exists. You must select if the bucket key that is checked is from a station bucket or object bucket, read more about the different buckets in chapter 2.
File exists	This filter may check if an external file with the path specified exists or not.

Misc filter

The misc filter is a special filter which may be used in various situations. This filter have the special restriction 'pass all' which will allow anything through the filter.

Misc filter preferences.

Restrictions for misc filter

Restriction	Description
Allow all	Lets all objects through.
Bucket	This filter will check a specific bucket key. The filtering may check if the bucket value is equal to a certain value or if the bucket value contains a text or if the bucket value at all exists. You must select if the bucket key that is checked is from a station bucket or object bucket, read more about the different buckets in chapter 2.
File exists	This filter may check if an external file with the path specified exists or not.

Timer filter

Special filter used with timer sources. No extra restrictions are used in this filter.

3.5 Actions

Objects that have been filtered might be handled in different ways. This is where the actions are used. An action is a 'handling' invoked on an object. This handling may affect the object (for example moving the object) or it may just use the object as reference for doing some other action (for example sending an email).

FileTrain has some default actions. However actions may be developed as plugins to FileTrain. Therefore very advanced actions may be invoked on an object. This may involve advanced merging of documents, updating of databases or other repositories or other integrations with existing systems and/or applications.

Each built-in action is described in more detail below where you will find information about what the action is handling. Some actions may add values in the station bucket, this information is also found in each description below.

> **Email handling**
>
> If an email is being filtered in the special email filter (see above) the email itself is treated but not the individual attachments.
>
> If on the other hand an email is filtered in a file filter the attachments will be treated individually.

Default actions overview

Applescript	Backup	Bucket	Callas pdfToolbox	Cargo - add file to job
Cargo - attach file	Cargo - create job	Cargo - update file	Char converter	Command line
Copy	Database	Delete	Email send	Email Copy
Email Move	Expand **(do not use)**	Feedback Upload	For each	
FTP send	Image processing [ImageMagick]	Image processing [SoftColor]	IPTC inserter	Move
Notifier	QuarkXPress PDF renderer	File permission	Stop station	Storage
Text file	Unzip	Wait	XMP inserter	

Applescript action

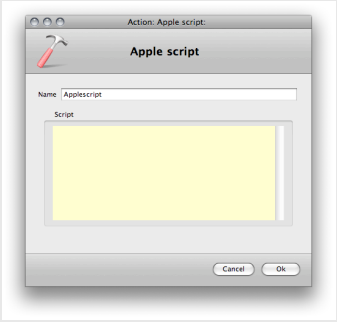
Handles
- Email
- Files
- Folders

This action is only available on the Macintosh platform. This action will execute an applescript.

The object that caused this action to execute is not affected by this action unless the applescript itself is having an effect on the object.

Table 3.5-1 Bucket fields affected by the applescript action

Bucket key	Value	Comment
ScriptResult-*<name of action>*	If the script that is executed returns a value this value is stored in this bucket value.	The <name of action> is the name of the FTP action that is given in the preferences. If you for example has a FTP action that is named 'execute script' the file list will be found in the bucket with key 'ScriptResult-execute script'.



Applescript action preferences

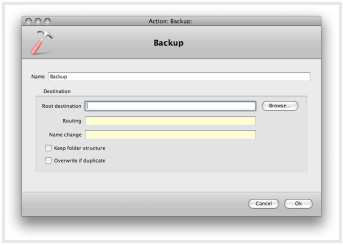
Settings for the applescript action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Script	This is the sript to execute.

Backup action

Handles
- Email
- Files
- Folders

This is a special version of the copy action which is suitable when using the copying as a backup. The difference compared to a copy action is that the backup action will first see if the file exists in the backup location. If the file exists and is the same (same size and modification time) it will be ignored and not added again by the backup action. If the file exists but is not the same the duplication handling will be used to determine what to do.



Backup action preferences.

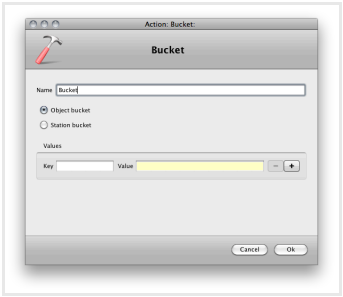
Settings for the backup action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Bucket action

Handles
- Email
- Files
- Folders

This action may be used to set one or more values in the current station's bucket. This may be very handy if you for example use the station for email handling and have a filter to handle only certain type of files but later would like to move all emails where such files have been found. You may in that case set a certain bucket value after handling the file. A latter email filter may then use this bucket value to filter only those emails that contained this type of file to handle the email.



Bucket action preferences.

Settings for the bucket action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Object bucket or station bucket	Select whether this action will affect the station bucket or the object bucket. Read more about the different buckets in chapter 2.
Values	Bucket setup is one or more bucket keys and values. The key is the bucket key and the value its value.

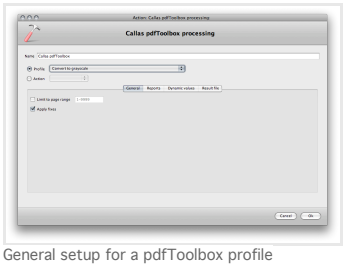
Callas pdfToolbox action

Handles
Email
Files
Folders

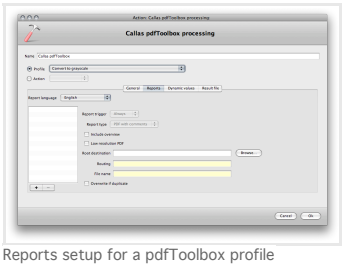
This action will use Callas pdfToolbox to perform an action or profile. To be able to use this action the Callas pdfToolbox configuration setup needs to be properly set. When the Callas pdfToolbox folder is set all the local profiles are loaded and analyzed in FileTrain. This loading may take a couple of minutes. Once the loading has finished the profiles may be used.
For more information regarding the setup in profiles or actions, see the Callas pdfToolbox documentation found at www.callassoftware.com.

Profiles

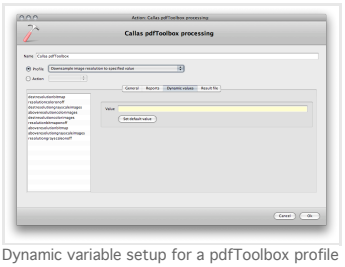
To perform a profile upon a file the profile radio button should be checked and the wanted profile may be selected. All the profiles have the same setup as described below.
One or more reports may be created in a profile.



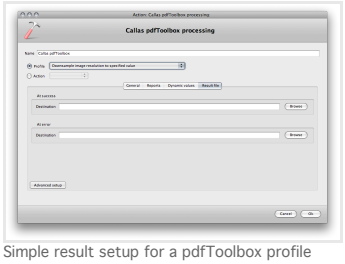
General setup for a pdfToolbox profile



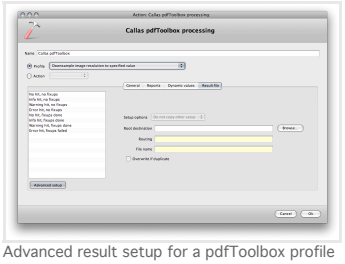
Reports setup for a pdfToolbox profile



Dynamic variable setup for a pdfToolbox profile



Simple result setup for a pdfToolbox profile



Advanced result setup for a pdfToolbox profile

Settings for the pdfToolbox profiles

General setup	
Setting	**Description**
Limit to page range	If the profile only should handle a certain page range this range should be entered here.
Apply fixes	If this option is not selected the fixups of the profiles will not be performed.
Reports setup	
Setting	**Description**
Report language	The wanted localization in the reports.
Report trigger	This selection determins when the current report should be created.
Report type	The type of report that should be created.
Include overview	If this option is selected the report will contain an overview. This option is only affecting PDF type reports.
Low resolution PDF	Select this option if the report should be created as a low resolution PDF.
Root destination	This is the root folder where the report will be saved. If the routing field (see below) is empty the report will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
File name	The file name for this report.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.
Dynamic value setup If the profile that is selected have dynamic variables these variables will be shown in the variable list. When a variable is selected the default value is first shown. This default value may be changed. To restore the variables value to the default, use the button under the value field.	

Result file setup

The result file setup has two views, one simple and one advanced. The simple setup contains the path to where a result file should be saved. There are two different paths depending on the outcome fo the profile, one for success and one for errors. The error path is used if an error hit is detected in the file while running the profile.

The advanced setup is used to set options for individual outcomes from the process.

Setting	Description
Setup options	If the currently selected outcome should be linked to the properties from another outcome, select what outcome to be linked to.
Root destination	This is the root folder where the result will be saved. If the routing field (see below) is empty the file will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
File name	The file name for this file.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

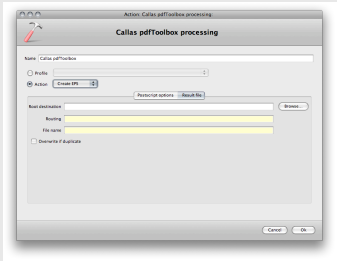
Actions

To execute a certain action the action radio button should be checked and the wanted action may be selected. Different actions have different setup as described below.

There are three actions implemented in this version of FileTrain:

- Create EPS
- Save as image
- Split PDF

All the actions have one setup tab in common, the **result file setup**. This setup describes where the output from the action should be placed. The individual action setup is described for each action below.

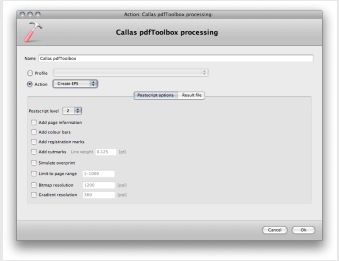


Result file setup for a pdfToolbox action

Settings for the pdfToolbox action result file

Setting	Description
Root destination	This is the root folder where the result file(s) will be saved. If the routing field (see below) is empty the file(s) will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
File name	The file name for the result file(s).
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Create EPS

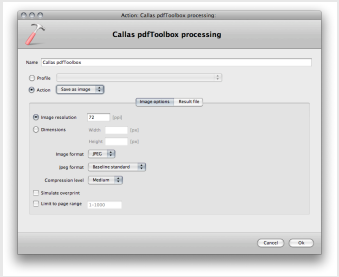


General setup for creating an EPS

Settings for the create EPS action in pdfToolbox

Setting	Description
Postscript level	The level of the postscript for the result file.
Add page information	If selected the result file will contain page information.
Add colour bars	If selected the result file will contain colour bars.
Add registration marks	If selected the result file will contain registration marks.
Add cutmarks	If selected the result file will contain cutmarks with the selected line width.
Simulate overprint	If selected the result file will simulate overprint.
Limit to page range	If the action only should handle a certain page range this range should be entered here.
Bitmap resolution	The final resolution on bitmap objects.
Gradient resolution	The final resolution on gradient objects.

Save as image

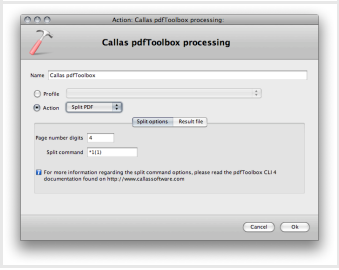


General setup for saving an image from PDF

Settings for the save as image action in pdfToolbox

Setting	Description
Image resolution	If selected the final resolution can be set.
Dimensions	If selected the dimension of the result image may be set.
Image format	The image format of the result file.
Jpeg format	If the image format is set to JPEG the format of the JPeg may be specified.
Compression level	The compression level of the result file if the image format supports this.
Simulate overprint	If selected the result file will simulate overprint.
Limit to page range	If the action only should handle a certain page range this range should be entered here.

Split PDF



General setup for splitting a PDF

Settings for the split PDF action in pdfToolbox

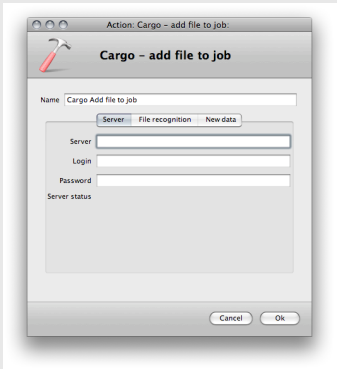
Setting	Description
Page number digits	The number of characters to use in the result file name where the page number is located.
Split command	This setup may be used to split a PDF in various ways. For more information see the Callas pdfToolbox CLI documentation found at www.callassoftware.com

Cargo - add file to job action

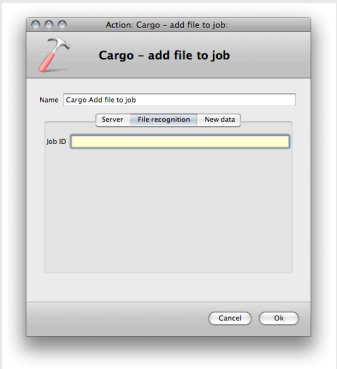
Handles
- Email
- Files
- Folders

This is a special action used in a system with Cargo.

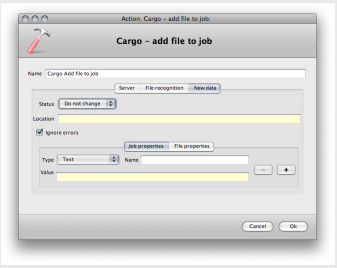
> The actions for Cargo requires a connection to a Cargo server. The connction is the full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive.
>
> When the cargo action is opened the action setup is trying to connect to the Cargo server to get necessary values from the server. If the user clicks the ok button before such a connection is made the settings may be damaged and the action may not work properly. Therefor it is important to always wait on the connection to finish. When the connection has finished the server tab is either showing a green check or a red cross to indicate successful or failed connection.



Cargo add file action preferences.



Cargo add file action preferences.



Cargo add file action preferences.

Settings for the Cargo add file action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Server	The full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive. Example) **http://localhost:8080/cargo/cargoServlet**
Login	The Cargo user login to use when connecting. The rest of the setup may depend on the login in terms of available profiles, status values etc.
Password	The password for the Cargo login.
Job ID	This is the ID of the job to which the file that is currently handled will be added.
Status	The new status for the file that is being added. If the selection 'do not change' is selected the status will be set by the profile to which the job belongs.
Location	This is the location that will be saved in the Cargo server to be the location of the file. Normally this would be **%FILE_PATH%** but if the file is mvoed to another location after this action this other location may be used instead.
Ignore errors	If this option is selected any error reported from the Cargo server will be ignored.
Job/File properties	This is a set of job/file properties that will be set when adding the file.

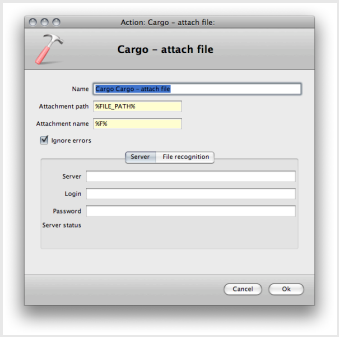
Cargo - attach file action

Handles
- Email
- Files
- Folders

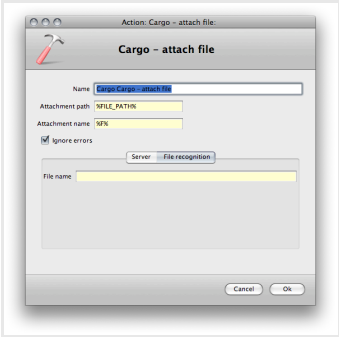
This is a special action used in a system with Cargo.

The actions for Cargo requires a connection to a Cargo server. The connction is the full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive.

When the cargo action is opened the action setup is trying to connect to the Cargo server to get necessary values from the server. If the user clicks the ok button before such a connection is made the settings may be damaged and the action may not work properly. Therefor it is important to always wait on the connection to finish. When the connection has finished the server tab is either showing a green check or a red cross to indicate successful or failed connection.



Cargo attach file action preferences.



Cargo attach file action preferences.

Settings for the Cargo attach file action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Attachment path	The file path to the attachment. Normally this would be **%FILE_PATH%**.
Attachment name	The name of the attachment when seen in the Cargo Dispatch interface.
Server	The full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive. Example) **http://localhost:8080/cargo/cargoServlet**
Login	The Cargo user login to use when connecting. The rest of the setup may depend on the login in terms of available profiles, status values etc.
Password	The password for the Cargo login.
File name	The name including the file extension of the file in the Cargo system to which this file will be attached.

Cargo - create job action

Handles
- Email
- Files
- Folders

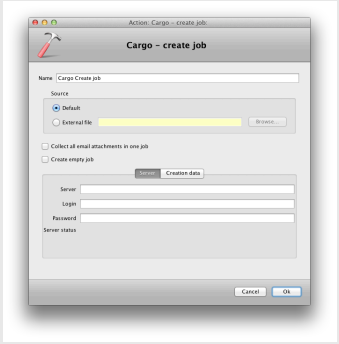
This is a special action used in a system with Cargo.

The actions for Cargo requires a connection to a Cargo server. The connction is the full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive.

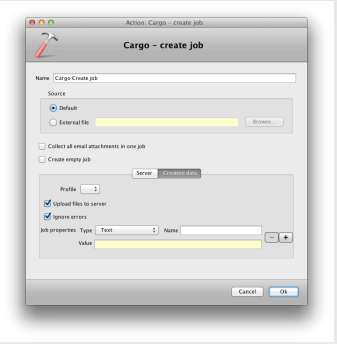
When the cargo action is opened the action setup is trying to connect to the Cargo server to get necessary values from the server. If the user clicks the ok button before such a connection is made the settings may be damaged and the action may not work properly. Therefor it is important to always wait on the connection to finish. When the connection has finished the server tab is either showing a green check or a red cross to indicate successful or failed connection.

Bucket fields affected by the FTP action

Bucket key	Value	Comment
jobID-*<name of action>*	The created job ID in the Cargo server.	The <name of action> is the name of the action that is given in the preferences. If you for example has an action that is named 'create job' the result job ID will be found in the bucket with key 'jobID-create job'.



Cargo create job action preferences.



Cargo create job action preferences.

Settings for the Cargo create job action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Collect all email attachments in one job	This option is only useful when this action is part of a station where the source is an email source. If this option is selected, all the current email's attachment conforming to the filter in which this action is placed will be added to the job that is being created. The other attachments will be prevented to create their own jobs. Example) If you have an email source and use a file filter with restriction size > 100 KiB and as action you have a Cargo Create Job with the option to collect all email attachments set. If an email has two attachments where both files are sized > 100 KiB, there will only be one job created where both the attachment files will be added.
Create empty job	If this option is selected, the job will be created without any initial file in it. Any file/email attachment etc that are being handled in this action will not be added to the job.
Server	The full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive. Example) **http://localhost:8080/cargo/cargoServlet**
Login	The Cargo user login to use when connecting. The rest of the setup may depend on the login in terms of available profiles, status values etc.
Password	The password for the Cargo login.
Profile	The profile on the Cargo server to use when creating the job. The user in the login setting may have restricted access to the profiles on the Cargo server.
Upload files to server	If this option is selected the files will be uploaded with HTTP to the Cargo server. Otherwise they will be handled from the location they currently have.
Ignore errors	If this option is selected any error reported from the Cargo server will be ignored.
Job properties	This is a set of job properties that will be set when adding the file.

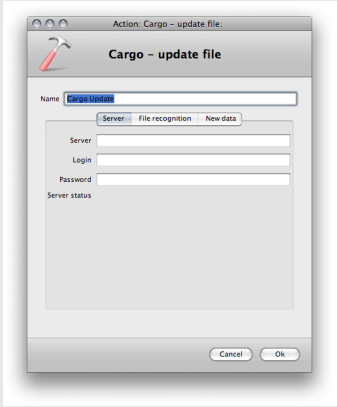
Cargo - update file action

Handles
- Email
- Files
- Folders

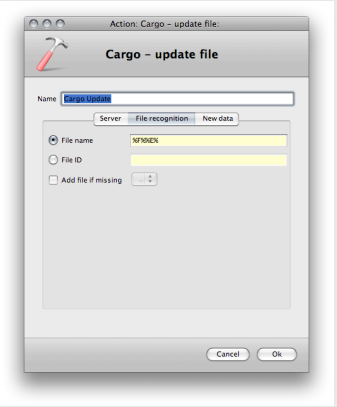
This is a special action used in a system with Cargo.

The actions for Cargo requires a connection to a Cargo server. The connction is the full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive.

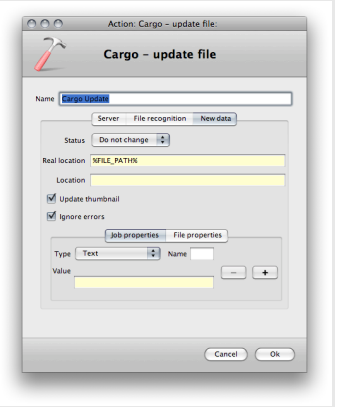
When the cargo action is opened the action setup is trying to connect to the Cargo server to get necessary values from the server. If the user clicks the ok button before such a connection is made the settings may be damaged and the action may not work properly. Therefor it is important to always wait on the connection to finish. When the connection has finished the server tab is either showing a green check or a red cross to indicate successful or failed connection.



Cargo update file action preferences.



Cargo update file action preferences.



Cargo update file action preferences.

Settings for the Cargo update file action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Server	The full http path to the Cargo servlet. Normally this servlet is located in the cargo folder on the web server and the name is 'cargoServlet'. The server path is case sensitive. Example) **http://localhost:8080/cargo/cargoServlet**
Login	The Cargo user login to use when connecting. The rest of the setup may depend on the login in terms of available profiles, status values etc.
Password	The password for the Cargo login.
File name	The file that should be updated could be identified with file name or Cargo file id. The file name field may be separated with a pipe character (|) in which case different file name options are available.
File ID	If the Cargo file id is known this can be used to indicate which file that is being updated.
Add file if missing	If the file is not found on the server it may be added to the selected profile (a new job with the file will be created).
Status	The new status for the file that is being updated. If the selection 'do not change' is selected the status will stay the same as it is on the Cargo server.
Real location	This is the current actual location of the file that is being updated (in order to create thumbnails and other data). Normally this would be **%FILE_PATH%**
Location	This is the location that will be saved in the Cargo server to be the location of the file. Normally this would be **%FILE_PATH%** but if the file is mvoed to another location after this action this other location may be used instead.
Update thumbnail	If this option is selected the Cargo server will update it's thumbnail and preview of the file.
Ignore errors	If this option is selected any error reported from the Cargo server will be ignored.
Job/File properties	This is a set of job/file properties that will be set when updating the file.

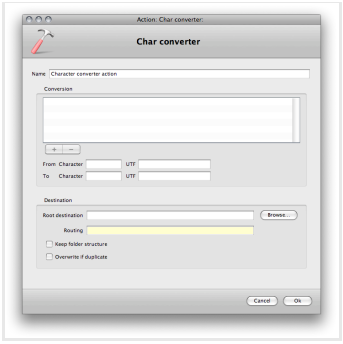
Char converter action

Handles
- Email
- Files
- Folders

This action will replace characters in a file name and then move the file to another destination. This action could be seen as a move action with the additional replacing of characters before the move.

To add conversion of characters you simply place the character(s) you want to replace in the 'From Character' field and in the 'To Character' field you enter what the new characters should be (leave empty if you want to remove the characters entered in the from field. Then you click the add button under the conversion list to add this replacement scheme. You may add as many conversions as you like.

There is also a UTF field for both the from and to fields. When you enter a character in the character field it's corresponding UTF value will be shown in the UTF field. If you enter more than one character the UTF field will have all the characters UTF values separated by comma (,). If you can not find the character you want to replace on your keyboard you may enter it's UTF value in the UTF field and the corresponding character will be shown in the character field.



Character converter action preferences.

Settings for the character converter action

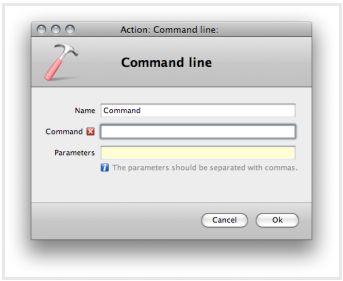
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Conversion	This list holds all the conversions. To add a new conversion, add the values in the from and to character fields (or use the UTF fields) and click the add button. To remove a conversion from the list, select it and click the remove button.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Command line action

Handles
- Email
- Files
- Folders

This action will execute a command in the command shell environment on the computer.

The object that caused this action to execute is not affected by this action unless the command itself is having an effect on the object.



Command line action preferences.

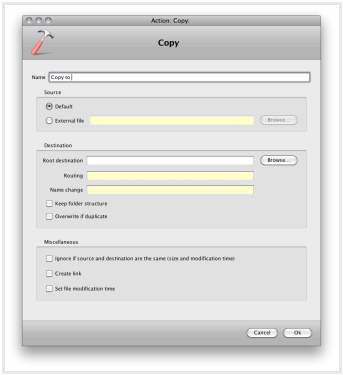
Settings for the command line action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Command	This is the main command to execute in the command prompt.
Parameters	This is optional parameters to the command. Each parameter must be separated with a comma (,).

Copy action

Handles
- Email
- Files
- Folders

The copy action will copy a file to a new location. The original object will remain intact.



Copy action preferences.

Settings for the copy action

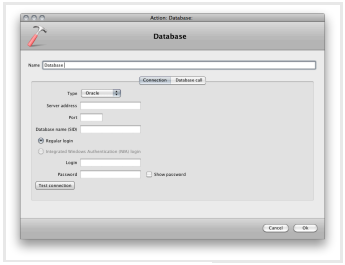
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.
Ignore if source and destination are the same (size and modification time)	If this option is selected the copying will not be performed if the destination file exists and is regarded to be the same (size and modification time are compared).
Create link	If this option is selected the destination will be a link to the original file.
Set file modification time	If this option is selected the copied file will get the current time as modification time.

Database action

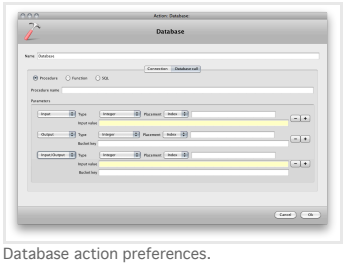
Handles
- Email
- Files
- Folders

This action may be used to access and perform SQL calls or procedure/function calls.

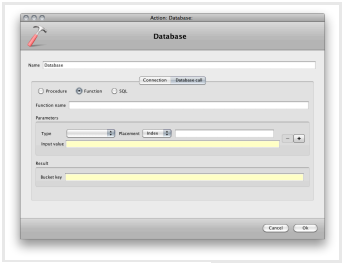
Note that if a SQL call returns multiple result rows the full list of result rows will be put in a bucket. The key of this bucket is the same key as stated in the setup but with the trailing **-list**, e.g. if you have the following SQL query: **SELECT paperName from paper** and the result column **paperName** with bucket key **pName**, there will be a bucket key **pName-list** which holds all the values from the query. Use for example the macro %BUCKET_LIST[...]% to get all the values from this list.



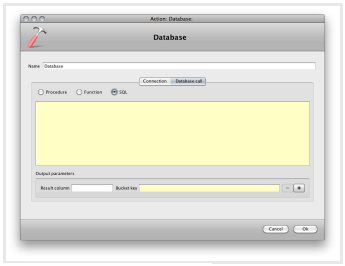
Database action preferences.



Database action preferences.



Database action preferences.



Database action preferences.

Settings for the database action

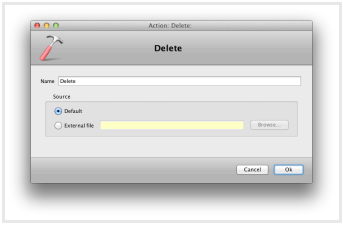
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Connection - Type	The type of database, the default types are Oracle and MySQL. If any other database type is wanted, please contact Laidback Solutions AB.
Connection - Server address	The IP number or DNS host name for the database server.
Connection - Port	The port to use when connecting to the database server.
Connection - Database name (SID)	The SID or database instance name to connect to.
Connection - Login	The user login to use when connecting to the database server.
Connection - Password	The password for the user login.
Database call	The database action supports regular SQL calls, procedure calls and function calls with separate setups.
Procedure	
Procedure name	The name of the procedure in the database.
Parameters	This is all the parameters to the procedure. Each parameter may be **input, output** or **input/output.** The type of the parameter is the data type. The placement is eitther indexed or key based. If indexed, write the index of the parameter (starting at 1) in the field. If key based, write the key in the field. For an input parameter there will be a field where the input value may be written. For output parameters there is a field for bucket key which enables to place the returned value in a specific bucket key.
Function	
Function name	The name of the function in the database.
Parameters	This is all the parameters to the function. The type of the parameter is the data type. The placement is eitther indexed or key based. If indexed, write the index of the parameter (starting at 1) in the field. If key based, write the key in the field. The input value is the value to send to the function for this parameter.
Result	The result of the function call may be placed in a special bucket key, specified in this field.
SQL	
SQL	Write yoru SQL call in the text area.
Output parameters	If your call will return values you may collect these and place them in different bucket keys. Select the result column (please note that the case may be of importance depending on the database) and which bucket key to place the returned value in.

Delete action

Handles
Email

Files
Folders

This action will delete an object. This action can handle both emails and files.



Delete action preferences.

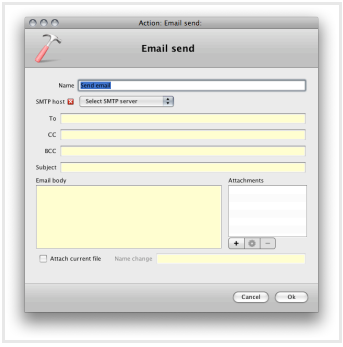
Settings for the delete action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.

Email send action

Handles
Email
Files
Folders

This action will send an email. Note that there is a special action for sending email based on another email (see the email [email] action). The object that caused this action to execute is not affected by this action.



Email action preferences.

Settings for the email action

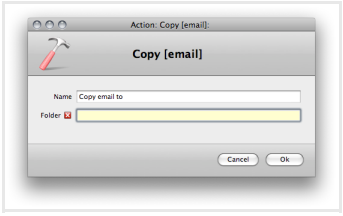
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
SMTP host	The SMTP settings to use when sending the email. If no proper SMTP host has been setup, you may create a new one from this popup.
To, CC and BCC	This is the list of receivers for the email. Comma separate different addresses
Subject	The email subject line.
Email body	The email body text. This may be HTML formatted if surrounded by *<html>* and *</html>*
Attach current file	If this checkbox is selected the file that caused the action to be executed will be attached to the email. The name change field can be used to set a certain name of the file when it is being attached to the email (note that the file anme on disc will not be changed).
Other attachments	This is a list of other static files to attach to the email. Note that the file location is a macro field and may therefor contain a path to a file with the same (or part of) file name as the file that caused the action to execute. For each attachment a name change may be applied. This name change will only apply to the attachment file and will not change any file name on disc.

Email Copy action

Handles
Email
Files

Folders

This action is only useful when the handled object is an email. This will make a copy of the email and put it in another email folder on the email server.



Copy [email] action preferences.

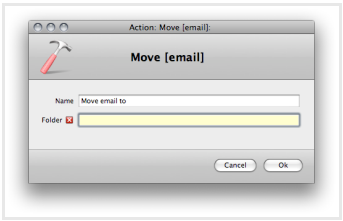
Settings for the copy [email] action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Folder	The name of the folder on the email server to put this email. Note that the folder must be accessible with the same account that is used in the email source from where the email that is handled is coming.

Email Move action

Handles
Email
Files
Folders

This action is only useful when the handled object is an email. This will move the email to another email folder on the email server.



Move [email] action preferences.

Settings for the move [email] action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Folder	The name of the folder on the email server to put this email. Note that the folder must be accessible with the same account that is used in the email source from where the email that is handled is coming.

Expand action

Handles
Email
Files
Folders

The expand action is an old action and should not be used unless for back compatibility reasons.

For both Macintosh and Windows the Stuffit application does not need to be started. FileTrain will automatically control the running state of the Stuffit application.

Also note that Stuffit Expander may show an error dialog when a file is wrongly handled in Stuffit. This dialog must manually be removed whenever appearing.

Stuffit Expander is required

This action requires that Stuffit Expander is installed on the same machine as FileTrain.

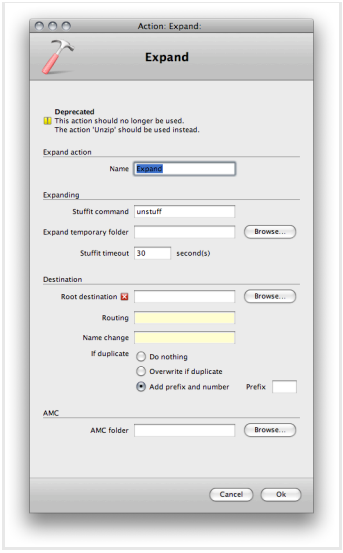
Macintosh vs Windows

This action works slightly different on Macintosh compared to Windows, it is therefore extra important to know the settings in the action.

See the individual settings in the table below for more information about the differences between Windows and Macintosh.

Important

On Macintosh the Stuffit application must be set to always create a surrounding folder when expanding.



Expand action preferences.

Settings for the expand action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Stuffit command	Only used on Windows. This is the command with which the Stuffit application is started. Older versions of Stuffit use unstuff and new versions may use other values. The command here is the command to start Stuffit through the console.
Expand temporary folder	On windows this is the folder where FileTrain will put the intermediate unstuffed files before further handlling. On Macintosh the Stuffit Expander application must be set to expand to a certain destination folder and this folder should be the same as the one entered here.
Stuffit timeout	This is the timeout in seconds to wait for Stuffit. If this timeout is reached and no result file is detected the action fails.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
If duplicate	With this setting you may select what will happen if there is a duplicate of the final destination location. There are three options; • Do nothing: This will cancel the action and leave the duplicate as it is. • Overwrite: This option will first delete the duplicate, then replace it with the new one from this action. • Add prefix and number: This option will leave the duplicate is it is. The new object from this action will be having an added number after the file name. If the prefix field is having a value this value will be added before the number. **Example with add prefix and number** If the prefix field has the value '-' (dash) and the final location file name was 'image.jpg' but this file already exists the new file will be named 'image-1.jpg'. The next file will be named 'image-2.jpg' and so on.
AMC folder	This is the folder where the expand action can find AMC files. If this folder is specified the action will try to find the corresponding AMC file for the file being expanded and update this AMC file with all the expanded file entries making the AMC file hold the information for these files as well as the containing archived file. AMC files are special XML formated files used by Laidback Solutions in other workflow applications. These AMC files contains meta information about other files.

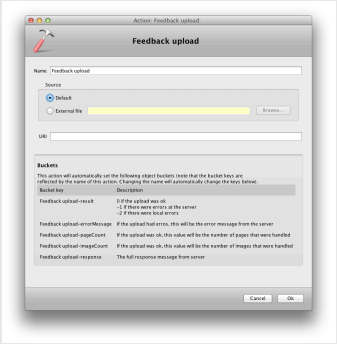
Feedback Upload

Handles
- Email
- Files
- Folders

This is a special action for uploading files to a Feedback Server.
The result from the upload will be saved in different bucket keys. Note that these keys are affected by the name of the action.

Bucket fields affected by the action

Bucket key	Value
<name of action>-result	0 if the upload was ok -1 if there were errors on the server side -2 if there were errors in FileTrain
<name of action>-errorMessage	The error message, if any
<name of action>-pageCount	The number of pages that were processed in this upload
<name of action>-imageCount	The number of images that were handled by the server
<name of action>-response	The full response from the server



FTP action preferences.

Settings for the FTP action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
URI	The URI for the upload site on the Feedback Server, e.g. http://127.0.0.1/feedback/processpdf/upload.php

For Each action

Handles
- Email
- Files
- Folders

The 'for each' action is not an action itself but will invoke other actions on a set of items. The action must have a list in form of a text string where a certain delimiter marks each item. For each item in the list a set of actions will be invoked.

Example with static value

We want to create five text files. Each text file should be named with todays date followed by a dash (-) and an index (1 to 5).

One way to accomplish this would be to create a for each action with the following setup.

Value: **1:2:3:4:5**

Delimiter: **:**

Bucket key: **myIndex**

Actions: A text action which have the following value in the file name field: **%D%-%BUCKET[myIndex]%.txt**

If you now change the value field in the example above to the following text: **help:content:todo:wanted** you will get four text files named
2013-02-04-help.txt
2013-02-04-content.txt
2013-02-04-todo.txt
2013-02-04-wanted.txt

If you create a separate text file which content is your list of values (e.g. the text file content may be **help:content:todo:wanted)** you can use this text file as source for the value in the for each action by using the macro **%FILE_CONTENT[...]%**

Example with value taken from database call

If we in the example above want to create text files based on a result from a database call we can use a database action combined with the macro **%BUCKET_LIST[...]%**.

Let's assume we have a database with a table named **DETAILS** and we have a column named **publicationDates**. We want to create text files with names taken from this column. We start with creating a database action where we get all the publicationDates (e.g. SELECT publicationDates from DETAILS). The database action has put the publicationDates in a bucket which we have named **pDate.** When the database action has multiple result FileTrain creates the additional bucket key **pDate-list** which is a list of all the values. This list is what we will use in our for each action.

We create the for each action with the following setup

Value: **%BUCKET_LIST[pDate-list,#]%**

Delimiter:**#**

Bucket key: **pDate**

Actions: A text action which have the following value in the file name field: **%BUCKET[pDate]%.txt**

'For each' action

Settings for the 'for each' action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Value	The value is the string that contains all the items which are used to loop the actions. Each item should be separated with the delimiter (see below). The value may be static text, a macro or a combination. There are some special macros which may be helpful setting the value.
Delimiter	The delimiter is the separator in the value string which separates each item.
Bucket key	Each item in the value string separated by the delimiter will cause the actions to be invoked. The current value of the item will be saved in this bucket key and therefor accessible for the actions.
Actions	This list contains all the actions that should be invoked on each item.

FTP send action

Handles
- Email
- Files
- Folders

The FTP action sends a file or folder to a certain FTP server using the *file transfer protocol*. This action will put some values in the station bucket which may be used by other actions. If a folder is being handled the folder itself might be omitted, i.e. the folder itself is not being sent to the FTP server.

The object that caused this action to execute is not affected by this action.

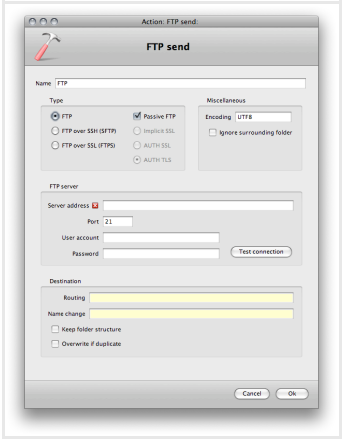
Important note about macro %DUPLICATE_FILE_EXTRA%

Please note that the macro %DUPLICATE_FILE_EXTRA% can not currently be used in this action setup.

Bucket fields affected by the FTP action

Bucket key	Value	Comment
FTPSendOkList-*<name of action>*	List of the files that has been sent to the remote server. May contain both files and folders.	The <name of action> is the name of the FTP action that is given in the preferences. If you for example has a FTP action that is named 'send images' the file list will be found in the bucket with key 'FTPSendOkList-send images'.
FTPSendErrorList-*<name of action>*	List of files that should have been sent but	See above for information about <name of action>.

	because of errors has not been able to be sent. May contain both files and folders.	



FTP action preferences.

Settings for the FTP action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Type	Select the type of transfer and individual setup for each type.
Encoding	If the FTP server is using a special character encoding you may specify that here. Default, UTF8 is used.
Ignore surrounding folder	If selected and the handling object is a folder, that folder itself is not sent to the FTP server, however all the content in the folder is sent.
Server address	This is the address to the FTP server to send the object to.
Port	The port on the FTP server to send to. Default FTP port is 21.
User account	The user login to the FTP server. If the server does not require any user name, use the name *anonymous*.
Password	The password for the user account. Leave blank if no password is required.
Routing	If the object should be put in any other directory than the default login folder that sub folder path should be entered here.
Name change	Enter a new name if the name of the object should change on the final destination.
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Image processing [ImageMagick] action

Handles
- Email
- Files
- Folders

This action is working with image files supported by ImageMagick.

This action may resize images in various ways and also change the format of a image from jpeg to png for example. More advanced color setup and handling of ICC profiles or adding text to an image may also be done.

Image format change

If the source file is in one format but the result file is wanted in another it is as simple as setting the name change to a proper name with the wanted file suffix.

If for example the source file is a jpeg but the result is wanted in png format, just enter **%F%.png** in the name change field and the result file will in fact be a png with the same name as the original file.

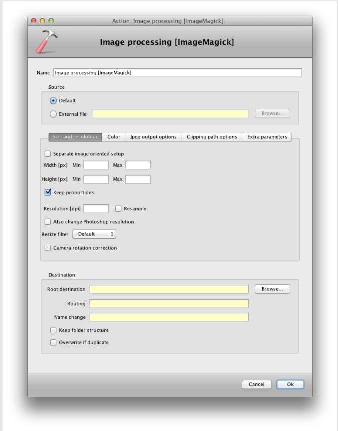


Image processing [ImageMagick] action preferences.

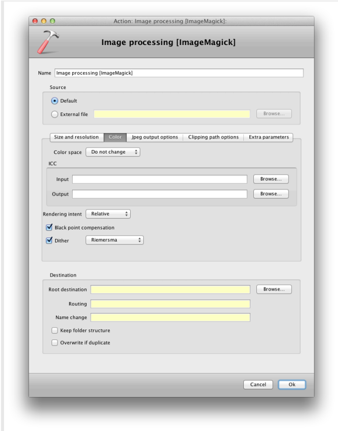


Image processing [ImageMagick] action preferences.

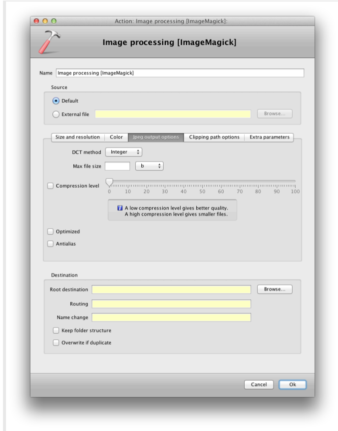


Image processing [ImageMagick] action preferences.

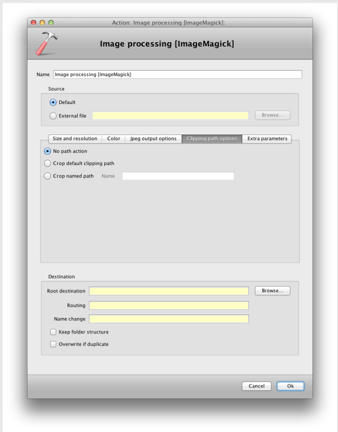


Image processing [ImageMagick] action preferences.

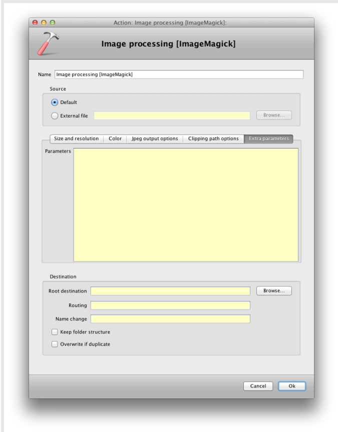


Image processing [ImageMagick] action preferences.

Settings for the Image processing [ImageMagick] action

Setting	Description
Size and resolution This panel shows setup for resizing options.	
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Separate image oriented setup	If this option is selected the resize options are divided into portrait and landscape options which may be setup differently. The different setups regards the source image, if the image is in landscape format (width>height) the landscape settings are applied. Note that the images may be having a rotation which is not applied before calculating the image format.
Width	The min and max may be left empty or have a value (specified in pixels). If the min value is set the result image will at least have this width. If the max value is set the result image will not exceed this value.
Height	This is similar to the width setting but for the height.
Keep proportions	If this selection is checked the proportions of the source image will be kept.
Resolution	The resolution in the result image (specified in dots per inch, DPI). This value may be left out and the resolution will not be changed. If the resample checkbox is selected the image will be resampled with this new resolution.
Also change Photoshop resolution	The Photoshop application are reading different resolutino flags than those changed in the regular resolution. To get Photoshop to show the new resolution this checkbox must be selected. This alsso require Exiftool to be installed. If this choice is selected and no Exiftool is found by FileTrain the user will be asked to install the bundled version of Exiftool that comes with FileTrain.
Resize filter	This is the method to use when resizing the image.
Color space This panel contains setup to change the color space or to insert a specific ICC profiel.	
Color space	Select the output color space if you like to change it.

ICC	If a file is selected this file will be inserted in the image and used as the ICC profile.
Jpeg output options If the output format is jpg some extra setup may be specified in this panel.	
DCT method	This is the method to use as the discrete cosine transformation when the output is a jepg image.
Max file size	You may specify a max file size for the output. Note that this opion is not always that accurate.
Compression level	The level of compression in the output file. A low compression gives a better quality but also a bigger file size. If the checkbox is not selected the compression level will not change.
Optimized	Select this box to make the jpeg output optimized.
Antialias	Select this box to make the jepg output antialiased.
Clipping path options This panel contains setup to crop the image speicied by a clipping path.	
No path action	This is the default option and means that no handling of clipping paths will occur.
Crop default clipping path	Select this option if you want the result image to be cropped (rectangularly) according to the default clipping path.
Crop named path	Select this option if you want the result image to be cropped (rectangularly) according to a specificly named clipping path.
Extra parameters In this area you specify any extra parameter that you would like to send to ImageMagick.	
Parameters	Enter the parameters as they would be in a command line interface.
Destination In this area you specify where the result image should be written. Note that if you use a name change with a new file suffix the actually file format is changed.	
Root destination	This is the root folder where the action will put the result image. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Image processing [SoftColor] action

Handles
- Email
- Files
- Folders

This action is working with image files to produce a new image file based on the setup seen below. This action is only available on Windows since SoftColor is only available for the Windows platform.

Note that SoftColor is a third party application which must be purchased separately from FileTrain.

For information about the individual settings, please see the SoftColor documentation found online at http://www.softcolor.fi/.

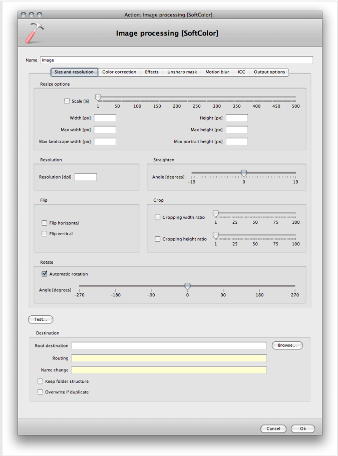


Image processing [SoftColor] action preferences.

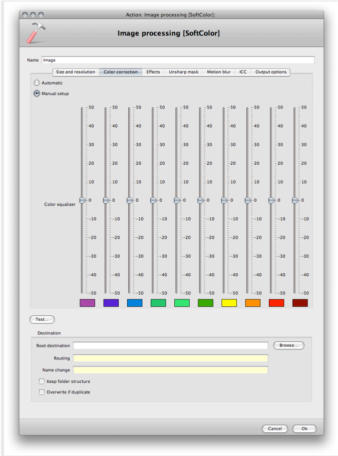


Image processing [SoftColor] action preferences.

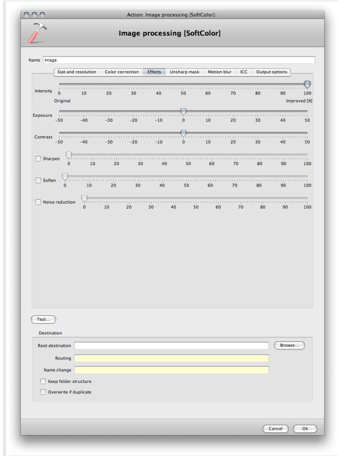


Image processing [SoftColor] action preferences.

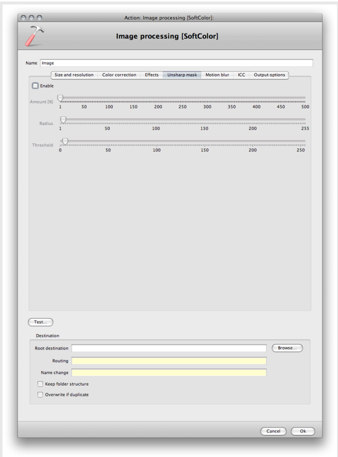


Image processing [SoftColor] action preferences.

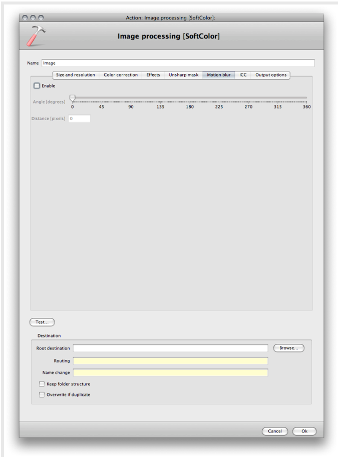


Image processing [SoftColor] action preferences.

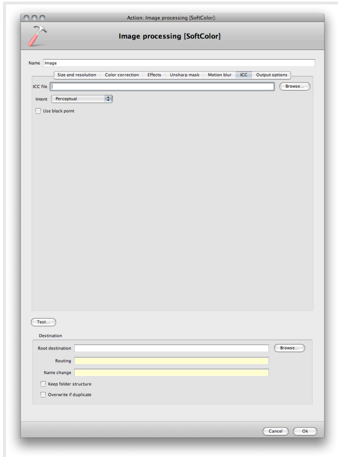


Image processing [SoftColor] action preferences.

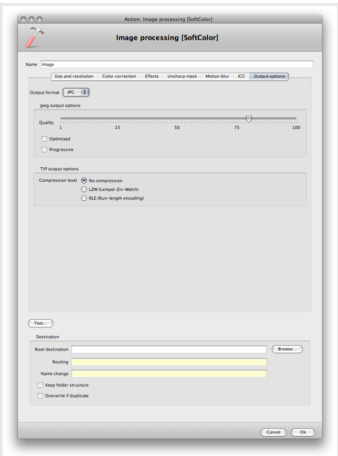


Image processing [SoftColor] action preferences.

Settings for the Image processing [SoftColor] action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Size and resolution	See the SoftColor documentation for detailed information.
Color correction	See the SoftColor documentation for detailed information.
Effects	See the SoftColor documentation for detailed information.
Unsharp mask	See the SoftColor documentation for detailed information.
Motion blur	See the SoftColor documentation for detailed information.
ICC	See the SoftColor documentation for detailed information.

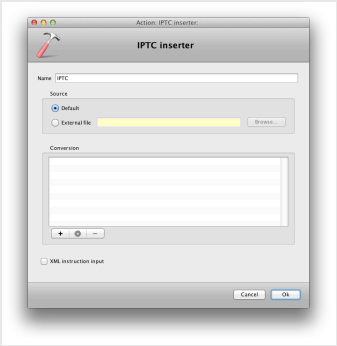
Output options	See the SoftColor documentation for detailed information.
Destination	
Setting	**Description**
Root destination	This is the root folder where the action will put the result image. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

IPTC inserter action

Handles
- Email
- Files
- Folders

The IPTC action is only guaranteed to work on JPG and TIFF files. This action will insert new IPTC information into the files with the possibillity to keep the old values.

To handle the insertion click the add button under the conversion list. This will open a dialog where the IPTC field can be choosen and the insertion text added. There is an option in this dialog to *replace*, *append* or *precede* the old value with the new value.



IPTC inserter action preferences.

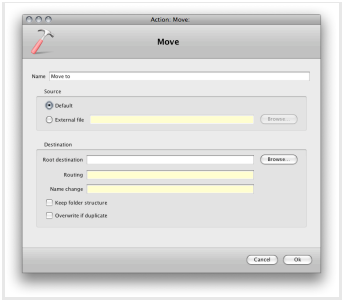
Settings for the IPTC inserter action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Conversion	This list contains all the insertions that should be done. To add a new insertion, click the add button under the list. To remove an existing insertion, select it in the list and click the delete button. To edit an existing insertion, select it and click the edit button.

Move action

Handles
- Email
- Files
- Folders

This action will move the object to a new location.



Move action preferences.

Settings for the move action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

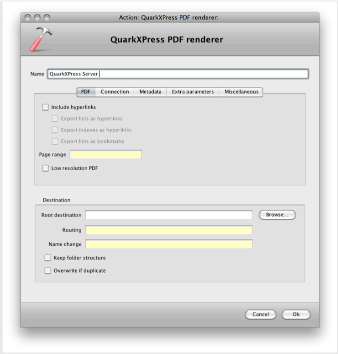
QuarkXPress PDF renderer action

Handles
- Email
- Files
- Folders

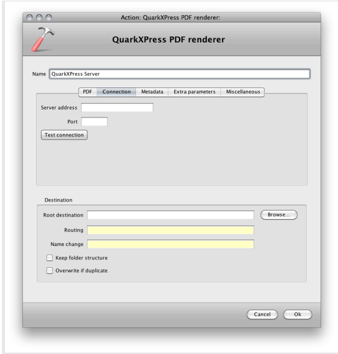
This action will render a PDF from a QuarkXPress document. The result PDF is placed in selected location. This action will connect to a QuarkXPress Server. When connected the current Quark document in scope will be uploaded to the QXPS document pool, rendered as a PDF and finally deleted from the document pool.

Bucket fields affected by the QuarkXPress action

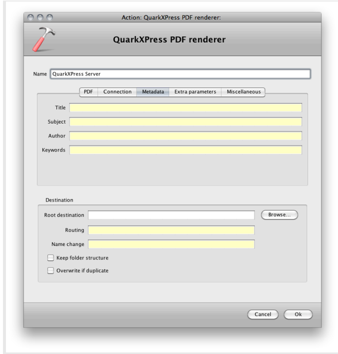
Bucket key	Value	Comment
<name of action>-linkedFiles	List of the files that has been found as linked files. The list is separated with ', '	The <name of action> is the name of the FTP action that is given in the preferences. If you for example has an action that is named 'convert' the file list will be found in the bucket with key 'convert-linkedFiles'.
<name of action>-linkedMissingFiles	List of the files that has been detected as linked files and are not found. The list is separated with ', '	See above for information about <name of action>.
<name of action>-linkedFoundFiles	List of the files that has been detected as linked files and are found. The list is separated with ', '	See above for information about <name of action>.



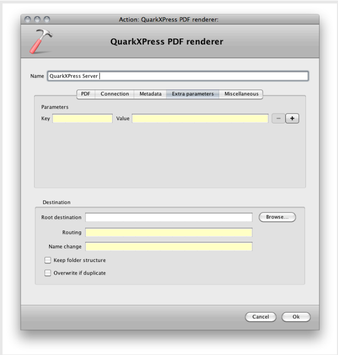
QuarkXPress PDF render action preferences.



QuarkXPress PDF render action preferences.



QuarkXPress PDF render action preferences.



QuarkXPress PDF render action preferences.

QuarkXPress PDF render action preferences.

Settings for the QuarkXPress PDF render action

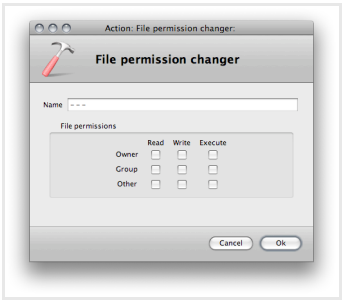
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
PDF - include hyperlinks	Specifies whether hyperlinks should be included in the PDF document.
PDF - export lists as hyperlinks	Specifies whether lists should be exported as hyperlinks.
PDF - export indexes as hyperlinks	Specifies whether the index should be exported as hyperlinks.
PDF - export lists as bookmarks	Specifies whether lists should be exported as bookmarks.
PDF - page range	If not all the pages should be rendered you may enter the page range, e.g. 1-3.
PDF - low resolution PDF	If this checkbox is selected the rendered PDF will be low resolution.
Connection - server address	The server address where the QuarkXPress Server is located.
Connection - port	The port on which the QuarkXPress Server is listening.
Metadata - title	Additional title information in the PDF.
Metadata - subject	Additional subject information in the PDF.
Metadata - author	Additional author information in the PDF.
Metadata - keywords	Additional keywords information in the PDF.
Extra parameters	In this tab you may enter any additional parameter key-value pair to the QuarkXPress Server (see the QuarkXPress Server web integration documentation for more information about parameters available).
Miscellaneous - Timeout	If this option is selected and the time is set, this will be used as timeout before action fails.
Miscellaneous - Check if linked files are missing	If this option is selected, FileTrain will do an internal search for the linked files. If selected, please note that the files will be searched for from the FileTrain server.
Root destination	This is the root folder where the action will put the final PDF file. If the routing field (see below) is empty the PDF will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination.

	Note The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the PDF will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%.pdf' in the name change field (see the section about macros for more information).
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

File permission action

Handles
- Email
- Files
- Folders

This action will try to change the file permission rights the object. This action is only available on Macintosh.



File permission action preferences.

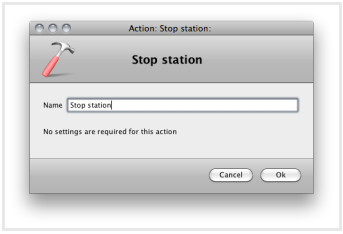
Settings for the file persmission action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
File permissions	If a checkbox is selected it means that the rights for that particular user (either owner, group or other) for the particular action (read, write or execute) will be set.

Stop station action

Handles
- Email
- Files
- Folders

This action will immediately stop the station.



Stop station action preferences.

Settings for the stop station action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.

Storage action

Handles
- Email

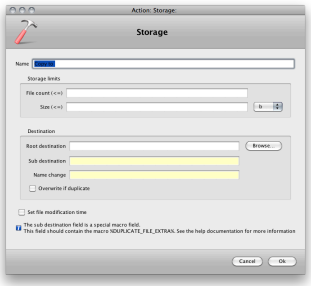
Files
Folders

This action is a modified version of the copy action. The action will copy the object to a new location but will in this location create folders to put the object in. When a folder is 'full' the next object will be placed in another folder with a new index. This means that the folder to where the object should be moved can be limited in size or number of items. This action is very handy if you for example collect your files into archives with certain sizes.

Example - create CD archives:

You want to create an archive but each folder in the archive must not exceed the size of a CD-ROM (700 MB). We select a root destination and then enter a name in the sub destination. In this example we will use 'CD %DUPLICATE_FILE_EXTRA[%FREE_INDEX%]%'. We set the size limit to 700 MB.

When the first object is handled it will end up in the folder 'CD' (sub folder to the root folder). All the coming objects will end up in this until an object that can't fit in this folder is handled. When this object is handled a new folder 'CD 1' will be created and the object will end up in this.
If a new object will fit in the first folder it will end up there even if a latter folder has been created. This will make sure that the folders are filled up to a maximum.



Storage action preferences.

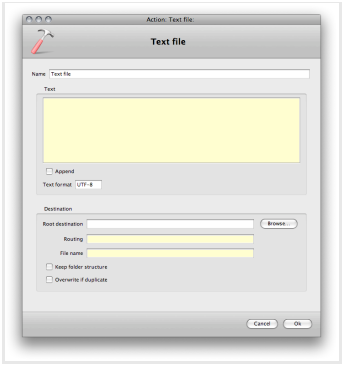
Settings for the storage action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Storage limits	This is the limits that the destination folders will be set to have. You may enter a count limit meaning that the number of items in the destination folder can not exceed that number. You may also set the size limit, meaning the total file size in the destination folder. These limits may be set individually or both at the same time. If both limits are set a new destination folder is created when any of the limits are reached.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Sub destination	This is the path to the sub location. This field is a macro field and may contain macros. If you for example enter '%D%' in this field the final destination folder will be a sub folder with the current date in the root destination. This field must contain the macro %DUPLICATE_FILE_EXTRA[...]% in which the macro %FREE_INDEX% us useful which will be replaced by a running number for the storage location. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	If this field is having a value the object will get this name at the final location. This can for example be used to add a special prefix on the object. If we need to add the prefix 'A-' on the object we will put 'A-%F%%E%' in the name change field (see the section about macros for more information).
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.
Set file modification time	If this option is selected the copied file will get the current time as modification time.

Text file action

Handles
Email
Files
Folders

This action will create a text file in any format. This action will not affect the object that caused this action to execute.



Text file action preferences.

Settings for the text file action

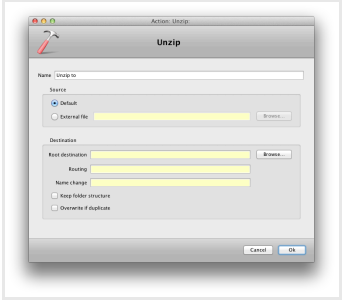
Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Text	The text that will be created. This text may contain macros and the text can be in any format.
Append	If this option is selected the original file content will be kept and the text will be appended to the current file content.
Text format	The format which should be used when writing the file. The default value is **UTF-8**. The value generally follows the conventions documented in RFC 2278: IANA Charset Registration Procedures. The value is not case-sensitive.
Root destination	This is the root folder where the action will put the object. If the routing field (see below) is empty the object will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
File name	The file name of the created text file. If this name is missing the name will be taken from the object being handled but with the file extension *.txt*.
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Unzip action

Handles
- Email
- Files
- Folders

This action will unzip a file and put the result files in a final destination.

This action will not affect the source zip file so if the source file is not needed after the unzip a delete action should be set after this action.



Unzip action preferences.

Settings for the unzip action

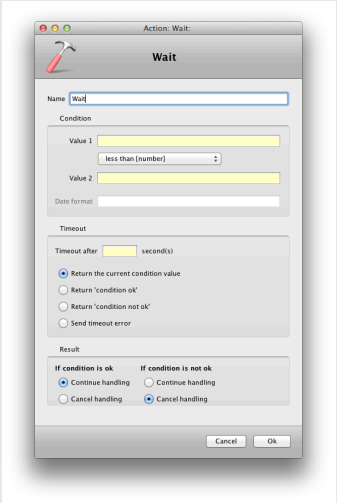
Setting	Description

Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Root destination	This is the root folder where the action will put the unzipped files. If the routing field (see below) is empty the files will be placed in the root destination, otherwise in a sub location described by the routing field. The location entered in this field must exist, see note for the routing field.
Routing	Use this field to enter a sub location to the root destination. This field is a macro field and may contain macros. If you for example enter '%D%' in the routing field the final destination folder will be a sub folder with the current date in the root destination. **Note** The root destination must exist for the action to be executed correctly. The final destination folder created with the routing does not need to exists, it will be created if it is not present.
Name change	The name change regards the unzipped files. See the special macro %ZIPF% for information how to use the original zip file's name for the result files. This field may be empty in which case the files will keep their original names.
Keep folder structure	The folder structure is the sub folder structure seen from the source main folder where it applies. Example: A folder source is set to watch folder A and also selected to search subfolders. The pdf file file.pdf is detected in the folder A/B/file.pdf. If the keep folder structure is selected the result file will be placed in the sub folder B also in the end location.
Overwrite if duplicate	This option will first delete the duplicate, then replace it with the new one from this action. See information of the special macro %DUPLICATE_FILE_EXTRA[...]% for more options regarding duplicate files.

Wait action

Handles
- Email
- Files
- Folders

This action will wait and hold the current process until a certain condition is fulfilled.
To avoid that the process is stalled forever the setup has a timeout setting.



Wait action preferences.

Settings for the wait action

Setting	**Description**
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Condition This is the setup for which the action will wait until it is fulfilled. An alternative timeout may be set to avoid an eternal wait.	
Value 1 and Value 2	The values to be calculated and compared. The popup between the values will set how the values will be compared. Example) If value 1 is set to "1" and value 2 is set to "2" and the comparison is set to "equals" this will mean that the condition is fulfilled and the wait will return immediately.
Date format	If the comparison popup has been set to compare two dates, this field will be enabled and you may set the date format for the values that are to be compared
Timeout	

If a timeout is set, and the condition is not fulfilled until the timeout is reached this section will decide what will happen.

Timeout after	This is the timeout in seconds.
Return the current condition value	If this option is selected when the timeout is reached a final check is made of the condition and if the condition is true, the result will be handled as 'ok'
Return 'condition ok'	If this option is selected the timeout will mean that the result will be forced to 'ok'
Return 'condition not ok'	If this option is selected the timeout will mean that the result will be forced to 'not ok'
Send timeout error	This will cause the action to send an error (remaining actions will not be performed). Additional setup for the error may be done in the station's error setup.

Result

This is what will happen when this action is done performing (unless the send timeout error is selected and timeout is happening).

The 'If condition is ok' is what will happen if the condition is fulfilled (either that the condition is actually ok or timeout setting is set to force an ok). The 'If condition is not ok' will be executed if the condition was not ok or timeout was forcing a 'not ok'.

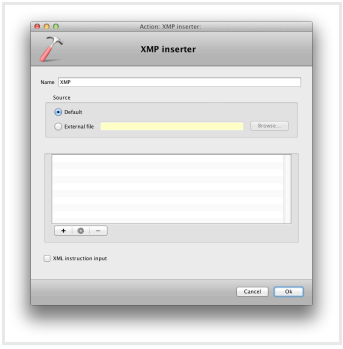
Continue handling will mean that the station will continue the workflow as usual. Cancel handling will force the station to cancel the remaining actions.

XMP inserter action

Handles
- Email
- Files
- Folders

The XMP action is only tested to work on JPG, TIFF and PDF files. This action will insert new XMP information into the files with the possibillity to keep the old values.

To handle the insertion click the add button under the list. This will open a dialog where the XMP field can be choosen and the insertion text added. There is an option in this dialog to *replace*, *append* or *precede* the old value with the new value.



XMP inserter action preferences.

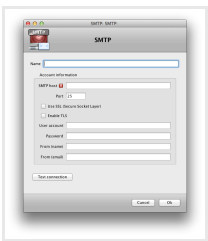
Settings for the XMP inserter action

Setting	Description
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
Source - default	The default source indicates that the action should be invoked on the object that is currently in scope. This is the normal way to use the action.
Source - external file	This option is useful if you would like the action to be applied on another file than the one currently in scope. This is convenient for example if you have a sidecar file that needs to be handled or if the file that should be handled is referenced in a text file.
Inserter list	This list contains all the insertions that should be done. To add a new insertion, click the add button under the list. To remove an existing insertion, select it in the list and click the delete button. To edit an existing insertion, select it and click the edit button.

3.6 SMTP servers

In some components in FileTrain there are settings involving SMTP servers. These settings will have a popup with all the available SMTP hosts created and the user may also create a new SMTP host from those popups. All the SMTP host settings are collected in the SMTP preferences area.

When testing the connection you may see a warning that a certain certificate is used (if using SSL), you may select to enable this certificate or not.



SMTP preferences

Settings for a SMTP host

Setting	Description

Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
SMTP host	The SMTP server name as IP number or DNS name, e.g. 192.168.40.3 or smtp.yourserver.com.
Port	The SMTP server port. The default SMTP port is 25.
Use SSL	If this check box is selected FileTrain will use the secure socket layer whilst communicating with the SMTP server.
Enable TLS	Enabling TLS communication with the email server
User account	The user login for the SMTP server.
Password	The password for the user account above.
From (name)	The personal name for the 'from' field in the email.
From (email)	The email address which will be the address from which the email is seen sent from.

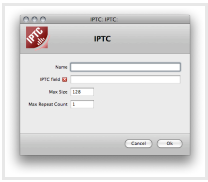
3.7 Customized IPTC

The image handling in FileTrain includes IPTC reading and writing. FileTrain comes with a default set of IPTC fields, the standard fields. Additional to these standard IPTC fields you may add custom IPTC fields in FileTrain to read and write to non-standard IPTC values.

In various settings where IPTC fields are required there is a popup menu with all the default and the custom IPTC fields. From this popup it is also possible to create new custom IPTC fields. In the customized IPTC part of the configuration you may view the custom IPTC fields. Here you can add, remove and edit the custom IPTC fields. All the custom IPTC fields which are created with the special IPTC popup will appear in this area.

Custom IPTC

Make sure that all custom IPTC fields that are added in the FileTrain setup have unique IPTC field numbers. The numbers used in the custom IPTC must also not collide with any standard IPTC field number.



IPTC preferences.

Settings for a custom IPTC field

Setting	**Description**
Name	The name for this component. This name is used throughout the rest of the settings in FileTrain and is acting as an identifier for the component.
IPTC field	The IPTC field number. This number must be a unique number and not be the same as another custom IPTC field or any of the standard IPTC fields.
Max size	The maximum number of bytes that this IPTC field should be. Some IPTC fields may only contain 1 byte while others may contain several kB of text.
Max repeat count	This is the maximum number of repeats that this field may occur in the IPTC data. Some IPTC fields may occur only one time while others may have several entries.

3.8 Customized XMP

FileTrain has a default set of XMP fields which may be used to get values from or set the values to. There are many more XMP fields available since each company may define their own XMP fields. In FileTrain you may add one or more customized XMP fields to comply with such fields that are not in the default list of FileTrain.

XMP preferences.

When you add or edit a customized XMP field you will get the setup dialog for that XMP field, as seen below.

XMP field setup.

To setup the XMP field you will need to enter all the details.

Settings for a custom IPTC field

Setting	Description
Name	The name for this XMP field. This name is only a name to show in the customized XMP field list and has no technical purpose.
Namespace	
URI	The namespace URI is the specific URI to identify the XMP group. See the documentation from the supplier for the URI details. Example) For the TIFF group this URI is **http://ns.adobe.com/tiff/1.0/**
Prefix	The namespace URI is not used when writing the XMP data, instead a prefix is used for this particular namespace, see the documentation from the supplier about the certain prefix used. Example) For TIFF XMP the prefix is **tiff**
Human name	This is used in the setup throughout FileTrain where this particular XMP field is shown.
Property	
Name	The name of the property in the XML. See the documentation from the supplier for the details. Example) For the TIFF XMP the compression has the property name **Compression**.
Human name	This is used in the setup throughout FileTrain where this particular XMP field is shown.
Value	
Type	Select the type of the property, see the documentation from the suppolier for details. Example) For the TIFF XMP the compression has the type Integer.
Limit	If the property has a certain value limit, this may be set here. use comma separated list to set a list of values. Use a comma to set a span of values. Example) For the TIFF XMP the compression may have a integer value ranging from 1 to 6. The limit would therefor be **1:6**.

3.9 External Tools

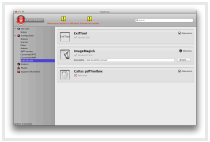
Depending on the setup of your FileTrain there may be external tools needed. If you have a setup which uses external tools you will be prompted a warning in the FileTrain main window if that tool can not be found by FileTrain. This warning will also contain (as a tooltip) where in your setup the tool is used.

FileTrain tries to locate all external tools automatically but in some cases you may have to do some manual setup in order to locate the correct tool.

Each external tool may have individual setup which you can see by clicking on the button 'Show more' on the right hand side of each external tool.

If the setup is correct and FileTrain is able to locate the tool you will see the version of that tool.

Note that the external tools are not included or shipped with FileTrain, they must be downloaded/purchased from external source.



The External Tools setup

Settings for external tool - Exiftool

Setting	Description
Executable	Here you select the executable file for Exiftool

Settings for external tool - ImageMagick

Setting	Description
Executable	Here you select the executable file for ImageMagick. **NB!** On Macintosh OSX you may have to set the executable manually and the path is then most likely the following **/opt/local/bin/convert**

Settings for external tool - Callas pdfToolbox

Setting	Description

Executable	Here you select the executable file for Callas pdfToolbox.
Callas profiles folder	The folder where the profiles for Callas pdfToolbox are located.
Temporary folder	The folder to use for temporary repository

3.10 XMP

In FileTrain there are some standard XMP fields added. The following namespaces are currently supported except for those properties stated below. See the section about customized XMP to add your own XMP fields.

XMP Basic (http://ns.adobe.com/xap/1.0/)

The property thumbnails is not yet supported.

Dublin Core (http://purl.org/dc/elements/1.1/)

EXIF (http://ns.adobe.com/exif/1.0/)

The following propertis are not yet supported: CFAPattern, ComponentsConfiguration, CompressedBitsPerPixel, DeviceSettingDescription, ExifVersion, Flash, FlashEnergy, FlashpixVersion, GPSDestBearing, GPSDestBearingRef, GPSDestDistanceRef, GPSDestLatitude, GPSDestLongitude, GPSImgDirection, GPSImgDirectionRef, GPSLatitude (only as string), GPSLongitude (only as string), GPSMeasureMode, GPSSpeedRef, GPSStatus, GPSTimeStamp, GPSTrack, GPSTrackRef, GPSVersionID, ISOSpeedRatings, OECF, SpatialFrequencyResponse, SubjectArea, SubjectDistance, SubjectDistanceRange, SubjectLocation,

Extended EXIF (http://ns.adobe.com/exif/1.0/aux/)

IPTC (http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/)

PDF (http://ns.adobe.com/pdf/1.3/)

Photoshop (http://ns.adobe.com/photoshop/1.0/)

The following propertis are not yet supported: DocumentAncestors and TextLayers

RAW (http://ns.adobe.com/camera-raw-settings/1.0/)

The following propertis are not yet supported: ToneCurve, ToneCurveName, WhiteBalance

Rights (http://ns.adobe.com/xap/1.0/rights/)

TIFF (http://ns.adobe.com/tiff/1.0/)

The following propertis are not yet supported: BitsPerSample, PrimaryChromaticities, ReferenceBlackWhite, TransferFunction, WhitePoint, YCbCrCoefficients, YCbCrPositioning, YCbCrSubSampling

4. Workflow examples and tips

4.1 Simple ad workflow

In this example we will make FileTrain watch a folder. PDF files will be moved to a PDF preflight workflow, InDesign and Quark documents will be moved to a folder for manual review, images will be moved to a FTP server for further handling and all the other files will be moved to a folder for manual handling.

The components we need to do this are;

- Source: A folder source that will watch the folder.
- Filter: A file filter to recognize PDF files.
- Filter: A file filter to recognize InDesign and Quark documents (note! one filter for both).
- Filter: A file filter for the image types that we want to transfer to FTP server.
- Filter: A file filter to catch all the files that are not recognized by the filters above.
- Action: A move action for the PDF files.
- Action: A move action for the InDesign and Quark documents.
- Action: A FTP action to send images to FTP server.
- Action: A delete action which will delete the local image when it has been uploaded to the FTP (note! if we don't delete the file it will be handled by the 'all files' filter and will be moved to the manual handling folder).
- Action: A move action for all the rest of files.

Sources

The only source we need is a folder source. In our example the folder */workflow/in* will be scanned. All detailed settings for this source can be viewed in image 4.1-1.

Image 4.1-1 Source settings for incoming files.

Filters

In this example we need four different filters. The last filter is to catch all the remaining files which has not been caught in the other three filters. We will be able to use two of the default filters in FileTrain, the one to catch PDF files (**** PDF document*) and the last filter which will catch all the remaining files (**** All files*). The images below shows the settings for the four filters.

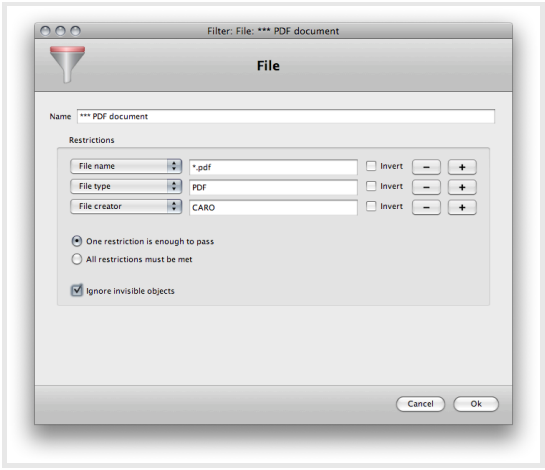


Image 4.1-2 Filter settings for PDF documents.

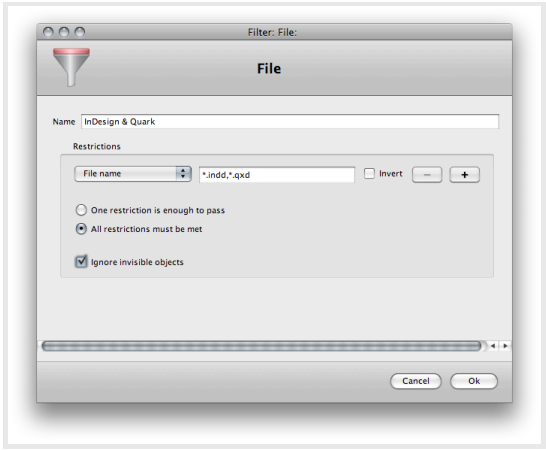


Image 4.1-3 Filter settings for InDesign and Quark documents.

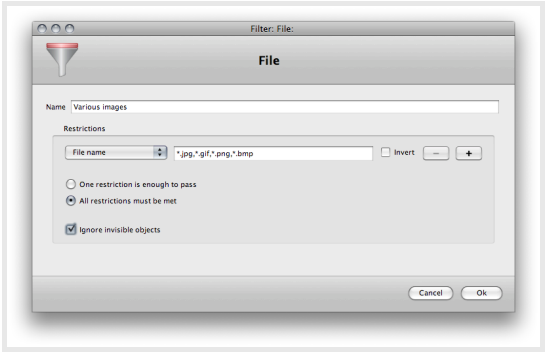


Image 4.1-4 Filter settings for images.

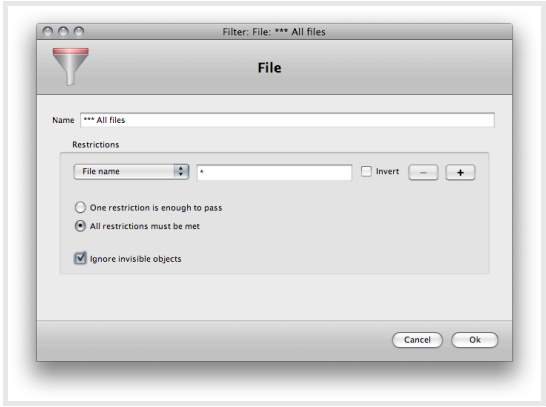


Image 4.1-5 Filter settings for 'all files'.

Actions

In this example we will need five actions, four obvious ones and one delete action after the FTP upload action since we don't want to save the images on the local drive after the upload. The settings for the actions are seen in the images below.

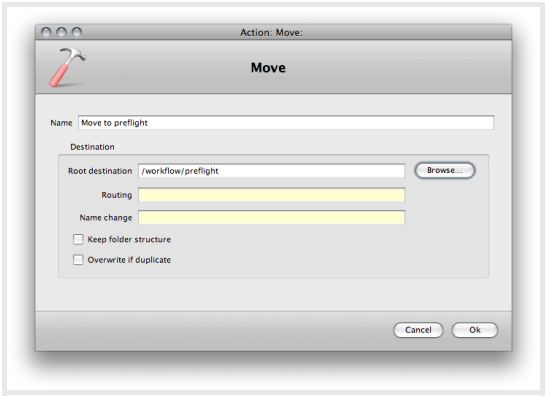


Image 4.1-6 Action settings for the move of PDF documents.

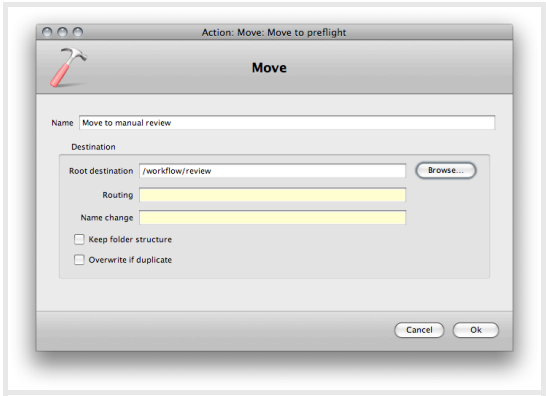


Image 4.1-7 Action settings for moving InDesign and Quark documents.

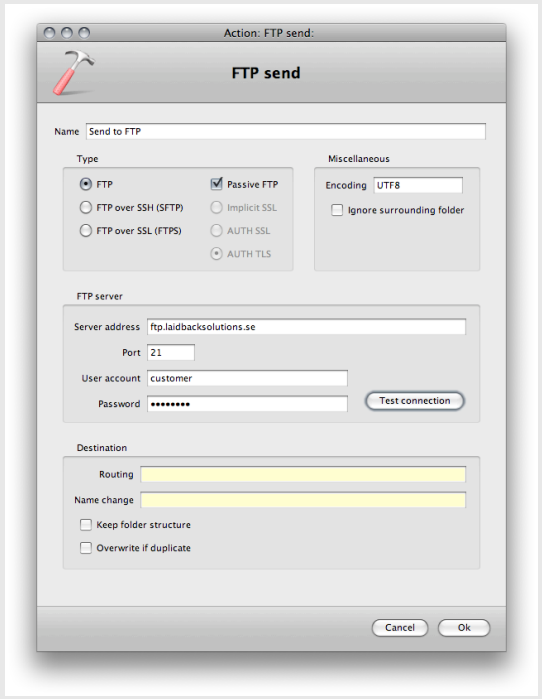


Image 4.1-8 Action settings for FTP sending.

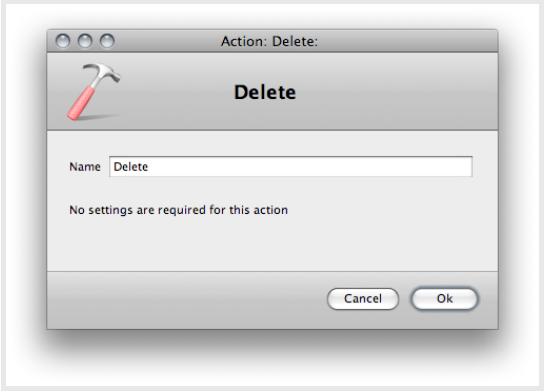


Image 4.1-9 Action settings for deletion.

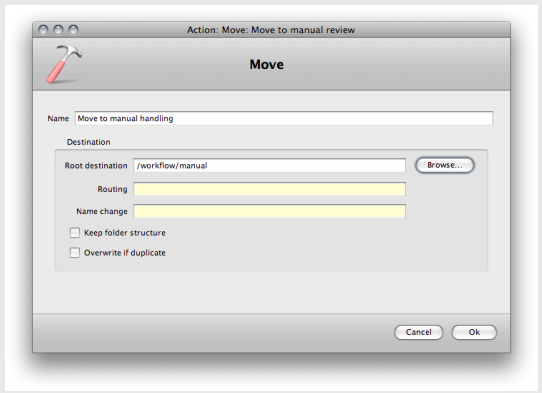


Image 4.1-10 Action settings for moving remaining files.

Station

When all the components above are created they are put together in the station, see the images below for information about the settings.

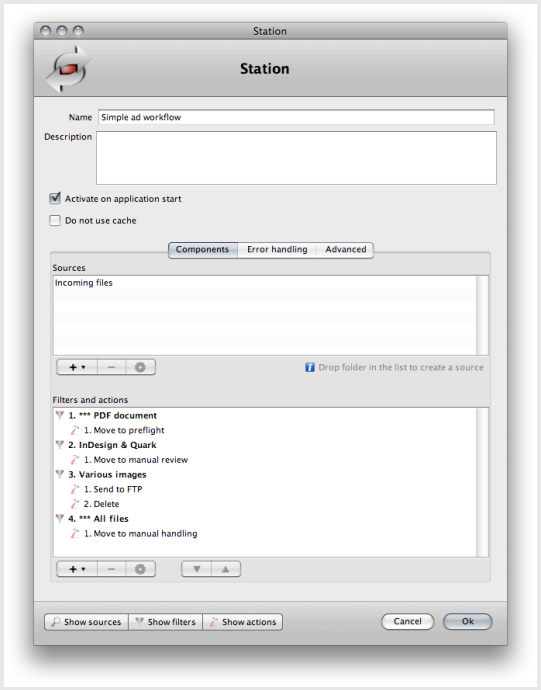


Image 4.1-11 Station component settings.

4.2 Getting selected information from XML file

In various situations you may want to get some portions of a XML file, e.g. inserting some IPTC information in image files where the information is from a certain tag in a XML file.
Such extraction of data is possible with the help of some macros in FileTrain.

From FileTrain version 6.4 it is possible to get XML content using XPath expressions. This method is the preferred and to use this, see the macros XPATH and XPATHLIST in the macro session for more details.

Before FileTrain version 6.4 you could use the following technique to get hold of the XML but this was only useful in more simple XML files. You should use the macro XPATH to get information from a XML.

To be able to get data from a XML file you will need to calculate the location and name of the xml file. This may be a static location or some location calculated from the file that is currently processing in FileTrain. In this example we assume that the xml file is located in the same folder as the file that is being handled (e.g. if an image file and xml file both are located in the same folder and the image file is handled).

This extraction is only possible if the XML structure is fairly simple. If the tag name is present several times this may be a problem but if the tag is only present once this example will work fine.

We assume that the information we look for is located between the start tag **<name>** and the end tag **</name>**. To get this information we will use the following macro text:

%FILE_CONTENT[%FILE_PARENT_PATH%/%F%.xml,UTF-8,%MATH[+,6,%INDEX_OF[%FILE_CONTENT[%FILE_PARENT_PATH%/%F%.xml,UTF-8]%,<name>,0]%]%,%INDEX_OF[%FILE_CONTENT[%FILE_PARENT_PATH%/%F%.xml,UTF-8]%,</name>,0]%]%

In the macro above there are three locations (red text above) that are important depending on the tag start and tag end. Two of these locations contain the start tag and the end tag. The third location (the number 6 above) is the number of characters found in the start tag (<name> contains six characters).

In this macro the path to the xml file is found in many places, the path is **%FILE_PARENT_PATH%/%F%.xml** which is the same path as the currently handled file with the xml extension.

5. Macros

A macro is a holder for a dynamic text. If you for example need to have the current date in your text you can either change your text every day or you can use a date macro which will replace the date macro with the current date. This behaviour makes the macro extremely powerful in tools which requires dynamic text.

A macro text may contain both macro tags and regular text. It may contain zero, one or more macro tags.

A macro field looks similar to a regular text field. The only difference is that the macro field has a yellow background while the regular text fields has a white background.

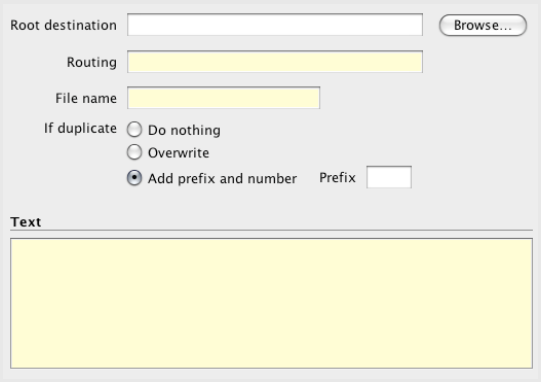


Image 13-1 Macro fields.

5.1 Syntax

A macro tag starts and ends with a percent sign (%). The macro name is found immediately after the first percent sign.

> **Example of a simple macro**
>
> The macro for the file path for a certain file is named 'FILE_PATH'. The macro is written as **%FILE_PATH%**

The example above contains a 'simple' macro. The simple macros only have a name. There are however 'complex' macros which need a little extra information to be handled correctly. Many macros have both a simple version and at least one complex version.

The complex macros are written as the simple macro with an extra part where the parameters are written. The parameters are written immediately after the macro name within straight brackets ([and]).

> **Example of a complex macro**
>
> The macro for the current date is named 'D'.
> This macro has the simple form **%D%**
>
> There is an extended alternative to the date macro where the format of the current date may be entered (read more about the formats in the macro specification), e.g. **%D[yyyy-MM]%**

The complex macros may have one or more parameters. In the example above the date macro is using one parameter in the complex form to indicate what format of the date that should be the result. If more than one parameter is used, each parameter is separated with a comma (,).

Macro with more than one parameter

The date macro has a version where the result date may be the current date plus or minus a certain number of days. If we want to add one day and want the result to be on the format yyyy-MM-dd the macro would be:

%D[+1,yyyy-MM-dd]%

5.2 Special macro characters

A macro is built with four special characters;

1. Percent sign: %
2. Left bracket: [
3. Right bracket:]
4. Comma character: ,

The parameters in a macro may be built with other macros which in their turn may contain more macros etcetera. A macro's parameters may not contain any of the four special characters except if a new macro is specified. None of the four special characters may be used directly as simple text in a macro, instead the following replacements can be used to get the characters as output;

1. Percent sign: %%
2. Left bracket: %CHAR[91]%
3. Right bracket: %CHAR[93]%
4. Comma character: %,

Note that this restriction of using the four characters is only necessary when static text is written as part of a macro's parameter list.

Special characters in macro parameter list

The macro LENGTH will return the number of characters in the parameter, e.g. %LENGTH[the text]% would return '8' (the space between the and text is also counted). If the text that is in the parameter contain any of the special characters it may cause an error, e.g. %LENGTH[100%]%.

In this case the % character in the text could be mistaken as a macro start and result in an error. To be able to count the characters in the text '100%' we will need to escape it with a percent character. The macro %LENGTH[100%%]% will return '4'.

Alternative to the escaping %

There is a special macro named CHAR which will return a character based on a UTF value. This macro may be used to return a wanted special character. If the text '100%' is wanted the macro text could be '100%CHAR[37]%'.

Note that the restriction of not using the special characters in the macro parameter list is only count for when the characters are used directly. If a macro is used which may result in any of the special characters this will not be an error.

Macros returning special characters are ok

Looking back to the examples above we want to have a macro that gives us the length of the text '100%'. We will use the macro CHAR to represent the single percent character (we could have used '%%' as well). The final macro (which is ok and will not raise an error) will be;

```
%LENGTH[100%CHAR[37]%]%
```

Note that this will not be an error where you might have suspected that the macro CHAR will result in a single percent and it would then be an error when the macro LENGTH is seeing the single percent character. The macros are parsed in a way that this error never will occur.

6. Macro tags for FileTrain

In the list below you will find the macro tags available throughout the FileTrain setup. By selecting a macro the more detailed description of that macro will show.

Some macro tags are only available in in certain fields in the setup. The description of the macro tags will let you know of any such restrictions. If no special usage is specified in the description field the macro tag may be used in all macro fields.

Macro list

%,	EMAIL_SENDER_ADDRESS	IMAGE_WIDTH	PF
%%	EMAIL_SENDER_NAME	INDEX_OF	REPLACE_ALL
ACTION_NAME	EMAIL_SENT	IPTC	SFN
APP	EMAIL_SUBJECT	LAST_INDEX_OF	SOURCE_FILE_EXISTS
BASE64	EXIF	LENGTH	SOURCE_FOLDER_FILE_COUNT
BASE64_FROM_FILE	F	LOWER	SOURCE_NAME
BUCKET	FILE_CONTENT	MATH	STATION_BUCKET
BUCKET_FILE_COUNT	FILE_EXISTS	META	STATION_NAME
BUCKET_FILE_NAME_LIST	FILE_PARENT_PATH	MOD	STRIP
BUCKET_LIST	FILE_PATH	MOD_TIME	SUBSTRING
BUCKET_LIST_REMOVE_DUPLICATES	FILL	MP3_ALBUM	SXML
CASE	FILTER_NAME	MP3_ARTIST	TRIM
CHAR	FOLDER_FILE_COUNT	MP3_COMMENT	UPPER
D	FOLDER_SIZE	MP3_GENRE	UUID
DATE_COMPARE	FOR_EACH_IN_LIST	MP3_TITLE	XMP
DUPLICATE_FILE_EXTRA	FORMAT_DATE	MP3_TRACK	XPATH
E	FREE_INDEX	MP3_TRACKCOUNT	XPATHLIST
EMAIL_BODY	HTML_ENCODE	MP3_YEAR	ZIPF
EMAIL_RECEIVED	IMAGE_HEIGHT	NR	

Macros

%,

Special macro that will return a single comma character ','.

%%

Special macro that will return a single percent character '%'.

%ACTION_NAME%

The name of the action in the current context where the macro is evaluated.

%APP[<key>**]%**

This macro is mainly for internal use.

In the container of the running application, servlet or applet there are some internal parameters which may be called with this macro. The parameter key differs for each container.

%BASE64[<text>, <optional encoding>**]%**

This will take a text and make a base64 encoded text from it. The optional encoding may be used if the source text is in a specific format.

%BASE64_FROM_FILE[<path>**]%**

This will take a file and put create a Base64 encoded string from the content. The path argument is the full path to the file to encode.

%BUCKET[<key>**]%**

Gets a value from the object's bucket (i.e. the object that is currently being handled). This can for example be useful in an action to get a value from another action that has been invoked before the current action.

The <key> is the key that previous action has used to put the value in the bucket. See the documentation for different actions to see what they put in the bucket.

Note that the station itself also puts some values in this bucket before any action is executed.

Note that this bucket is different from the station bucket which may be used with the macro %STATION_BUCKET[...]%.

Read more about the different buckets in chapter 2.

%BUCKET_FILE_COUNT[<key>,<onlyFiles>**]%**

This will return the number of files that are present in the object bucket with the key *<key>*. If *<onlyFiles>* is equal to 'true' then the count will only include the files, if 'false' the count will include files and folders. The value in the bucket has to be a list of files (see different actions for more information what they put into the bucket).

%BUCKET_FILE_NAME_LIST[<key>,<onlyFiles>**]%**

This will return the names of files that are present in the object bucket with the key *<key>*. If *<onlyFiles>* is equal to 'true' then the names will only include the files, if 'false' the names will include files and folders. The value in the bucket has to be a list of files (see different actions for more information what they put into the bucket).

The returned list is a comma separated list of file names.

%BUCKET_LIST[<key>,<delimiter>**]%**

This is a macro that will join a list to a single string with a certain delimiter.

This macro will take a bucket value, regard this value as a list of items (for example from the special -list bucket key created in the database action) and join these items with the specified delimiter.

Example: If the bucket key **paper-list** contains a result list from database action and has the following items: **news, sports, ads**. Using the macro **%BUCKET_LIST[paper-list,#]%** will result in the following string: **news#sports#ads**

NB! If a comma (,) is wanted as the delimiter, the comma must be preceeded with a percent sign (%,).

%BUCKET_LIST_REMOVE_DUPLICATES[<key>**]%**

This macro will work on a bucket list (e.g. a value from a database call). The macro will not return anything, it will only remove all the duplicate entries in the given bucket value.

Example: If the bucket key **paper-list** contains a result list from database action and has the following items: **news, sports, news, ads**. Using the macro **%BUCKET_LIST_REMOVE_DUPLCIATES[paper-list]%** will change the items in the bucket, resulting in the following items: **news, sports, ads**

%CASE[<exp>,<t>,<f>**]%**

This macro is evaluating a condition and will return different values depending on the condition result.

The paramater <exp> which is evaluated must be a valid expression, see below.

If the expression is evaluated to true, the <t> text (which may contain other macros) is parsed and returned. If the expression is false the <f> text is parsed and returned.

The expression <exp> must contain one (the first occurance is taken) of the following mathematical expression

Expression	Meaning	Description
<	less than	This expression is only valid if the left and right component are numbers.
<=	less than or equals	This expression is only valid if the left and right component are numbers.
>	bigger than	This expression is only valid if the left and right component are numbers.
>=	bigger than or equals	This expression is only valid if the left and right component are numbers.
==	equals	Checks if the left and right components are equal.
<>	not equals	Checks if the left and right components differ.

%CHAR[<no>**]%**

Will return the character that has the unicode decimal number <no>.

%D%

%D[<format>**]%**

%D[+<d>,<format>**]%**

%D[-<d>,<format>**]%**

Simple form

In the simple form, returns the current date in the format yyyy-MM-dd.

Variant [<format>]

If <format> is specified the result will be in the specified format. The format may contain special characters such as dot (.), comma (,) (note! if a comma is to be used as separator it must be escaped with a '%' in order to be handled correctly) and hyphen (-). The characters below shows how to get a certain format. The number of each character decide the result date string, e.g. in the year 2006, yyyy would result in 2006 while yy gives 06.

Format character	Description
y	The year
M	The month in year
w	The week in year
W	The week in month
D	The day in year
d	The day in month
E	Day in week as text
a	AM/PM
H	Hour in day (0-23)
k	Hour in day (1-24)
K	Hour in am/pm (0-11)
h	Hour in am/pm (1-12)
m	Minute in hour
s	Second in minute
S	Millisecond

Variant[+<d>,<format>] or [-<d>,<format>]

If the first parameter starts with a plus sign or minus sign the specified <d> number of days will be added or withdrawn respectively from the current date before the result is calculated. The <format> is as described above.

%DATE_COMPARE[<date1>, <date1 format>, <date2>, <date2 format>]%

This macro will compare two dates and return the difference in seconds between them.

The parameter date1 and date2 are strings representing dates. Each of them should conform to the date1 format and date2 format (see macro %D% for more information about the formats).

The returned value is the difference between the two dates in seconds. If the dates are the same 0 is returned. If date1 is before date2 a negative number is returned.

%DUPLICATE_FILE_EXTRA[<text>]%

This macro is only used in actions where a destination file is created. This macro will have no effect if the file that is created does not exists. If the file being created exists this macro will determin how a new name will be created. Note that the special macro %FREE_INDEX% may be used in this macro's text parameter.

In most cases where this macro may be used there is a checkbox in the setup allowing the file to be overwritten. This checkbox has no effect if this macro is used in the action.

This macro replaces the setup in FileTrain version 5 and older where the user could specify a prefix to add to the file name if the result file existed. With this macro the file name may be changed in more ways than just a prefix.

NB! This macro can not currently be used in the FTP send action!

Example 1

If we have setup a workflow where a source file is detected and moved to a destination folder and a file with this name already exists in the destination folder we could use the checkbox in the move action allowing us to overwrite the destination file with our newly found file.

If we want to keep both files in the destination folder we must change the name for this new file. Let's assume we would like to add a hash character (#) and a running number on the new file. We would then use the name change field in the move action and put the following value:

%F%**%DUPLICATE_FILE_EXTRA[#%FREE_INDEX%]%**%E%

In the example above the bold text is the magic which is added if the file is a duplicate file. Without the duplicate macro the file name would become %F%%E%, i.e. the original file name.

If we rather have the string 'version X' in front of the file name in case of duplicate (where X will be the first available number) we could use the following in the name change field:

%DUPLICATE_FILE_EXTRA[version %FREE_INDEX%]%%F%%E%

%E%

Note that this macro is only useful in the context where a single file is referred.

File name suffix (dot included) if any. If the file does not have any suffix an empty string is the result.

%EMAIL_BODY%

%EMAIL_BODY[<mime>]%

%EMAIL_BODY[<mime>**]%**

Simple form

If the object being handled is an email this will return the body of the email. If the email contains more than one body the returned text will primarly be the plain text body, secondly the html coded body will be returned.

Variant [<mime>]

This will primarly return the email body in the special mime format <mime>. If the email does not contain any body in the specified mime format an empty string is returned. Example of mime types: text/plain or text/html

---

%EMAIL_RECEIVED%

%EMAIL_RECEIVED[<format>**]%**

%EMAIL_RECEIVED[+<d>,<format>**]%**

%EMAIL_RECEIVED[-<d>,<format>**]%**

This will, if the object being handled is an email, return the email's received date.

This macro works as the date macro %D% with all the variants of parameters.

%EMAIL_SENDER_ADDRESS%

If the object being handled is an email, this will return the sender's email address.

%EMAIL_SENDER_NAME%

If the object being handled is an email, this will return the sender's full name.

%EMAIL_SENT%

%EMAIL_SENT[<format>**]%**

%EMAIL_SENT[+<d>,<format>**]%**

%EMAIL_SENT[-<d>,<format>**]%**

This will, if the object being handled is an email, return the email's sent date.

This macro works as the date macro %D% with all the variants of parameters.

%EMAIL_SUBJECT%

If the object being handled is an email, this will return the email's subject text.

%EXIF[<no>**]%**

If the object being handled is an image with EXIF information, this will return the EXIF information found in EXIF field number <no>.

%F%

%F[<s>,<e>**]%**

%F[O,<ch>**]%**

%F[O,<ch>,<no>,<l>**]%**

%F[OI,<index>,<no>,<l>**]%**

Note that this macro is only useful in the context where a single file is referred.

Note that this macro is only useful in the context where a single file is referred.

Simple form
The file name stem (i.e the file name but without the file name suffix and belonging dot).

Variant [<s>,<e>]

Part of the file name stem. Start at index <s> (the first character has index 1) and end at index <e>. If <s> is not specified, 1 is used. If <e> is missing the last character index is used. Note that the comma characters must be present even if either <s> or <e> is left out.

Variant [O,<ch]>

The first parameter is the letter 'O'. This gives a part of the file name stem starting at the first character and take all characters before the first occurance of <ch>. If <ch> is not found in the file name the result is the same as %F%.

Variant [O,<ch>,<no>,<l>]

The first parameter is the letter 'O'. This gives a part of the file name stem starting at the first occurance of <ch>. Then move <no> steps (a minus sign may preceed <no> to go backwards). This is now the start position and <l> number of characters are read. If <l> is missing the rest of the file name stem is read. Note that the last comma must be present even if <l> is left out.

Variant[OI,<index>,<no>,<l>]

The first parameter is the letters 'O' and 'I'. This gives part of the file name stem starting at character <index> (first character is index 1, if the last character is wanted then use -1). Then <no> (number) steps will be taken (minus sign may preceed <no> to go backwards). This is the start and from here <l> number of characters are read. If <l> is missing the rest of the file name stem is read. Note that the last comma must be present even if <l> is left out.

%FILE_CONTENT%
%FILE_CONTENT[<file>,<format>,<start>,<end>]%

Reading the file with path <file> (the path is depending on the environment from which this macro is called) and returns the content in the file in the selected text format <format>.

If the format is omitted, 'UTF-8' is used as default format. Note that the comma must be present even if <format> is left out.

Optionally the <start> and <end> parameters may be used. These will indicate the start and end location to get a portion of the file content. The start and end parameters are the byte location. The first location in the file is zero (0).

The simple macro without any parameters will get the content of the file currently in the scope of the macro.

When using parameters the <file> parameter may be empty in which case the file currently in scope will be used, e.g. %FILE_CONTENT[,ISO-8859-1]%

%FILE_EXISTS[<path>]%

Checks if a certain file (with the path <path>) exists. This will return "true" if the file exists and "false" otherwise. The path that is given must be availalbe from the FileTrain server. See also %SOURCE_FILE_EXISTS[...]%.

%FILE_PARENT_PATH%

Note that this macro is only useful in the context where a single file is referred.

The current file's parent folder path.

%FILE_PATH%

Note that this macro is only useful in the context where a single file is referred.

The full path to the current file.

%FILL[<start | end>,<text>,<filler>, <max length>**]%**

This will fill the <text> with a certain <filler>. The first parameter is either **start** or **end** depending on whether you want the filler to precede or append the text. The **max length**

parameter is the max number of characters that the result will have.

Example) Using %FILL[start,12345,0,10]% will result in **0000012345**

%FILTER_NAME%

The name of the filter in the current context where the macro is evaluated.

%FOLDER_FILE_COUNT[<path>**]%**

This macro will check a certain folder (disc folder available from the FileTrain server), count all the files in that folder and return the count.

The result is the number of non-hidden files (excluding folders) found directly in the folder.

%FOLDER_SIZE[<path>**]%**

This macro will check a certain folder (disc folder available from the FileTrain server), count the number of bytes of all files, including subfolders and return the count.

The result is the byte size of all non-hidden files (including subfolders) found in the folder.

%FOR_EACH_IN_LIST[key,tmpKey,macro**]%**

This macro will act on each item in a bucket list.

The **key** parameter is the bucket key that currently holds a list of values (for example from a database search).

The parameter **tmpKey** can be any value and this will be used to place each entry in the bucket list to perform whatever is set in the macro parameter. Make sure that you use a unique value for the tmpKey parameter.

The last parameter, **macro**, can be any other macro or static text. Each entry in the bucket list will be set to the result of this macro parameter.

Example:

LEt us assume we have a database search where we have saved a list of values in the bucket key **dbResult-list**. TLet us further assume that the values in this list are a set of dates on the format yyyy-MM-dd and we want to change this format on each element in the list to the new format dd/MM/yyyy.

When using this macro we must make sure that the temporary key that we specify is unique or we may set a current bucket key to a fault value. In this case we will use the name **tmpValue**.

The resulting macro will be (the last parameter, macro, is in bold text for clarification purpose)

%FOR_EACH_IN_LIST[dbResult-list,tmpValue,**%FORMAT_DATE[%BUCKET[tmpValue]%,yyyy-MM-dd,dd/MM/yyyy]%**]%

%FORMAT_DATE[<text>,<from>,<to>**]%**

This macro is used to convert a text from one date format to another. For the different pattern characters, see the %D% macro.

The <text> is the source date specified in the <from> format. The result is the same date specified in the <to>

format.

%FREE_INDEX%

This macro is a special macro only used within the macro %DUPLICATE_FILE_EXTRA[...]%

This macro will return the first free index in a duplicate file scenario.

%HTML_ENCODE[<text>]%

Returns a HTML formated text based on the <text>. Entities such as ä will be used instead of ä and so on.

%IMAGE_HEIGHT[<format>]%

Returns the image height of the currently handled image file (if any). The **format** parameter is optional and may be either **mm** or **px** resulting in the image height specified in millimeters and pixels. If the format parameter is not present (using %IMAGE_HEIGHT%) the height is returned in pixels.

%IMAGE_WIDTH[<format>]%

Returns the image width of the currently handled image file (if any). The **format** parameter is optional and may be either **mm** or **px** resulting in the image width specified in millimeters and pixels. If the format parameter is not present (using %IMAGE_WIDTH%) the width is returned in pixels.

%INDEX_OF[<s>,<t>,<start>]%

The text <s> is searched and the index of the first occurance of <t> is returned. The index is zero based meaning that if <s> starts with <t> the returned index is '0'. If the <t> is not found in <s> -1 will be returned.

Both <s> and <t> might contain macro text themselves.

If the optional <start> parameter is present this is the start index in the text from which the search starts. Any occurance of <t> before this index is not counted. The <start> index is zero-based meaning that the first character in <s> has the index 0 (zero).

%IPTC[<no>]%

If the object being handled is an image with IPTC information, this will return the IPTC information found in IPTC field number <no>.

NB This will only read from the regular IPTC information, not the XMP information. For XMP data use the XMP macro.

%LAST_INDEX_OF[<s>,<t>,<start>]%

The text <s> is searched and the index of the last occurance of <t> is returned. The index is zero based meaning that if <s> starts with <t> the returned index is '0'. If the <t> is not found in <s> -1 will be returned.

Both <s> and <t> might contain macro text themselves.

If the optional <start> parameter is present this is the index in the text from which the search starts (seen from the end). Any occurance of <t> after this index is not counted. The <start> index is zero-based meaning that the first character in <s> has the index 0 (zero).

%LENGTH[<text>]%

Returns the length of the <text>. The number of characters in <text> is counted and returned.

This macro may be very useful used for example in an expression, see macro CASE for more information.

%LOWER[<text>]%

The returned value is <text> where each character is in lower case in the returned value.

%MATH[<math command>, <leftExpression>, <rightExpression>, <decimals>**]%**

This will invoke the mathematical expression <math command> on the <leftExpression> and <rightExpression>. The returned number will have the maximum decimal count that is expressed in the parameter <decimals>. The <leftExpression> and <rightExpression> may contain numbers (possibly from other macros).

The <decimals> parameter is optional. If <decimals> is left out the last comma should also be left out.

The <math command> may be any of the following;

Math command	Description
+	Will add <leftExpression> with <rightExpression>, e.g. %MATH[+, 12, 3]% will return '15'
-	Will subtract <leftExpression> with <rightExpression>, e.g. %MATH[-, 12, 3]% will return '9'
*	Will multiply <leftExpression> and <rightExpression>, e.g. %MATH[*, 12, 3]% will return '36'
/	Will divide <leftExpression> with <rightExpression>, e.g. %MATH[/, 12, 3]% will return '4'

%META[<id>**]%**

This macro is not currently used in FileTrain.

%MOD%

%MOD[<format>**]%**

%MOD[+<d>,<format>**]%**

%MOD[-<d>,<format>**]%**

Note that this macro is only useful in the context where a single file is referred.

The macro will return the modification time for the current referenced file. The variants for the macro follow the D macro where more information may be found for the different parameters.

%MOD_TIME%

%MOD_TIME[<path>**]%**

Gets the modification time of a file in milliseconds.

The default version will get the modification time from the current file in scope. The alternative is to pass a file path as parameter which will get the modification time of that file.

%MP3_ALBUM%

If the object being handled contains ID3 tags the value from the album tag is returned.

%MP3_ARTIST%

If the object being handled contains ID3 tags the value from the artist tag is returned.

%MP3_COMMENT%

If the object being handled contains ID3 tags the value from the comment tag is returned.

%MP3_GENRE%

%MP3_GENRE%

If the object being handled contains ID3 tags the value from the genre tag is returned.

%MP3_TITLE%

If the object being handled contains ID3 tags the value from the title tag is returned.

%MP3_TRACK%

If the object being handled contains ID3 tags the value from the track tag is returned.

%MP3_TRACKCOUNT%

If the object being handled contains ID3 tags the value from the track count tag is returned.

%MP3_YEAR%

If the object being handled contains ID3 tags the value from the year tag is returned.

%NR[<text>**]%**

The returned value is <text> where each carriage return and line feed is removed. The carriage return is replaced with a space character if there is no space preceeding the carriage return. This means that even if there are multiple carriage returns immediately following eachother in the original text, they will only be replaced by maximum one space character.

%PF%

Note that this macro is only useful in the context where a file is referred.

This macro is similar to the F macro but the name is taken from the parent folder. All variants of the macro that are applicable to the F macro are also available to this macro.

%REPLACE_ALL[<s>,<source>,<target>**]%**

Returns a modified version of <s> where all the occurancies of <source> are replaced with <target>.

> **Regular expression**
>
> <source> is a regular expression so use this macro with care. Make sure you are familliar with regular expressions in Java before using this macro.

%SFN[<text>**]%**

Safe File Name - This will return the <text> where all occurancies of the following characters are replaced with underscore (_);

Character	Character name
?	Question mark
"	Double quotes
!	Exclamation mark
&	Ampersand

/	Forward slash
\	Backward slash
:	Colon
.	Dot. Only if the name is starting with a dot that particular dot is replaced as well, other dots within the file name are not replaced.

%SOURCE_FILE_EXISTS[<path>]%

This macro may be used to check if a certain file exists. The macro will use the specific source in the scoop of the macro to find the file. If this source is a FTP source the FTP server will be checked for the file wanted.

Compared to the macro %FILE_EXISTS[...]%, this macro may also be used in a FTP source context to check if a certain file exists on the FTP server.

%SOURCE_FOLDER_FILE_COUNT[<path>]%

This macro will check a certain folder found on the source, count all the files in that folder and return the count.

The result is the number of non-hidden files (excluding folders) found directly in the folder. The macro will use the specific source in the scoop of the macro to find the file. If this source is a FTP source the FTP server will be checked for the file wanted.

Compared to the macro %FOLDER_FILE_COUNT[...]%, this macro may also be used in a FTP source context to get the count from the FTP server.

%SOURCE_NAME%

The name of the source which is the invoking source of the object in the current context where the macro is evaluated.

%STATION_BUCKET[<key>]%

Gets a value from the current station's bucket. Teh station bucket is a static bucket which holds information that will remain in the station as long as the station exists. This bucket is different than the object bucket which is used with the macro %BUCKET[...]%.

Read more about the different buckets in chapter 2.

%STATION_NAME%

The name of the station within which context the macro is evaluated.

%STRIP[<text>]%

Special macro that will replace all characters except the ones in the list below with a underscore (_)

Characters that will remain unchanged are:

. (dot)

- (dash)

_ (underscore)

0-9 (numbers)

a-z (character a to z)

A-Z (character A to Z)

%SUBSTRING[<s>,<start>,<end>]%

Returns a portion of the soruce string <s>. The portion is taken from index <start> (included) and ends at index

<end> (not included). If <end> is missing the rest of the string is taken from the start. If <start> is missing the substring will be taken from the start. The index are zero-based meaning that the first character in <s> is having number 0.

Note that if <start> and/or <end> is left out the commas still need to be present.

%SXML[<text>**]%**

Returns a modified version of <text>. The return value is specialized to fit in XML documents where characters are replaced with entities, e.g. 'å' is replaced with 'å'. Characters with UTF character number more than 128 are replaced with the entity '&#<no>;' where <no> represents the UTF value of the character.

The string <text> may contain macros.

%TRIM[<text>**]%**

The result from this macro is the text parameter without any space characters before and after the actual text. Space characters that will be removed includes space, tab, carriage return and line feed.

%UPPER[<text>**]%**

The returned value is <text> where each character is in upper case in the returned value.

%UUID%

The returned value is a unique ID.

%XMP[<xmpID>**]%**
%XMP[<xmpID>, filePath**]%**

The XMP data from the file isread and the value from the specified xmpID is returned.

The xmpID is an identifier which is unique for each XMP data. The default XMP data fields' identifiers may be seen as tooltip in the special XMP popup menues that are available in the setup. Customized XMP data has their own identifier specified when created chosen by the user. The xmpID is the XMP prefix followed by a colon and then the property field name, e.g. **tiff:Compression**

The value returned is always a string representation even if the value may be a date or another object.

If no filePath parameter is given, the file currently in scope of the macro will be read, otherwise the file given by the filePath.

%XPATH[string, xpath expression,prefix=URI,prefix=URI,...**]%**

This macro will calculate an XPath expression from the given string (XML format). The returned result is a string representing the result of the XPath expression.

The parameter **string** is the XML string on which the XPath expression will be calculated.

Only the string and xpath expression parameters are compulsory, the rest of the parameters are optional and if exists they should all have the following structure: prefix=URI. These parameters are used to identify namespaces in the XML string, the prefix is the namespace prefix and the URI is the namespace URI. For example the IPTC core namespace would have the following value: **Iptc4xmpCore=http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/**

For more information about valid XPath expressions, see http://docs.oracle.com/javase/tutorial/jaxp/xslt/xpath.html

---

%XPATHLIST[string, xpath expression, bucket key, prefix=URI,prefix=URI,...**]%**

This macro will calculate an XPath expression from the given string (XML format). The returned result is a list of values and will be saved in the current bucket with the given **bucket key**.

The parameter **string** is the XML string on which the XPath expression will be calculated.

Only the string, xpath expression and bucket key parameters are compulsory, the rest of the parameters are

optional and if exists they should all have the following structure: prefix=URI. These parameters are used to identify namespaces in the XML string, the prefix is the namespace prefix and the URI is the namespace URI. For example the IPTC core namespace would have the following value: **Iptc4xmpCore=http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/**

The list which is placed in the bucket key may be used in other special macros handling lists, e.g. %BUCKET_LIST[...]%.

For more information about valid XPath expressions, see http://docs.oracle.com/javase/tutorial/jaxp/xslt/xpath.html

%ZIPF%

Note that this macro is only useful in the context where a zip file is referred.

This macro is similar to the F macro but the name is taken from the zip file (e.g. in the unzip action where %F% will give the unzipped file name but %ZIPF% will give the original zipped file name). All variants of the macro that are applicable to the F macro are also available to this macro.