

Media Server Manual

TABLE OF CONTENTS

Introduction	2
Main Operation	2
Installation and initial configuration	3
Media Server Configuration Tool	3
Main interface	3
Menus and buttons	6
File menu.....	6
Close.....	6
Save all changes	6
Reload all jobs	6
Add job	6
Import	6
Import 2.0	7
Export.....	7
Services menu	8
Start/Stop service.....	8
Log files menu	8
Result log.....	8
Event log.....	9
Settings menu.....	9
Database settings.....	9
Application	9
Twitter accounts	9
Facebook accounts.....	11
License.....	14
About.....	14
Object related menus.....	15
Job	15



Transformation	16
Source	16
Filter	16
Context menus	17
Job configurations	17
Job	17
Show results panel	18
Add transformation.....	19
Transformation (general)	19
Transformation (Twitter stream)	19
Add filter	20
Link between transformations	21
Source	21
Filter	21
Preview window	22
Edit	22
Edit panels.....	22
Media Server Service	28
Scheduler.....	28
Alter job detection	29
Logging	29

The Media Server application is meant to constantly execute scheduled jobs, to get and transform source data so it can be used in the other Netpresenter software.

Media Server 3 is the successor of the current Media Server 2 (also called Newsfeed Server). It consists of 2 elements:

1. The Media Server Service
2. The Media Server Configuration Tool (user interface)

The Media Server Service is a Windows service meant to constantly execute jobs, without user interaction.

The Media Server Configuration Tool is a Windows Desktop application to configure jobs, to start and stop the Media Server service and to access log files and manage the application in general.

MAIN OPERATION

The Media Server periodically fetches data from one or more sources and converts them into a destination format usable in the other Netpresenter software (channel, image, ...).



Reading and processing the source data is done using one or more so-called transformations. A transformation executes actions like:

- Reading source data
- Selecting
- Sorting
- Transforming
- Joining
- Get extra data (e.g. images in an RSS feed)

Multiple transformations can be combined in a job.

The final result of a job can be a Netpresenter channel, with all content and layout attached to it, but it can also be a list of images or an encoded output for the Netpresenter App. Its results can also be used for applications outside the Netpresenter software suite.

INSTALLATION AND INITIAL CONFIGURATION

The Netpresenter support team will help you install and configure the Media Server. Please contact techsupport@netpresenter.com to schedule an online meeting.

MEDIA SERVER CONFIGURATION TOOL

Main interface

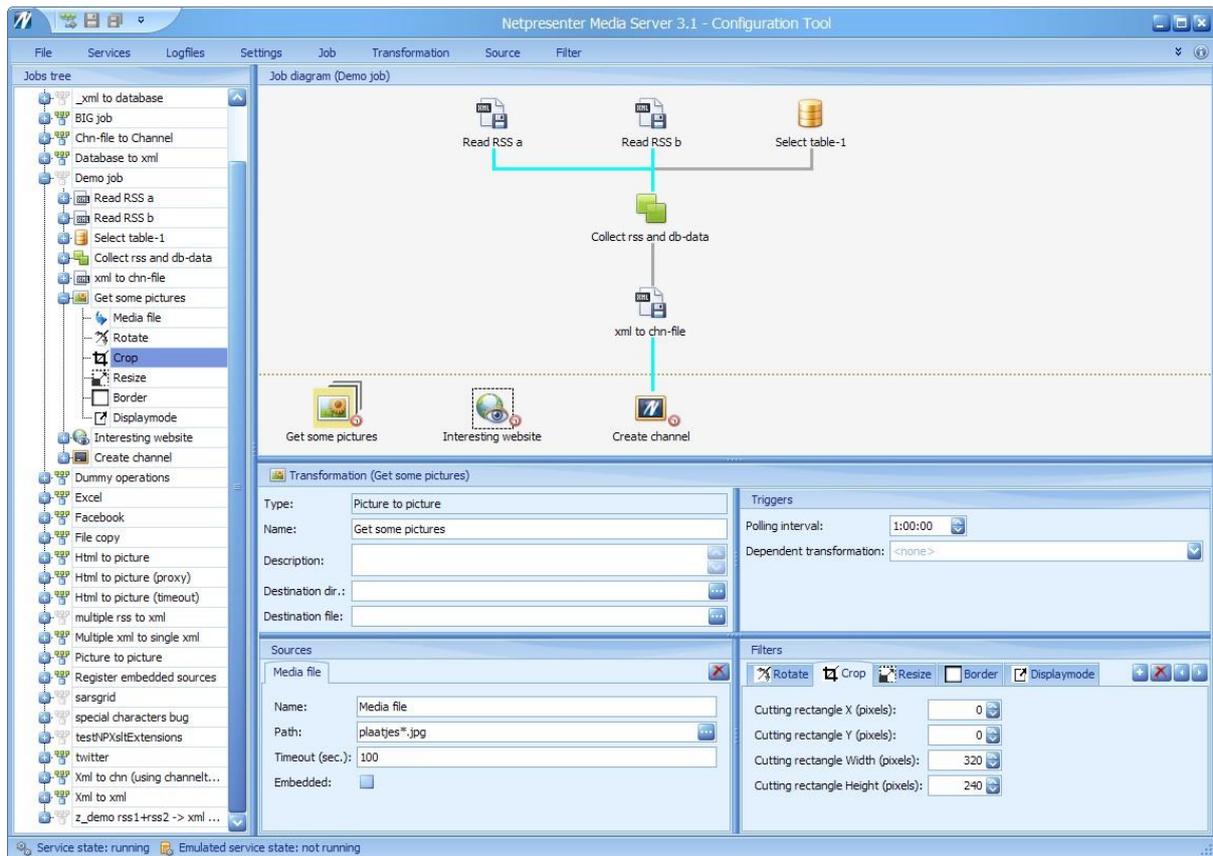
This section will describe the global architecture and functions of the main interface of the application.

The main screen is used to:

- a. Address almost all functions of the Media Server Configuration Tool;
- b. To see, manage, mutate and check job configurations.

The screenshot below shows an example of the main screen with a number of jobs created.





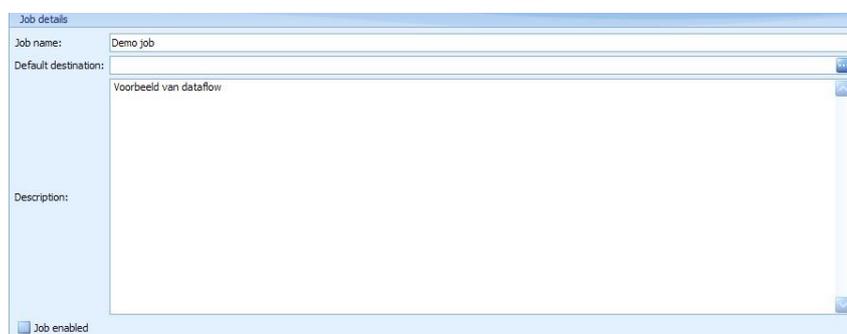
On the left side of the screen, you can see the *Jobs tree*, a list of all jobs configured until now. A gray icon means the job is not enabled. The service will only execute jobs that are enabled.

A job contains 1 or more transformations and a transformation can have 0 or more sources and filters. You can select a job by clicking its name in the job tree, in the example above you can see the *Crop* filter being selected, and it is part of the transformation *Get some pictures*, which is in turn part of the *Demo* job.

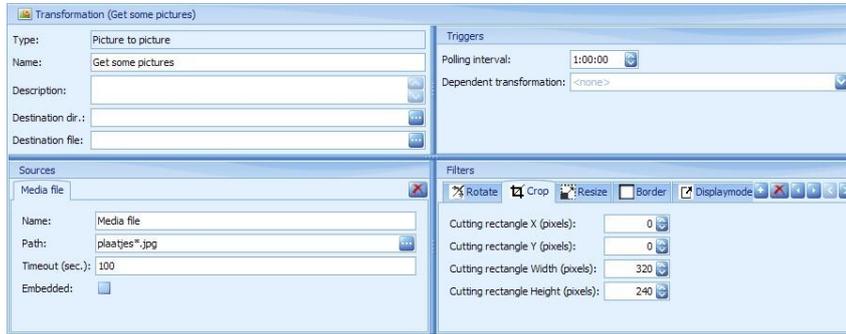
On the right side of the jobs tree, all the details of the job are shown. The upper half contains the job diagram, a graphical representation of the entire job. Meaning, all transformations and their interdependences. You can use the job diagram to select a transformation and activate functions via the context menus.

The region below the job diagram is used to display:

- a. job-properties (if you've selected a job, without selecting a transformation)



- b. the properties and configuration of the selected transformation (if you've selected a transformation or lower level object (source, filter))



The visual representation of a transformation contains 4 quadrants:

- a. top left: the properties of the transformation;
- b. top right: the triggers to start the transformation;
- c. bottom left: 0 or more sources for the transformation, every source will have its own tab containing its properties;
- d. bottom right: 0 or more filters for the transformation, every filter will have its own tab containing its properties. There are many types of filters, the filter type can be recognized by its name (=type) and the icon on the tab.

The buttons next to the tabs can be used for the following purposes (): add an object, delete the selected object, move object to the left, move object to the right, scroll to left tab, scroll to right tab. These buttons are only visible when relevant.

Via the main screen you can create, modify, check, test and delete all objects (jobs, transformations, sources and filters).

You can also check the status of transformations while being executed. For example: the green check next to this transformation:



means that the transformation was successfully executed. By hovering over the check;



you will see additional information

All the way at the bottom of the screen you can see the status of the Media Server Service:



 Service state: running

MENUS AND BUTTONS

This section sums up and describes all functions in the Media Server Configuration Tool.

File menu



Contains functions on application level.

Close

Closes the application.

Save all changes

Saves all changes in jobs.

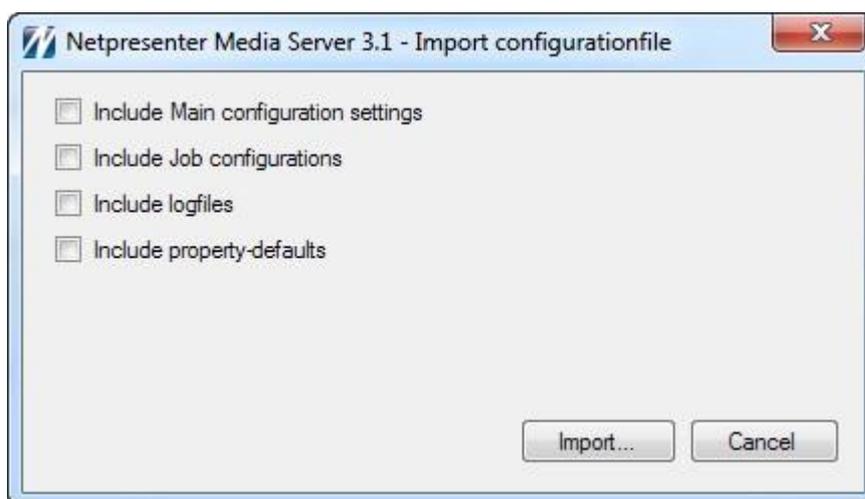
Reload all jobs

Reloads all jobs from the database. Unsaved changes will be lost.

Add job

Adds a new, empty, job to the configuration.

Import

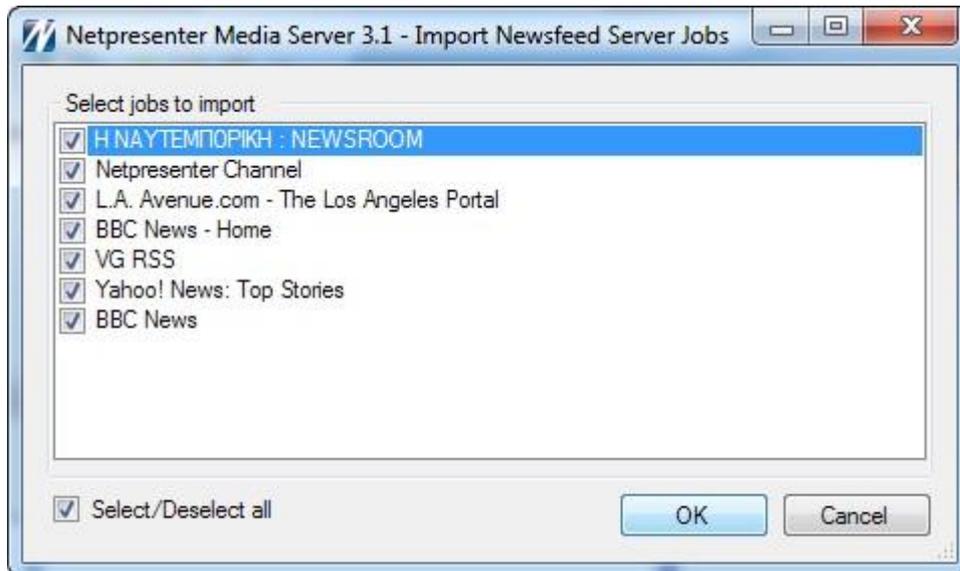


Imports settings from an XML file, created using the Export function. Please note: existing data in the application will be overwritten.

Import 2.0

Import from Newsfeed Server 2. It is possible to import jobs as an XML file from Newsfeed server 2.

Start by selecting the source XML file. Select all jobs to import into 1 single job in the Media Server and repeat for every new job you would like to build from the Newsfeed Server export.



Export



Can be used to export the current configuration as an XML file. These can be imported at a later stage. This can be useful for backup purposes or to migrate the configuration from one environment to another. When having an issue with the Media Server, you can send this XML file to the Netpresenter support team for analysis.

Services menu



Start/Stop service

To start or stop the Media Server Service.

Please note: The service can also be stopped and started via the Windows Services system tool.

Log files menu



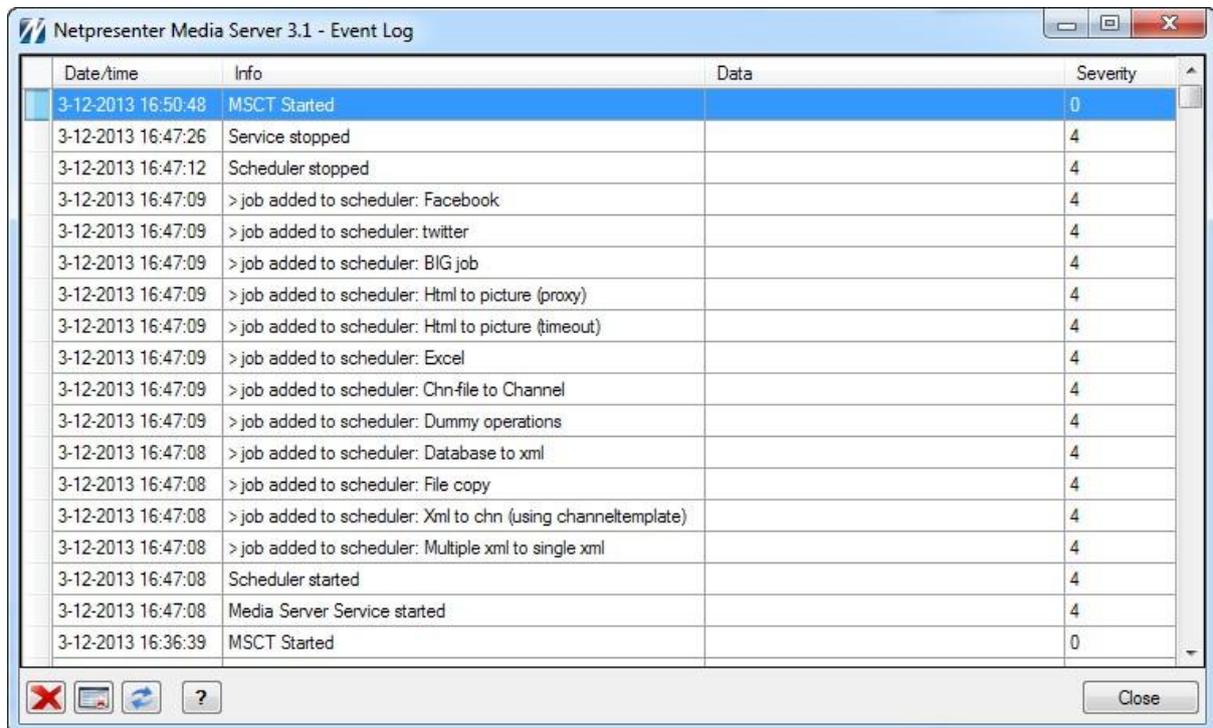
These log files will show status changes and information of general Media Server events and of the execution of transformations. Some messages will be published to the Windows event log (e.g. database errors).

Result log

Date/time	Job	Transformation	Info	Severity
3-12-2013 16:47:25	Html to picture	Html to picture (no wildcards)	Ended: Html to picture (no wildcards)	4
3-12-2013 16:47:12	Dummy operations	wait and crash	Ended: wait and crash	4
3-12-2013 16:47:12	Dummy operations	wait and timeout	Ended: wait and timeout	4
3-12-2013 16:47:12	Dummy operations	wait and timeout	Started: wait and timeout	4
3-12-2013 16:47:12	Dummy operations	wait and crash	Started: wait and crash	4
3-12-2013 16:47:12	Html to picture	Html to picture (no wildcards)	Started: Html to picture (no wildcards)	4
3-12-2013 16:47:12	Xml to xml	Multiple xml to single xml	Ended: Multiple xml to single xml	4
3-12-2013 16:47:12	Xml to xml	Multiple xml to single xml	Started: Multiple xml to single xml	4
3-12-2013 16:47:12	Xml to xml	wijzig tagnamen	Ended: wijzig tagnamen	4
3-12-2013 16:47:12	Xml to xml	wijzig tagnamen	Started: wijzig tagnamen	4
3-12-2013 16:47:11	File copy	Copy txt-files en plaatje	Ended: Copy txt-files en plaatje	4
3-12-2013 16:47:11	Database to xml	Database to xml (4)	Ended: Database to xml (4)	4
3-12-2013 16:47:11	Database to xml	Database to xml (3)	Ended: Database to xml (3)	4
3-12-2013 16:47:11	Database to xml	Database to xml (4)	Started: Database to xml (4)	4
3-12-2013 16:47:11	File copy	Copy image from web	Ended: Copy image from web	4
3-12-2013 16:47:11	Database to xml	Database to xml (3)	Started: Database to xml (3)	4
3-12-2013 16:47:10	Database to xml	Database to xml (2)	Ended: Database to xml (2)	4
3-12-2013 16:47:10	Database to xml	Database to xml (2)	Started: Database to xml (2)	4
3-12-2013 16:47:10	Database to xml	Database to xml (1)	Ended: Database to xml (1)	4
3-12-2013 16:47:10	Database to xml	Database to xml (1)	Started: Database to xml (1)	4

Shows an overview of all transformation starts and stops, also shows timeouts and errors at transformation level.

Event log



Date/time	Info	Data	Severity
3-12-2013 16:50:48	MSCT Started		0
3-12-2013 16:47:26	Service stopped		4
3-12-2013 16:47:12	Scheduler stopped		4
3-12-2013 16:47:09	> job added to scheduler: Facebook		4
3-12-2013 16:47:09	> job added to scheduler: twitter		4
3-12-2013 16:47:09	> job added to scheduler: BIG job		4
3-12-2013 16:47:09	> job added to scheduler: Html to picture (proxy)		4
3-12-2013 16:47:09	> job added to scheduler: Html to picture (timeout)		4
3-12-2013 16:47:09	> job added to scheduler: Excel		4
3-12-2013 16:47:09	> job added to scheduler: Chn-file to Channel		4
3-12-2013 16:47:09	> job added to scheduler: Dummy operations		4
3-12-2013 16:47:08	> job added to scheduler: Database to xml		4
3-12-2013 16:47:08	> job added to scheduler: File copy		4
3-12-2013 16:47:08	> job added to scheduler: Xml to chn (using channeltemplate)		4
3-12-2013 16:47:08	> job added to scheduler: Multiple xml to single xml		4
3-12-2013 16:47:08	Scheduler started		4
3-12-2013 16:47:08	Media Server Service started		4
3-12-2013 16:36:39	MSCT Started		0

Shows an overview of events at application level. Also shows which jobs were added to or removed from the scheduler.

Settings menu



Database settings

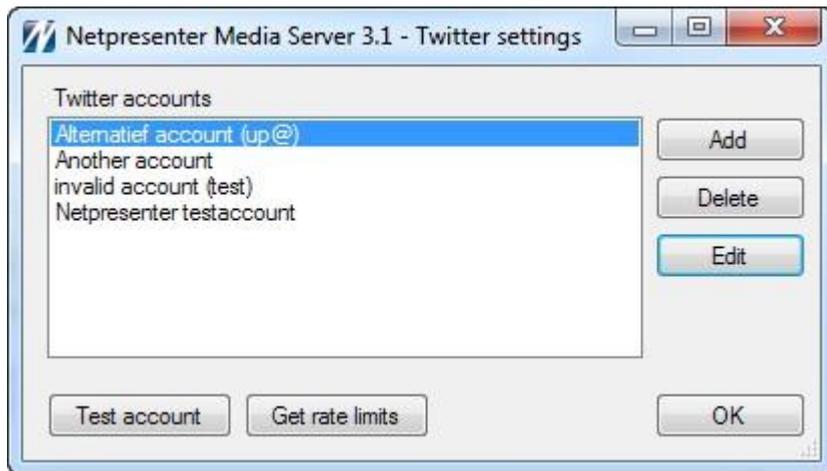
To set the connection parameters for the database.

Application

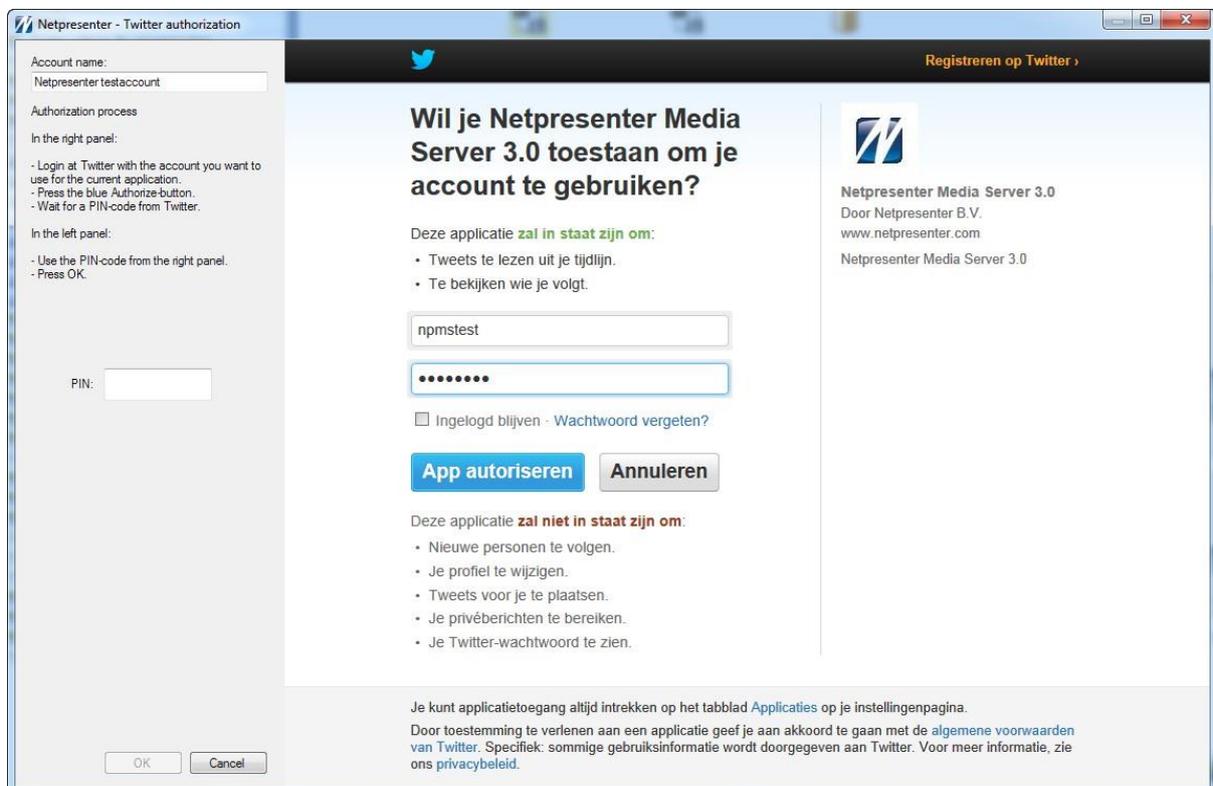
To set the application settings.

Twitter accounts

To be able to fetch tweets, the Media Server requires a valid Twitter account. The owner of the account needs to authorize the Media Server to access Twitter data via this account.



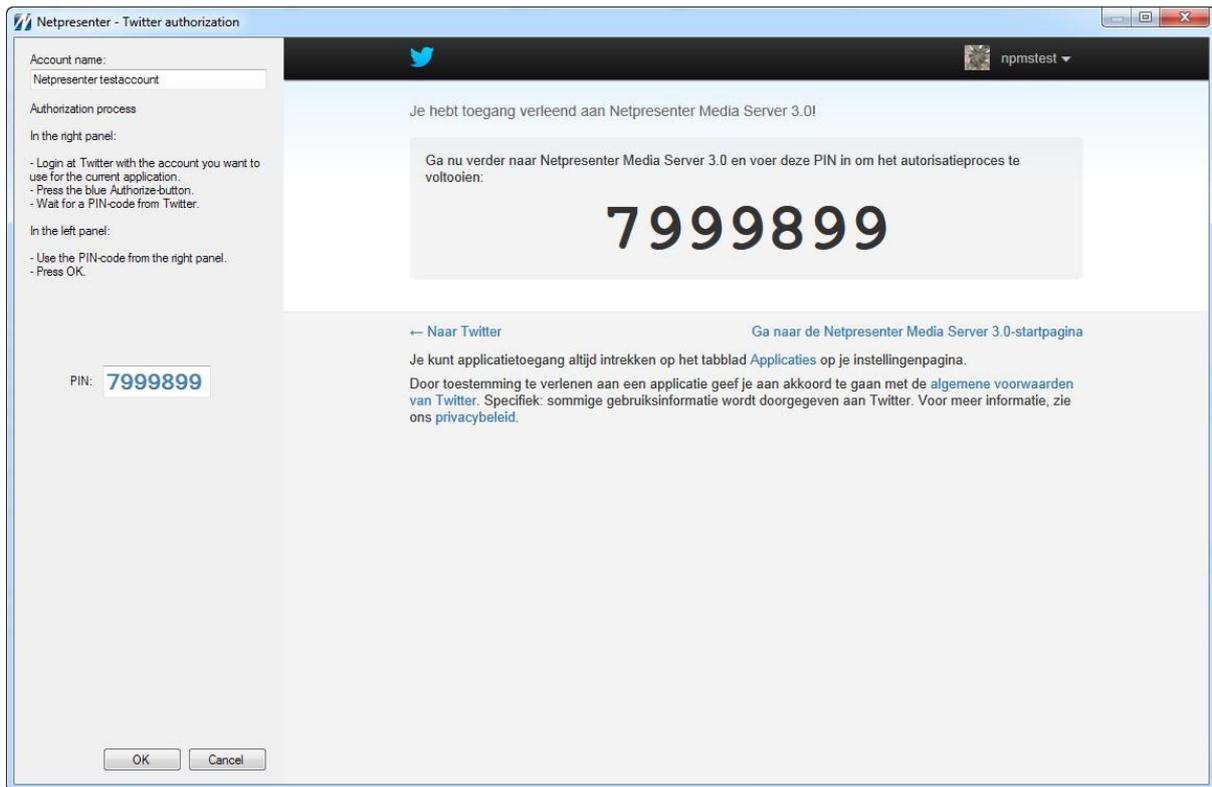
Choose the account and use Edit to change it. Use Add to add new accounts to the list.



Log in to Twitter with this account.

Please note: If you're logged in to Twitter automatically, please check whether it is the correct account you'd like to add. If not, log out and log in using the correct account.

Click 'Authorize App'.



Twitter now generates a PIN-code to be entered in to the PIN text field.

Click OK afterwards.

You can test the account using the 'Test account' button:



Important: a Twitter account can be used to read Twitter streams, but this can only be done for one stream per account.

Facebook accounts



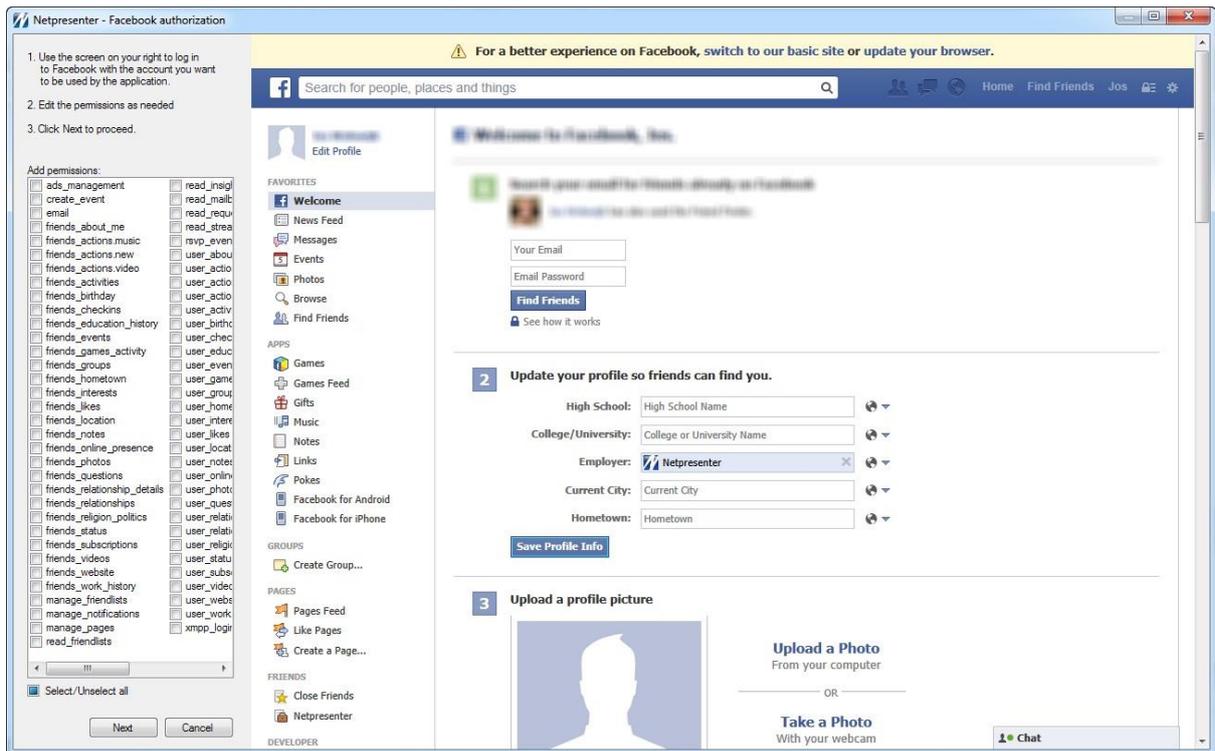


Select an existing account to extend it or to change the permissions for it. Use Add to add a new account.

If you're not currently logged in to Facebook, you will see the screen below:



Please note: if you are logged in automatically, be sure to check the account, if it is not the correct account, please log out and log in with the correct account.

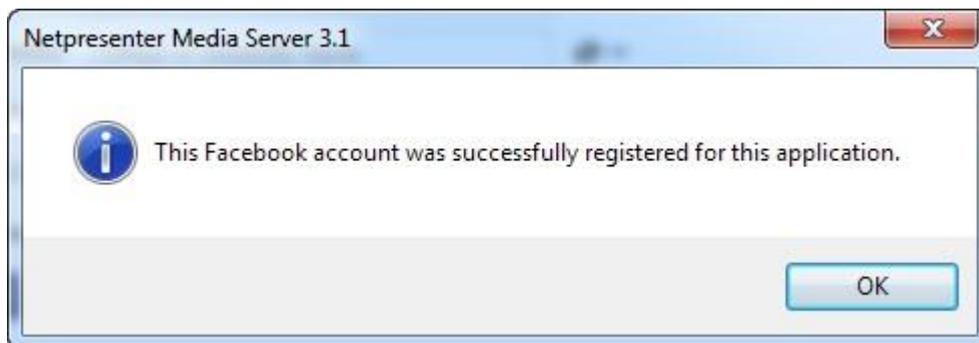


Change the permissions if necessary, based on the type of queries you plan to execute.

Click Next.



Click OK.



Important: a Facebook account expires after some time (usually 3 months). You will always need to extend the account manually, the Media Server cannot do this automatically.

License

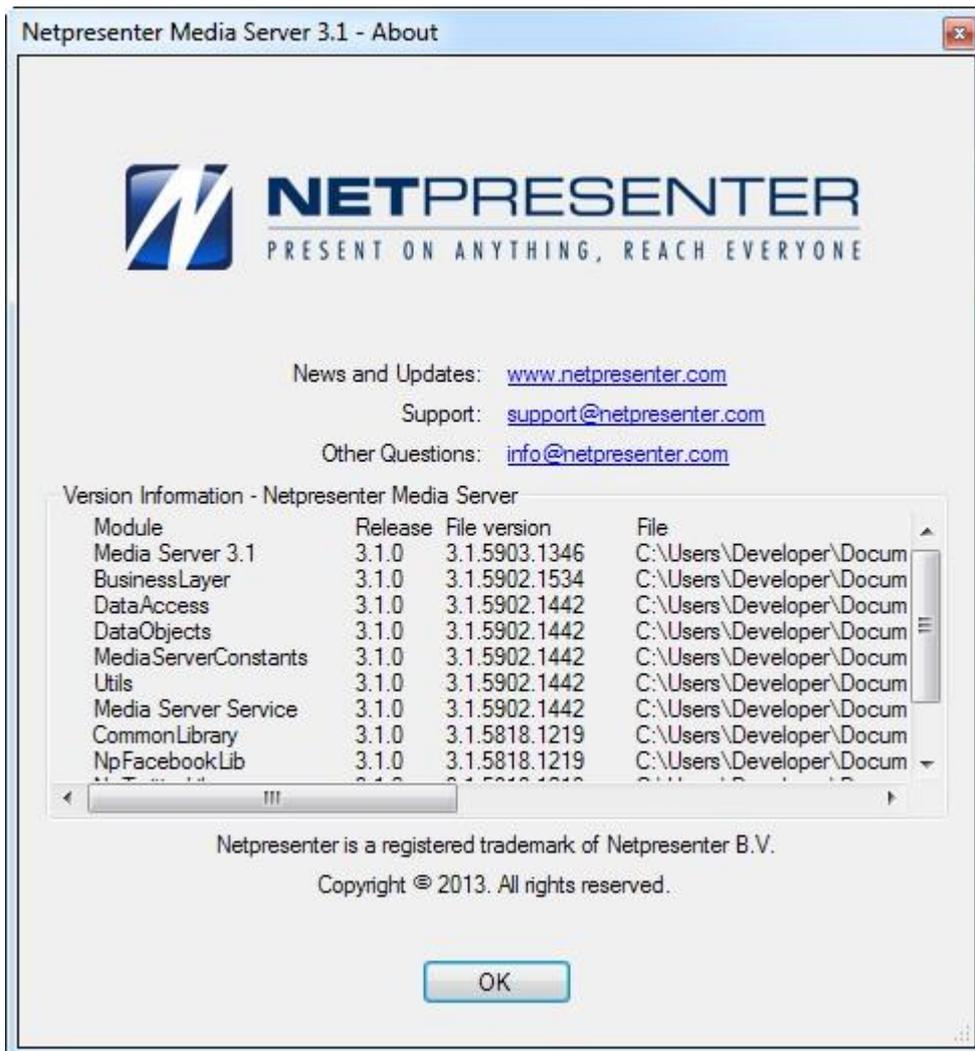
To check or update the license key.

About

This button is shown on the right side of the menu bar.



It will show a window with information about the current version of the Media Server.



Object related menus

Most of these menu options are self-explanatory and have tooltips to explain their functionality. This section contains a short explanation for each menu and button.

Job



Add transformation

Select the desired transformation type in the sub menu.

Validate job

To check the job for errors. Errors will be displayed in the Info/Results window.

Test all transformations

This starts all transformations for the selected job for one single run. All transformations are started simultaneously, it does not take intervals or depending transformations into account.

Start/Stop job test

This acts like a toggle (on/off) to start a 'service' for this job only. Results of the job's transformations will be shown in the Info/Results window.

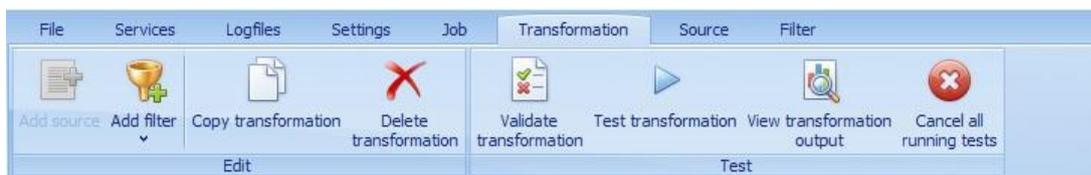
This simulates a job being executed by the actual Media Server Service, with a couple of differences:

- the job is started as visible in the interface (this might differ from the saved version);
- results are not logged in to the result log, but in to the Info/Results window.

Enable/Disable job

By enabling a job, a validation is done implicitly. If the job is invalid, an error will show and the job will remain disabled.

Transformation



Copy transformation

Creates an exact copy of the original transformation in the same job.

Delete transformation

Removes the transformation including all related data (sources and filters).

Please note: this function does not ask for confirmation, and there is no undo function. You can however cancel the job edit by returning to a previously saved version of the job.

Test transformation

This starts a single run of the selected transformation.

Source



Filter



Context menus

This section describes different context menus and consists of general descriptions. For individual descriptions of menu options, you can refer to the tooltips in the application (with the exception to some options that do require more explanation).

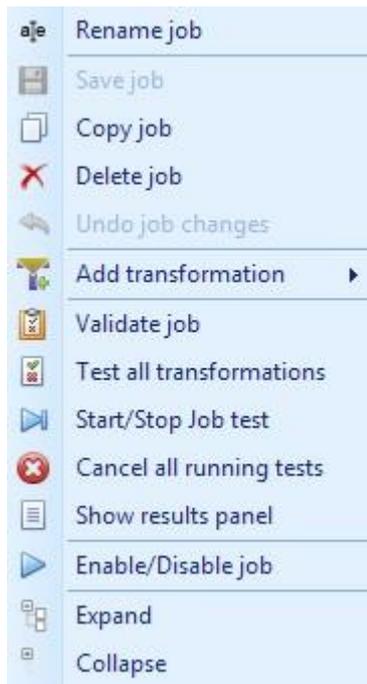
Job configurations

This context menu belongs to the root level when you select the root item in the job tree.



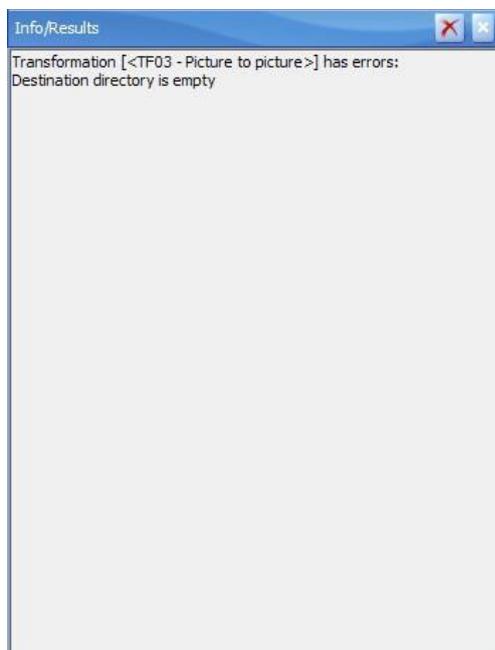
Job

This is the context menu of a job. Select any job in the job tree or click any free space in the job diagram.



Show results panel

Test- and processed results are viewable in the (floating) Info/Results panel.

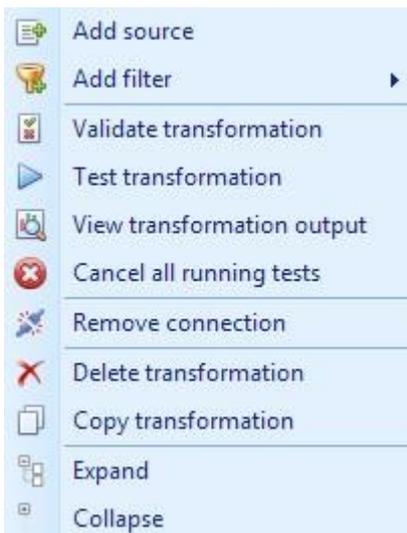


This panel will open automatically when information is outputted. It is possible to clear and hide this panel, change its position and shape. Aside from this, the panel can also be opened through the menu option Show results panel.

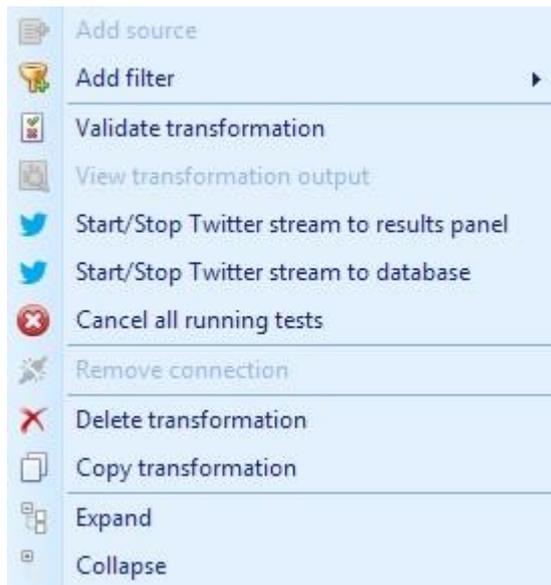
Add transformation



Transformation (general)



Transformation (Twitter stream)



Add filter

(The available filters depend on the kind of transformation).



-  Surround xml with header and footer
-  Rotate
-  Crop
-  Resize
-  Border
-  Displaymode
-  Target database
-  Target table
-  Insert query
-  Stream parameters
-  Search
-  Xslt transformation
-  Channel template
-  Target database
-  Target table
-  Insert query
-  Update query
-  Xslt transformation

Link between transformations

-  Set auto link
-  Set manual link
-  Remove connection

Source

-  Delete source
-  Copy source

Filter

-  Delete filter
-  View original
-  Preview filtered

Preview window



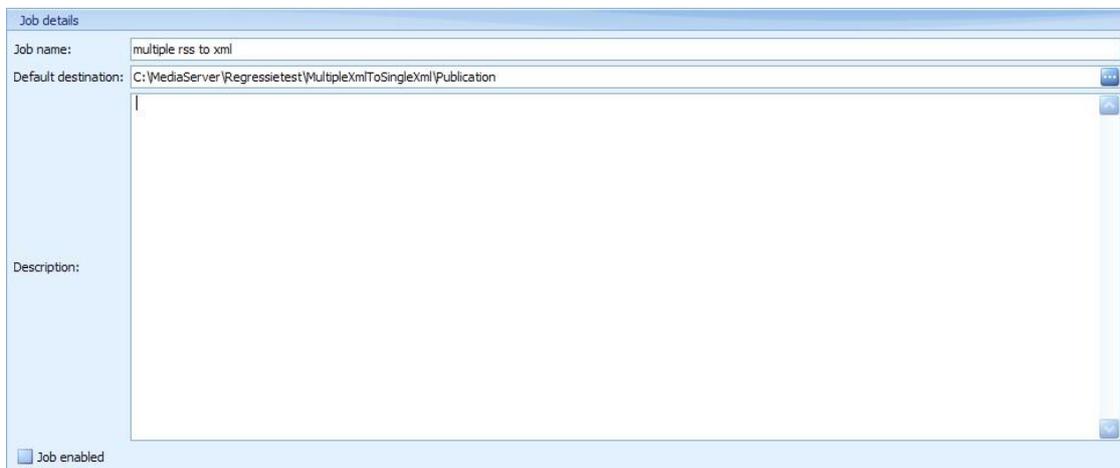
Edit



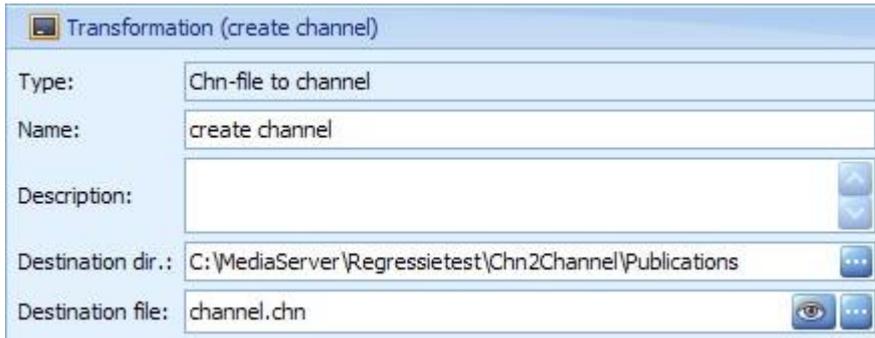
This context menu belongs to many different text-edit fields. There are no tooltips, but the options go without saying.

Edit panels

Job details



Transformation



Transformation (create channel)

Type: Chn-file to channel

Name: create channel

Description:

Destination dir.: C:\MediaServer\Regressietest\Chn2Channel\Publications

Destination file: channel.chn

Use the ellipsis button  to select a path.

Use  to view the file in an editor that belongs to its file type.

Triggers

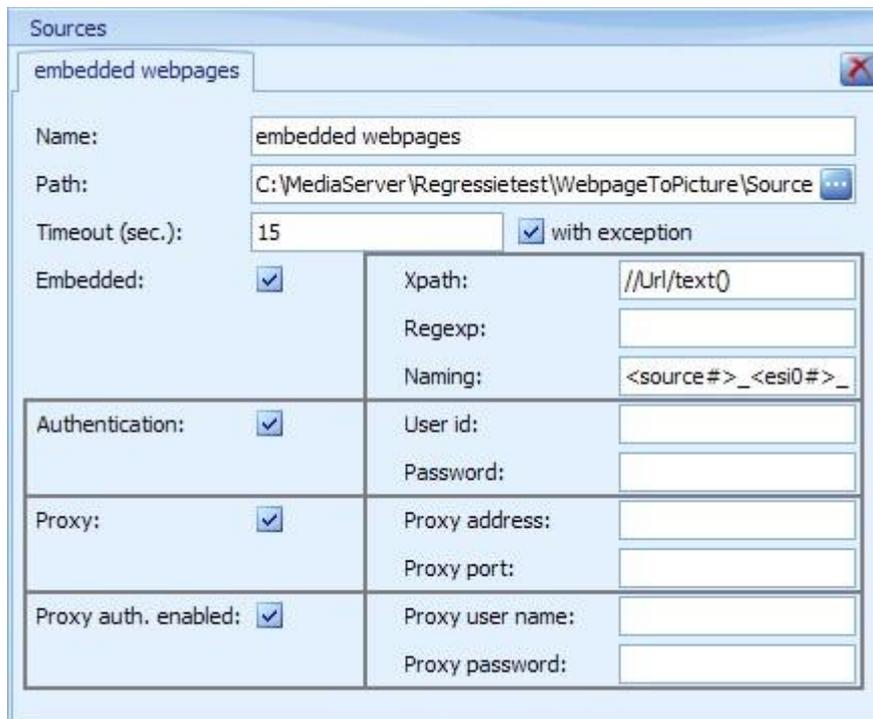


Triggers

Polling interval: 0:01:00

Dependent transformation: <none>

Sources



Sources

embedded webpages

Name: embedded webpages

Path: C:\MediaServer\Regressietest\WebpageToPicture\Source

Timeout (sec.): 15 with exception

Embedded:

Xpath:	//Url/text()
Regexp:	
Naming:	<source#>_<esi0#>_

Authentication: <input checked="" type="checkbox"/>	User id:
	Password:

Proxy: <input checked="" type="checkbox"/>	Proxy address:
	Proxy port:

Proxy auth. enabled: <input checked="" type="checkbox"/>	Proxy user name:
	Proxy password:

Timeout

When using the timeout function, you will be able to prevent the job from taking too long to read a source. The default settings is configured in the application settings (*Source read timeout default*).

With exception

Depending on what kind of transformation you are using, this field is not always available.

Through this option, you will be able to tell the media server what to do when the timeout has been reached. When With exception is selected (default) an exception will occur and reading the source will be considered as failed.

When this option is disabled and the timeout is reached, reading the source will be terminated without any exception whether it completed or not. It will be considered successful.

This could prove useful for certain websites that would never acquire a status of being read completely. In that case, the timeout should always be adjusted to a time where the page would be read completely.

Embedded

By ticking this property, you indicate that the specified file isn't the actual source, but a container that contains the actual source. So-called *embedded source items*. These can be combined with the *Xpath* or *Regexp* properties (see below).

With embedded source items, you could have a conversion of a container file or a couple of conversions of embedded source items. A transformation with embedded sources exists out of a combination of these conversions.

The conversion of the container file consists of copying the source file where the addresses of the embedded source items are replaced with the addresses of the transformed embedded source items. The name of the transformed container file is set by the *Destination file* setting in the transformation properties. Filters do not apply to this part of the transformation.

The conversion of the embedded source items consists of applying the actual transformation including the defined filters. The property *Naming* defines the naming of the resulting files.

All destination files (container and embedded source items) will be written in the *Destination directory*.

Xpath

A valid xpath expression to be applied to the source. When evaluating this expression should return a collection of embedded sources (a list of URLs).

An example of an xpath expression is: `//url/text()`

Please note: Using an xpath expression requires a valid XML source.

Regexp

A valid regular expression that will be applied to each line of the source file. Evaluating this expression should return the embedded source (a list of URLs) that match the expression. An example of a regular expression is: `^*BIMAGE \"(.*)\".*`

Please note: Using Regexp requires that the source is a readable text file, and an embedded source needs to be recognizable by a regular expression.

Xpath and Regexp are only available if Embedded sources is checked. Only one type of expression can be used at the same time.

Naming

The names of the embedded source items is based on the *Naming* field.

If the *Naming* field is empty, the original file name is used for the destination file.



If this field has a value, this name is used for the destination file. The value needs to have placeholders to result in unique file names. These placeholders are available:

- <source>
Is replaced by the name of the container, this is the full path of the original file.
- <source#>

Index of the found file. Useful for sources with wildcards.

- <esi>

Is replaced by the file name (no path) of the embedded source item.

- <esi#>

Index of the found file in the container file.

- <#>

Index of the found file, counted across all container files.

Instead of #, 0# or 00# can also be used to add leading zeros.

Please note

It can occur that authentication will not work properly on some websites due to configurations in IE or recent Windows updates. In such cases you could apply the following workaround.

Incorporate the username and password in the URL as follows: [http\(s\)://user:password@www.myurl.com](http(s)://user:password@www.myurl.com)

Attention: The login information is now visible to anyone with access to the application. Do not use this workaround if it violates security protocol.

Next, it could be possible to add two new values in the registry-key below:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet  
Explorer\MAIN\FeatureControl\FEATURE_HTTP_USERNAME_PASSWORD_DISABLE]  
"iexplore.exe"=dword:00000000  
"explorer.exe"=dword:00000000
```

Comment

Gray fields will only become available if they're enabled in the application settings.

Database-source

If the source would be a database, other fields will be available. Depending on the database type, these fields can differ.



Sources

zo'n ouderwets access-bestandje

Name: zo'n ouderwets access-bestandje

Timeout (sec.): 15

Database type: Microsoft Access (.mdb)

Database: C:\MediaServer\Regressietest\DatabaseToXml\Sources\access.mdt

Password:

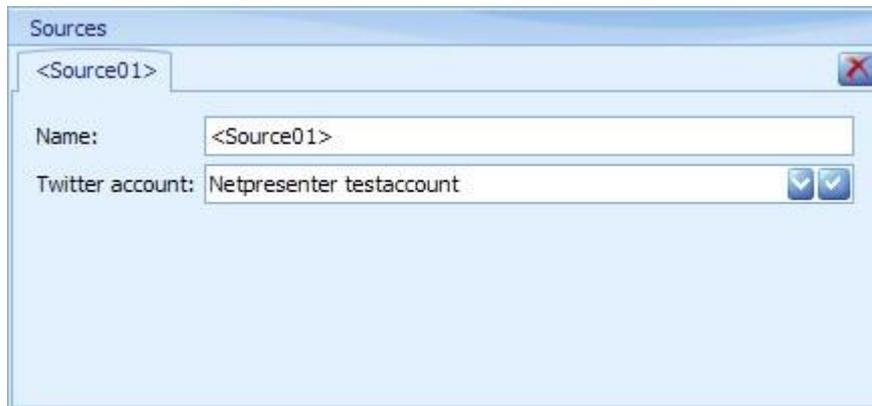
Workgroupfile:

User id:

Password:

Twitter or Facebook source

When a transformation uses Twitter or Facebook, it uses the pre-defined account for either transformation (see settings).

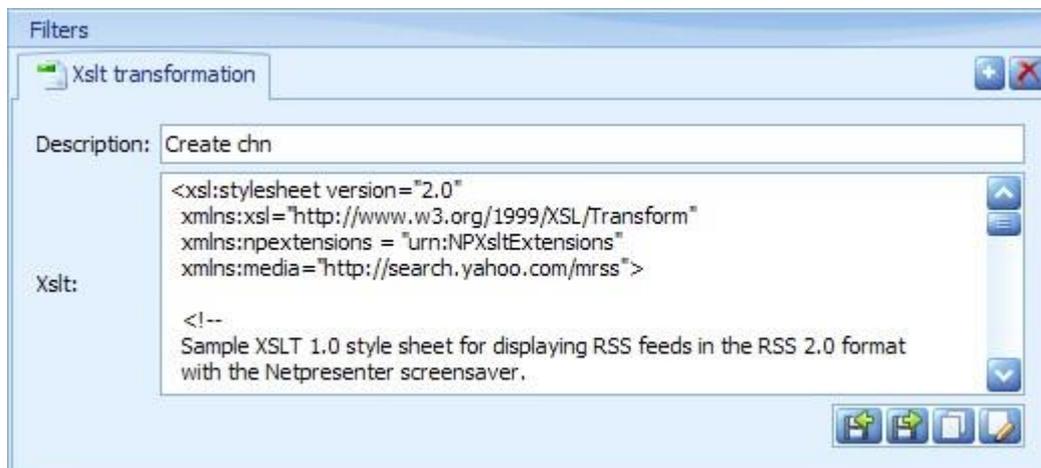


Filters

The filter properties differ for every filter type. You could come across the following fields: checkbox, color picker, directory picker, dropdown box, dropdown List, file picker, numeric up/down, memo, password and text.

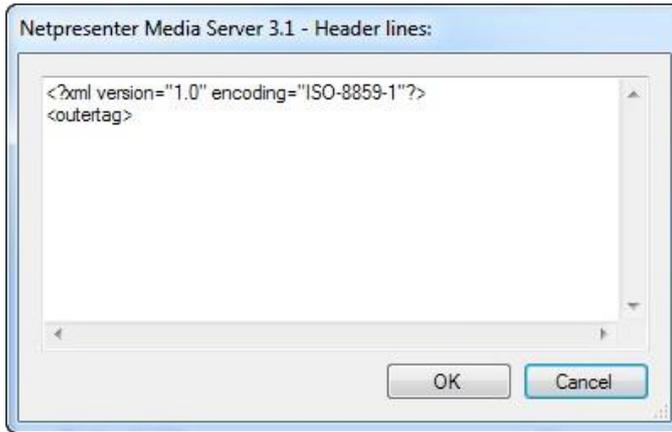
Every property has a tooltip which you can hover over for more information.

Example of text- and memo-field:



Use these buttons  to:

- Import content from a file
- Export content to a file
- Copy content to Windows Clipboard
- Edit content with a floating text editor



Comment

When there are multiple filters, the order of execution will be from left to right (in the tree view from top to bottom).

Mandatory filters

Some transformations require one or more mandatory filters. These filters are added automatically with default values, when the transformation is created.

Tip:

To reset the default values, remove the filter and add it again.

MEDIA SERVER SERVICE

The Media Server Service is added as a Windows service. Some characteristics of a Windows service are:

- No user interface;
- Uses a defined user account;
- Can be automatically started at Windows startup

The Media Server service must be active to run the jobs that are configured within the Media Server Configuration Tool.

Media Server Service tasks:

- Determine which jobs need to
- Schedule, Start (run) and stop transformations
- Provide feedback in Log files

Scheduler

The Media Server uses an internal *scheduler*. This scheduler creates and maintains a schedule for all tasks (Transformations) from all jobs that need to be executed periodically (based on a timer). When it's time for a transformation to run, the scheduler starts this process and will recreate its schedule when it is completed.



This only applies to transformations that have a Polling Interval configured. Without a Polling Interval, these Transformations need to be activated manually.

Alter job detection

The scheduler periodically checks the job configurations for changes. When a change is noticed, the scheduler will be reloaded.

Mandatory and dependent transformations

When a transformation needs to start, all dependent Transformations will follow automatically. The dependent transformation will only start when its former transformation is completed.

A transformation cannot be restarted when it is still being executed through the scheduler.

LOGGING

The Media Server Configuration Tool is able to provide log files either through starting the tool or testing & running jobs.

There are 2 available log records: Event log and Result log. While the Event log provides information about the application and service (starting, stopping and fatal exceptions), the Result log notes the process of jobs.

These logs can be consulted within the Configuration tool.