



# Axiell Ingest 7.6

## Axiell ALM Netherlands B.V.

---

Copyright © 2021 Axiell ALM Netherlands B.V.® All rights reserved.  
Adlib® is a product of Axiell ALM Netherlands B.V.®

The information in this document is subject to change without notice and should not be construed as a commitment by Axiell ALM Netherlands B.V. Axiell assumes no responsibility for any errors that may appear in this document. The software described in this document is furnished under a licence and may be used or copied only in accordance with the terms of such a licence. While making every effort to ensure the accuracy of this document, products are continually being improved.

As a result of continuous improvements, later versions of the products may vary from those described here. Under no circumstances may this document be regarded as a part of any contractual obligation to supply software, or as a definitive product description.

---

# Contents

<b>Introduction .....</b>	<b>1</b>
<b>Using Axiell Ingest .....</b>	<b>3</b>
<b>Setting up Axiell Ingest .....</b>	<b>7</b>
1.1 The Setup tab .....	8
1.2 The Ingest sources node .....	11
1.3 The Ingest source tab for a selected source.....	11
1.4 The Mappings node .....	14
1.5 The Mapping tab for a selected field mapping .....	15
1.6 Transcoders per selected source .....	19
<b>Media metadata viewer .....</b>	<b>23</b>



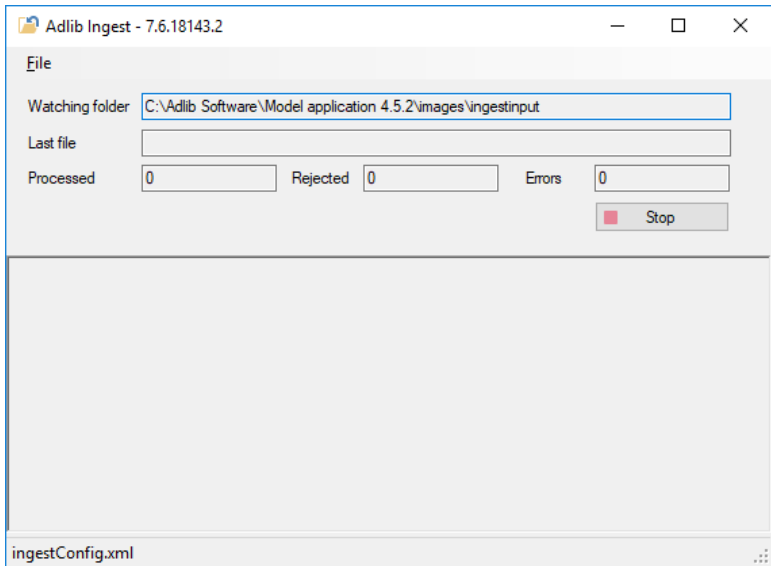
# Introduction

Axiell Ingest is an easy to use tool with which you can automatically create database records for Axiell Collections or Adlib for Windows, containing metadata of files. After having started the program, the user simply needs to copy a selection of files (e.g. documents of different sorts, image files, MP3 files (currently the only supported audio file type, from version 7.5.18073.1), Word, Excel or PowerPoint documents (these three supported from version 7.6.18227.1) or other file types) to a so-called hot folder or instead drag-and-drop one or more files from within Windows Explorer onto the program, to automatically have a database record created per file, in which a selection of metadata of the file and possibly a link to the file itself will be included. Which file types and metadata can be recorded this way, is determined completely by the setup of the program via the *ingestConfig.xml* (or similarly named) configuration file which probably resides in the same folder as the *AdlibIngest2.exe* executable. When a record is being created and populated with metadata, the processed file will be moved or copied to a preset destination folder, during which the file may or may not get a new and unique name (in the form of a Global Unique Identifier).  
Axiell Ingest only works for SQL databases.



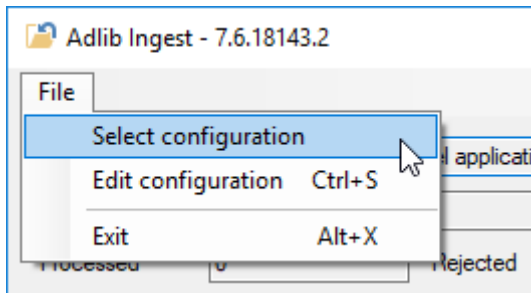
# Using Axiell Ingest

1. If a desktop shortcut to Axiell Ingest has been created for you, double-click it to start the program. Otherwise, look up the Windows folder containing the Axiell Ingest software and double-click the *AdlibIngest2.exe* file.



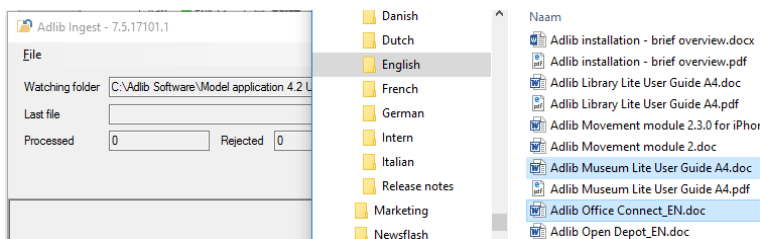
If the *ingestConfig.xml* configuration file has already been set up properly (see next chapter) and is located in the folder from which you started Axiell Ingest, then the *Watching folder* (the “hot” folder) property will be filled in for you.

However, the *ingestConfig.xml* configuration file doesn’t need to be in the *AdlibIngest2.exe* folder per se and doesn’t need to be named this way either: if *ingestConfig.xml* can’t be found by Axiell Ingest, you’ll see an error message in the lower part of the Axiell Ingest window and you’ll still have to select a proper configuration file. (If you were to close Axiell Ingest now, it would create a basic *ingestConfig.xml* configuration file for you, which would still need to be completed by you manually though.) Use the *File > Select configuration* option to select the desired config file if it is located in a different folder or if it has a different name. This functionality also allows you to have different configuration files for different applications, while requiring only a single Adlib Ingest folder.



All fields and boxes in this window are read-only. The bottom part of the window displays a log when you're processing files.

2. Now there are two ways to feed Axiell Ingest the files which you'd like to be processed:
  - a. Position the *Axiell Ingest* window next to Windows Explorer. In Windows Explorer then open the folder which contains the files you want to process (this can be any folder) and select all desired files (by **ctrl**+clicking or **shift**+clicking for example). Subsequently, click the selection, hold the mouse button down and drag the selection into the lower half of the *Axiell Ingest* window. When a plus sign appears underneath your mouse cursor, let go of the mouse button and the relevant files will be processed. The original files won't be moved nor deleted, but they will be copied to a destination folder as set up in the configuration file.



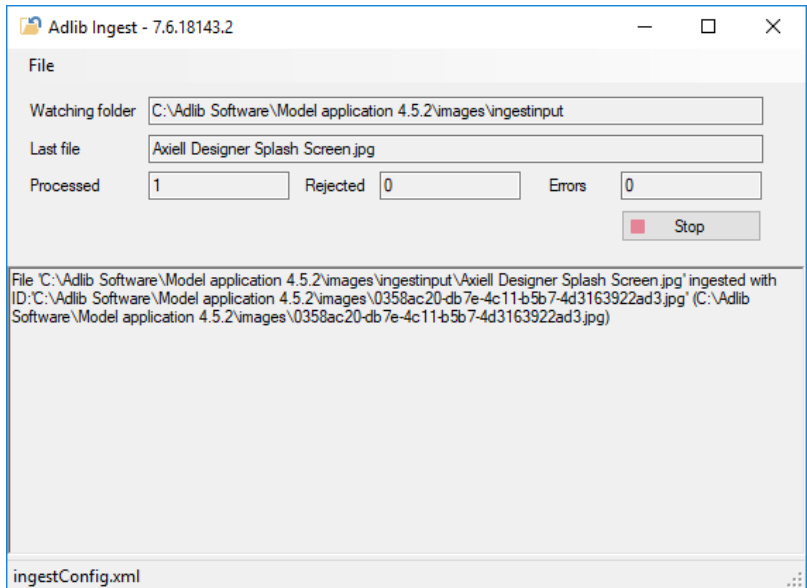
- b. The second way is to copy the desired files to the *Watching folder* in Windows Explorer. This is the hot folder set up in the configuration file for Axiell Ingest, specifically for this purpose. Simply copy and paste the desired files to this folder while Axiell Ingest is running, and that's it. (Files already present in the hot folder before you started Axiell Ingest, won't be processed.)



Note that any files you paste in this folder will be moved to the destination folder also set up in the configuration file. So if you'd like the original files to remain in their original folder as well, make sure you copy and paste, not cut and paste, the files to the hot folder.

If during or after an ingest operation errors occur, you can block further operation by clicking the *Stop* button in the Axiell Ingest window. This allows you to fix your configuration before trying again by clicking the *Start* button.

3. While the files are being processed, a report of the proceedings is displayed in the *Axiell Ingest* window. Any errors will show up as well.



A log of everything you do and have done in previous sessions will be saved in a log file called *ingest.log* by default, once you close the program. You can find this file in the folder containing *AdlibIngest2.exe* and you can open it as a text file in Windows Notepad or some other text editor.

4. The relevant files have now been moved or copied (depending on what you did in step 2) to the destination folder. You should always leave the files in this folder, because they are now referenced from within database records! You can look up the new

## *Using Axiell Ingest*

records in your Axiell Collections or Adlib for Windows application to see if they were processed correctly. In which database the new records were created, depends on how Axiell Ingest was set up by means of the *ingestConfig.xml* configuration file.

5. When you're done with Axiell Ingest, close the program via *File > Exit*, shortcut **Alt+X** or shortcut **Alt+F4**. The hot folder will no longer be active until you restart Axiell Ingest.

## Setting up Axiell Ingest

There is no installation procedure for Axiell Ingest. A Windows folder containing the Axiell Ingest files plus a correctly set up *ingestConfig.xml* (or similarly named) configuration file in the same folder is all you need. You can edit the XML in this file directly via a text editor like Notepad++ or MS Visual Studio or edit the settings via the user-friendly interface in Axiell Ingest itself. We'll discuss the latter option only.

From within an active *Axiell Ingest* window, choose *File > Edit configuration* or press **Ctrl+S** to open the *Axiell Ingest Setup* window. This is just a user-friendly shell around the *ingestConfig.xml* configuration file. Every setting you make or edit in here will be saved in the configuration file as soon as you close the *Axiell Ingest Setup* window. Some experience with Axiell Designer is recommended before you start editing this configuration: you probably need to look up the names of databases and fields.

The XML elements from the configuration file are represented here in a similar tree view in the left window pane. Click a + sign in front of a node to fold it out. Select a node of interest to see its properties in the window pane on the right. If a property has a white entry box you can type directly into it; if it is grey with a ... button behind it, then click that button to search for the relevant path, file or field via a dialog; if it is grey with a downward pointing arrow on the right side of the box, just click the field to open a drop-down list from which you can select a fixed value.

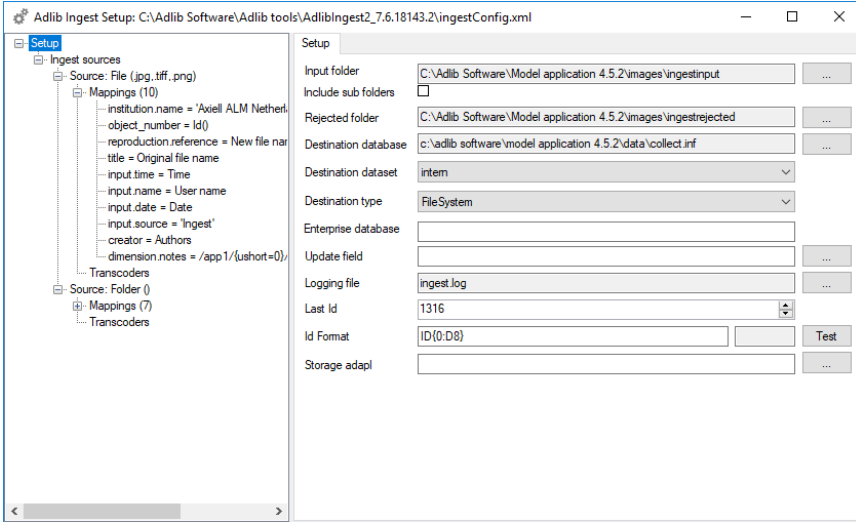
Once you click the ... button behind a field property, the *Select a field from <database>* window opens with an unsorted fields list. The best way to find the field you are looking for is to guess and type a part of the field name in the entry box to limit the fields list to all field names containing the string you entered.

Select a field from collect

object_name.notes	dimension.notes	location_history.notes	disposal.notes
material.notes	inscription.notes	location_check.notes	documentation.
technique.notes	exhibition.notes	valuation.notes	part_of.notes
frame.notes	research.notes	acquisition.price.notes	parts.notes
title.notes	condition.notes	acquisition.notes	related_object.r
production.date.notes	completeness.notes	owner_hist.notes	production.note

<  >

## 1.1 The Setup tab



The *Setup* tab offers general options that apply to all ingest sources:

- **Input folder** must be filled with the path to an existing folder, namely the hot folder (aka *Watching folder*) to which files can be copied to be ingested. Use the ... button to search for the path. A path may contain spaces, for example: C:\Axiell Software\Axiell Ingest\input files
- **Include sub folders** must be marked if sub folders of the hot folder must be watched too. This typically applies to an enterprise setup where the hot folder often contains a sub folder for every branch of the enterprise.
- **Rejected folder** is optional and may contain a valid path to which rejected files will be copied. If you leave the property empty then any rejected files will just be deleted from the input folder. Ingested files will be rejected if their file type is not one of the supported file types for ingestion, as set up in this configuration file or if certain conditions aren't being met. Example: C:\Axiell Software\Axiell Ingest\rejected
- **Destination database** must contain a valid path to the database in which records for the ingested files must be created. The database doesn't necessarily need to be a media or photo database, you can also ingest into a database that has a linked field to a media or photo database. You would typically ingest an

image file directly into a media database if you just want to create media records and figure out to which object records they must be linked later on. On the other hand, if you'd like to create some base object record linking to a new media record, for each ingested image file, you'd choose `collect.inf` as your destination database. You cannot provide a dataset name in this property. Example in which `collect` is the name of the database definition file: `C:\Axiell Software\Model application 4.2\data\collect.inf`

- **Destination dataset** must contain the name of the dataset in the database in which records for the ingested files must be created, if the database has datasets. Example: `intern`
- **Destination type** is to allow for exceptional cases (*Rosetta*) where the destination is not in the file system. In most cases this option must be set to `File System`.  
*Rosetta* (ExLibris) allows you to specify the particular folder structure for ingested files, that it requires. It involves creating a csv-like file and a folder containing the images, after which that package must be moved to a folder for *Rosetta* to process it. <no further information yet>
- **Enterprise database** must contain a valid, but partially variable path to the database in which records for the ingested files must be created if you have an enterprise setup in which each branch has its own version of the `.inf` file in a sub folder. To represent all possible sub folders at the same level, use the `{0}` parameter instead, for example:  
`G:\Axiell applications\museums\{0}\model\data\media.inf`  
Since the `{0}` parameter can only be substituted by Axiell Ingest if files are dropped into the relevant input sub folder directly, not if they are dropped into an Axiell Ingest window, you must still also fill in the *Destination database* property with some fixed database path as a fallback option.
- **Update field** (optionally) contains the name of a uniquely indexed field which you'd like to use as a so-called update field for all sources. Using an update field means that if Axiell Ingest finds an existing record in the destination database in which this field contains the same value as Axiell Ingest is about to import, then no new record will be created for the file to be ingested, but the existing record will be updated with the metadata from the ingested file instead.

## Setting up Axiell Ingest

- **Logging file** (optionally) contains the path to a logging file which will log the activities and any errors of Axiell Ingest. New activities will be appended to the existing log file. If no log file has been specified here, then the activities of the program won't be logged.
- **Last Id** is a mandatory property that must contain the last ID (an integer value) used. This value is updated automatically by the Ingest program - so users working with Axiell Ingest must have write access to the folder containing the configuration file that we are editing here - and is increased every time a new file is ingested. Typically you'd use the ID to write it to a uniquely indexed field in the target record, like the `object_number` field to identify the record (more about this later).
- **Id Format** is an optional .NET format string which can be used to lay out the new record identifier. You can read about .NET format strings [here](#) and [here](#). For example: `ID{0:D8}AX`, formats the integer value from *Last Id* by placing the string "ID" in front of it and "AX" behind it, while formatting the number as an 8-digit number with precursor zeros. So if *Last Id* would be 1246, then after formatting, the resulting string becomes `ID00001246AX`. Here the format string can only have the single `{0}` parameter and this parameter is mandatory in the format string. Use the *Test* button behind this property to see if the format string actually does what it should do.
- **Storage adap1** (optionally) specifies the path to a storage adap1 (.bin) which will be executed 3 times, first after the metadata-to-field mapping has been applied but before the new record is created and saved in Collections or Adlib (&1=27), then again before storage of the record (&1=11) and once more after storage of the record (&1=14). Even though you use field names in the mapping, you must use the associated field tags in the adap1: Axiell Ingest will match field names to tags. You can use an adap1 if metadata must be edited in some way before it is imported into a database field.  
Further note that the default storage adap1 for the target database, as set up in the relevant *.inf* file, will never be executed, while the default storage adap1 for any linked database in which data is forced by the ingested data will be executed normally. (This option is available from version 7.2.14308.3.)
- **Start** (in 7.6.18227 still missing from the GUI) must be checked if you'd like the ingest process (the watching of the hot folder and the ability to actually ingest files into Collections or Adlib) to

become active as soon as you open Axiell Ingest: typically you would want this if you only have a single *ingestConfig.xml* file. The *Start/Stop* button in the Axiell Ingest window will read *Stop* initially, so you can stop the already started process if you'd like. On the other hand, if you have multiple *ingestConfig.xml* files (for different applications), then it might be a good idea to deselect this option (set the `<Start>` option underneath the `<ingestConfiguration>` node in *ingestConfig.xml* to `false`) so that users must first select the proper configuration file before activating the ingest process via the *Start/Stop* button which will display *Start* after opening Axiell Ingest.

- `<ExcludeList>` (in 7.6.18227 still missing from the GUI) may contain a list of files which are never allowed to be ingested, `web.config` and `adlibweb.xml` for example. This list currently has to be set in the *ingestConfig.xml* directly, underneath the `<ingestConfiguration>` node. Enclose each file name in a `<string>` node, like so:

```
<ExcludeList>
  <string>web.config</string>
  <string>adlibweb.xml</string>
</ExcludeList>
```

---

## 1.2 The Ingest sources node

In the tree view on the left, right-click the *Ingest sources* node and select *Add ingest source* in the pop-up menu to add a new *Source* node. A source groups certain file types to be treated identically. Each ingest source has its own definition and field mapping. This allows you to map a different metadata per file type group, such as documents and images. You must specify at least one source.

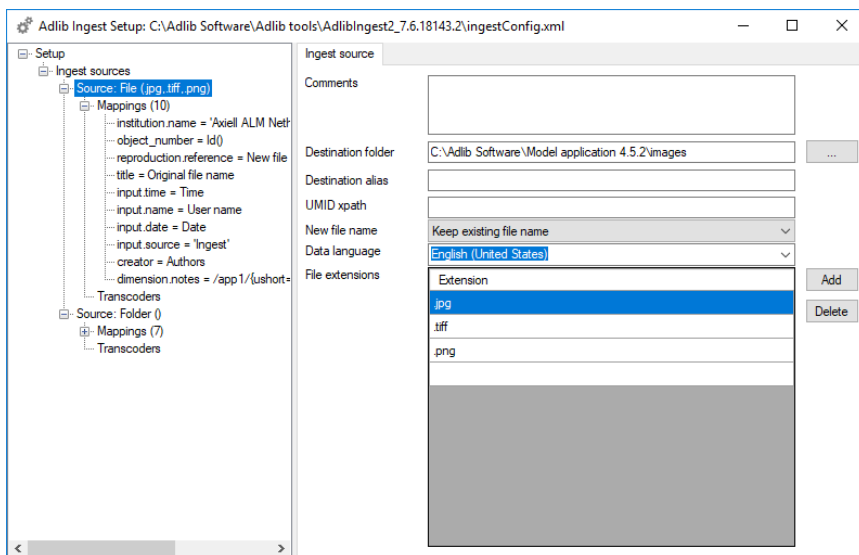
To delete a *Source* node and everything in it, right-click the relevant *Source* node and select *Delete ingest source* in the pop-up menu.

---

## 1.3 The Ingest source tab for a selected source

An ingest source is a group of one or more file types which must be treated in the same way. There may be more than one ingest source, each with its own specific mapping. A source can also have so-called transcoders with their own mapping: transcoders allow you to have Axiell Ingest automatically create altered copies of ingested images, as far as image size, file type, destination folder and mapped image field are concerned (see chapter 1.6).

## Setting up Axiell Ingest



The options on the *Ingest source* properties tab are as follows:

- **Comments** (optional) allows you to describe what this source and its settings are for, for later reference.
- **Destination folder** must contain a path to the folder to which ingested files must be moved or copied (and possibly renamed) or a URL to a service which deals with the ingested image. The folder must exist and it must be accessible by the ingest program. This setting is mandatory for each ingest source. The destination folder must point to the same location as the *Retrieval path* option for the field in which the digital reference will be stored, otherwise the linked documents can't be opened from within their record in Collections or Adlib. Note that the destination folder is a single location, even for enterprise solutions.
- **Destination alias** <no information yet>
- **Alternative update field** (optionally) contains the name of a uniquely indexed field which you'd like to use as a so-called update field for this particular source, instead of the global *Update field* you might have set (option available from 7.6.18221.1). Using an update field means that if Axiell Ingest finds an existing record in the destination database in which this field contains the same value as Axiell Ingest is about to import, then no new record will be created for the file to be ingested, but the existing record will be updated with the metadata from the ingested file instead.



- **UMID xpath** can be used to specify an XPath to retrieve an ID from the response XML of a service to which an ingested file is uploaded. This ID can be mapped to a target field using the (*Source*) *External id* field mapping option. <no further information yet>
- **New file name** must contain either the value *Keep existing file name* (causing an ingested file to keep its original name if it doesn't exist in the target folder yet, otherwise it'll be rejected) or *Generate a new GUID* (to generate a new and unique name, in the form of a Global Unique Identifier) for each ingested file whilst keeping the old extension.
- **Data language** can be set to a target data language (which translates to an [LCID language identifier](#) in *ingestConfig.xml*) for all multilingual fields in the current ingest source, although a different target language can still be set per field mapping. (If neither *Data language* option has been set, then US English (LCID 1033) will be the implicit target language for multilingual fields.)
- **File extensions** must contain a list of file extensions that can be processed in this *File* source (not for *Folder* sources). The file extension is not restricted to three characters, but must start with a `.` character. Add or delete rows to the list using the buttons next to it.  
At least one extension must be present in the configuration to run Axiell Ingest. You group similar file types in a single source because you associate a metatag-to-field mapping with it that is shared among these file types. Therefore you would group extensions like *.pdf*, *.doc*, *.ppt*, *.xsl* etc. because they can all be regarded as documents of which you'd like to register identical metadata fields. Different image file types can be listed in their own source.  
Note that upon ingestion, currently no file type validation is taking place, to validate that the file is containing data that actually matches the file extension, but normally that is not a problem of course. However, file type validation is planned for a future release of Axiell Ingest.
- **Type** (in 7.6.18227 still missing from the GUI) must contain the type of the ingested items for this source. This can only contain the setting *File* or *Folder*. While the option is not present, the setting defaults to the *File* type. With *File*, only files with the specified File extension will be read from the input folder. With *Folder* on the other hand, all files (regardless of their extension) will be read from the input folder.

## Setting up Axiell Ingest

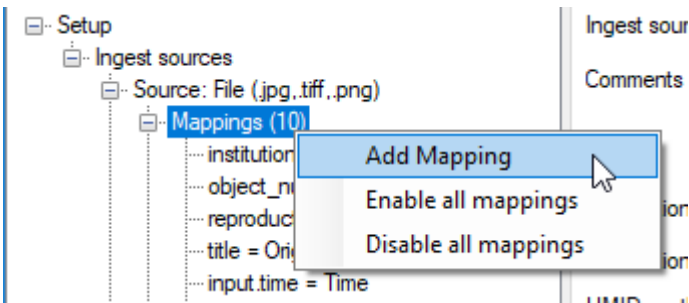
(After closing the *Axiell Ingest Setup* window, you can set the type to `Folder` by editing the *ingestConfig.xml* file directly, in the `<IngestSources<IngestSource><Type>Folder</Type>...` node. Do remember to reload the configuration settings into Axiell Ingest by choosing *File > Select configuration.*)

---

### 1.4 The Mappings node

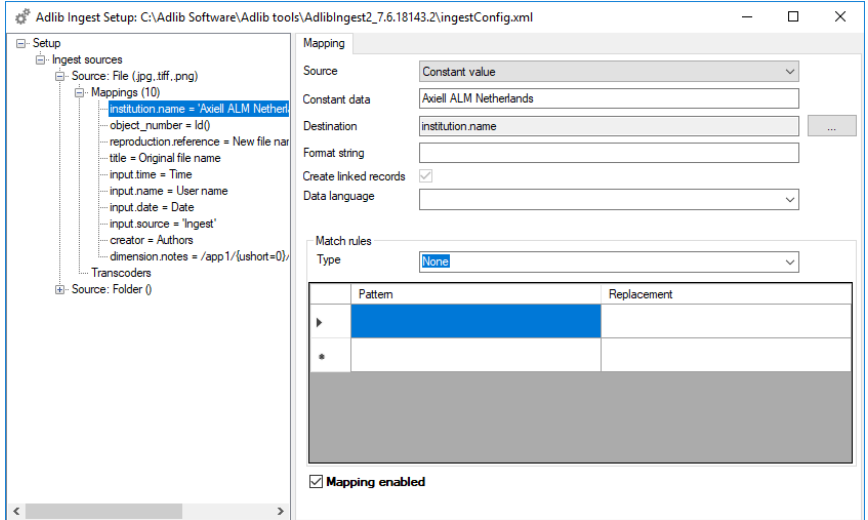
The *Mappings* node must contain one or more metatag-to-field mappings, to allow Axiell Ingest to import metadata into the appropriate record fields.

In the tree view on the left right-click the *Mappings* node and select *Add mapping* in the pop-up menu to add a *Mapping* sub node. To enable disabled mappings for a source or to disable enabled mappings, right-click the relevant *Mappings* node and make your choice from the pop-up menu.



Right-click a single mapped field and select *Delete mapping* to delete the single mapped field or click *Add constant* to add a constant value to a single field mapping. The latter allows you to have one or more constant values filled in in destination fields of your choice, only if the current metatag-to-field mapping succeeds: per *Constant mapping* you can enter *Constant data* and a *Destination* field. A constant mapping can be deleted via its right-click pop-up menu.

## 1.5 The Mapping tab for a selected field mapping



A mapping defines how a single metadata element is converted to a field in the database. At least one mapping, that of the Id tag is mandatory. An ingest source is not valid without a mapping of the Id field. There is no limit to the number of mappings that can be defined. Each mapping must be expressed in at least a *Destination* (the desired and appropriate target field in a Collections or Adlib database) and a *Source* value (a Windows metadata tag, some system variable or a constant).

- **Source** (mandatory) specifies the data source of the mapped field. The supported options are:
  - *Meta data* - indicates that a metadata element from the file should be stored in the field. If this *Source* option is used then you must also select a *Metadata tag* from the list below, to specify one of sixteen Windows metadata tags of which the contents should be stored in the destination field.
    - **Metadata Tag** must contain the name of the file metadata element of which the contents needs to be ingested. There's the following limited list of Windows metadata elements you can try, which might only work for *.jpg* image files though: Application Name, Author, Camera Manufacturer, Camera Model,

## Setting up Axiell Ingest

Comment, Copyright, Date Taken, Format, Keywords, Rating, Subject, Title, Location, File Size, Width in pixels, Height in pixels. If these are not sufficient or do not work, use the Meta data query *Source* instead of Metadata Tag.

- Meta data query - indicates that a metadata element from the file should be stored in the field. If this *Source* value is used then the *Tag path* property below must also be filled in to specify the general metadata tag of which the contents should be stored in the destination field.
  - **Tag path** must contain the exact source field path of the metadata you'd like to ingest. This type of metadata is retrieved from the file itself through metadata query paths. To find out what metadata elements are present in a particular file type, use the Axiell *ShowImageMetaData* tool which comes with the Ingest module, to view all metadata of a file (see next chapter). Example: `/app1/{ushort=0}/{ushort=34665}/{ushort=40962}`
- Generated Id - the new sequential number for the ingested item, optionally formatted using the *Id Format* option on the *Setup* tab. The content of this field is automatically generated using the *Last Id* property (also on the *Setup* tab). There must be exactly one *Generated Id* mapping per source of the *File* type. Typically you would map this id to a field like `object_number`.
- New path of the file - generates the path of the new location of the ingested file (without the file name itself). This is generated automatically after the file has been moved or copied to the destination folder.
- New full path and file name - generates the path of the new location of the ingested file plus the (possibly new) file name itself. This is generated automatically after the file has been moved or copied to the destination folder.
- New file name - the name of the ingested (new) file. This is generated automatically after the file has been moved or copied to the destination folder. Typically you would map this metadata source to a reference field, like `reproduction.reference` in

collect.inf. For the resulting link in the Collections or Adlib record to be correct, make sure a correct storage path for the field has been set in Axiell Designer: the target folder for ingestion should be the same as the folder targeted by the storage path.

- `Original file name` – yields the original file name of the ingested file (before it possibly got renamed).
- `Current Date` – produces the date of the ingestion, in ISO format (yyyy-mm-dd).
- `Current Time` – the time of the ingestion (in hh:mm:ss format).
- `Current User Name` – the name of the user (process) that ran the ingest procedure.
- `Folder name` – the full path to ingest input folder.
- `Subfolder name` – the name of the ingest input subfolder into which the file to be ingested was dropped.
- `Constant value` – indicates that the constant value as specified in the *Constant data* property below must be entered in the destination field when a file is being ingested. You could use it to fill the `institution.name` field with the name of your institution, for example.
  - **Constant data** must contain the fixed data for a `Constant value` *Source* in this mapping. This element is only valid for a `Constant value`.
- `External id` – the ID retrieved using the source *UMID xpath* option.
- `XPath query` – allows you to specify an XPath query with which to retrieve data from the XML returned by the service (if you've set a URL to a service as the source *Destination* folder), to map it to a target field.
  - **XPath** must contain an XPath query <no example yet>
- `Subfolder name and original file name` - the name of the ingest input subfolder into which the file to be ingested was dropped, followed by a slash and the original file name. (In 7.6.18227, only the original file name is retrieved.)

## Setting up Axiell Ingest

- **Destination** (mandatory) must contain the (English) field name of the field into which the source metadata must be ingested.
- **Format string** (optional) is an optional .NET format string which can be used to lay out the mapped metadata. You can read about .NET format strings [here](#) and [here](#). For example: `{0:D8}`, formats an integer metadata value by formatting the number as an 8-digit number with precursor zeros. Here the format string can only have the single `{0}` parameter and this parameter is mandatory in the format string. (This doesn't work in 7.6.18227.1.)
- **Create linked records** must be marked for linked fields if you'd like a new linked record to be created automatically for each non-existing value ingested into this field. If you leave it deselected for linked fields, then non-existing values can't be ingested into this field.
- **Data language** can be set to a data language (which translates to an [LCID language identifier](#) in *ingestConfig.xml*) for multilingual fields to indicate a target language different from the *Data language* set for the current ingest source. Leave it empty for unilingual fields. (If neither *Data language* option has been set, then US English (LCID 1033) will be the implicit target language for multilingual fields.)
- **Match rules** (optional) is meant to set a condition which metadata must meet before it is ingested. This is done by specifying a regular expression: metadata must match this regular expression to be ingested. You can choose to ignore (skip) the metadata for the current field only or to ignore the entire ingested file (so that no record will be created) if there's no match. You could use this to skip the ingest of images of which the resolution is too high, for example. First set the desired *Type*, then the *Pattern*:
  - **Type** can be either `None` (default, leaves any rule inactive), *Ignore field on unsuccessful match* (the current field mapping will be ignored if there's no match with the provided pattern) or *Reject file on unsuccessful match* (the currently processed file won't be ingested if there's no match with the provided pattern).
  - **Pattern** must contain one or more [regular expressions](#) to which the metadata from the current source field must be compared. To skip the field or file if the current metadata value consists of 4 or more digits (the width of an image for ex-

ample) you would include the following pattern: `[0-9]{0,3}`  
- **Replacement** (optionally) may contain a string per pattern that should replace the matched pattern in the metadata once ingested.

(In 7.6.18227, match rules do not work correctly.)

- **Mapping enabled** allows you to switch the current mapping on or off, so you don't have to remove it per se if you don't want to use this mapping right now.

---

## 1.6 Transcoders per selected source

Transcoders (configured transcoding settings really, internally available in Ingest, no external services or WebAPI required) offer the possibility to automatically create one or more altered copies of ingested images: thumbnails or other lower resolution images than the original, possibly even with a different file type. First you must specify one or more transcoders in a separate XML configuration file. Then you can refer to one or more of these transcoders per ingest source. Transcoded copies will always be renamed to a GUID (Global Unique Identifier).

### ■ Creating the `imagertranscoder.xml` file

The file specifications for all transcoders must be made in a separate `imagertranscoder.xml` file in the same folder as the currently edited `ingestConfig.xml` file. You can't do this in the *Axiell Ingest Setup* window. Here's an example of two specified transcoders:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfImageTranscoder
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ImageTranscoder>
    <Size>1200</Size>
    <ImageType>Jpeg</ImageType>
    <Name>Image1200pxJpeg</Name>
  </ImageTranscoder>
  <ImageTranscoder>
    <Size>1600</Size>
    <ImageType>Jpeg</ImageType>
    <Name>Image1600pxJpeg</Name>
  </ImageTranscoder>
</ArrayOfImageTranscoder>
```

`<Name>` must a descriptive name to id the relevant transcoder. Also specify the file extension with which the copy must be saved (this may

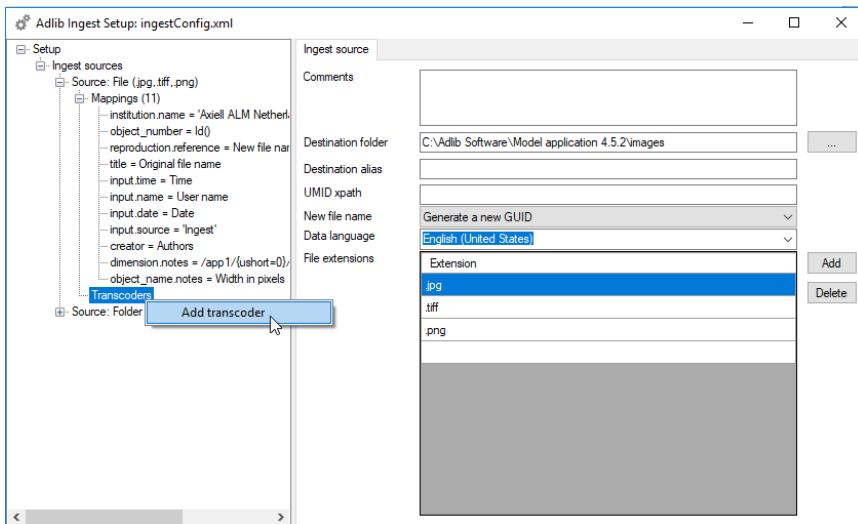
## Setting up Axiell Ingest

be different from the original file extension) and the image width in pixels to scale the copy to.

Any target image file format you can think of can be used. For a full list of supported image formats, click [here](#).

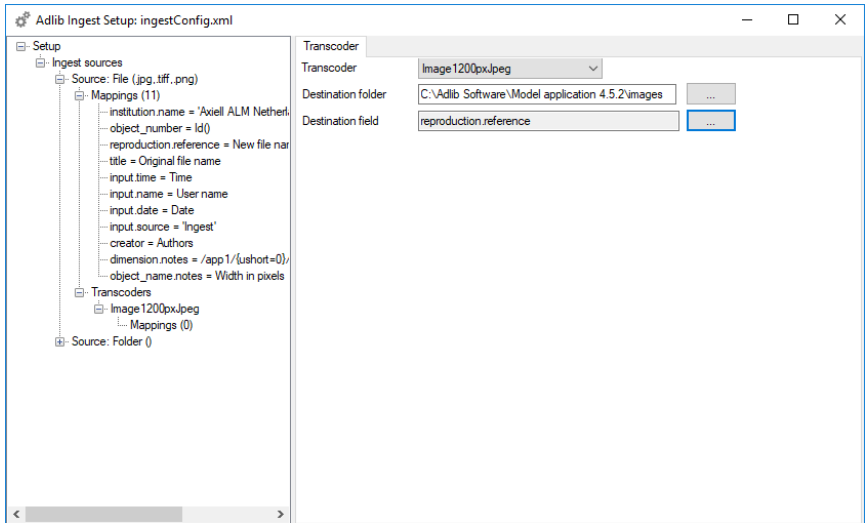
### ■ Setting up a source image transcoder

In the *Axiell Ingest Setup* window, right-click the *Transcoders* node underneath a source in the tree view on the left and select *Add transcoder* in the pop-up menu.



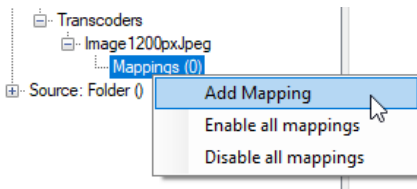
Then select the blank node underneath and fill in the properties of this transcoder reference.



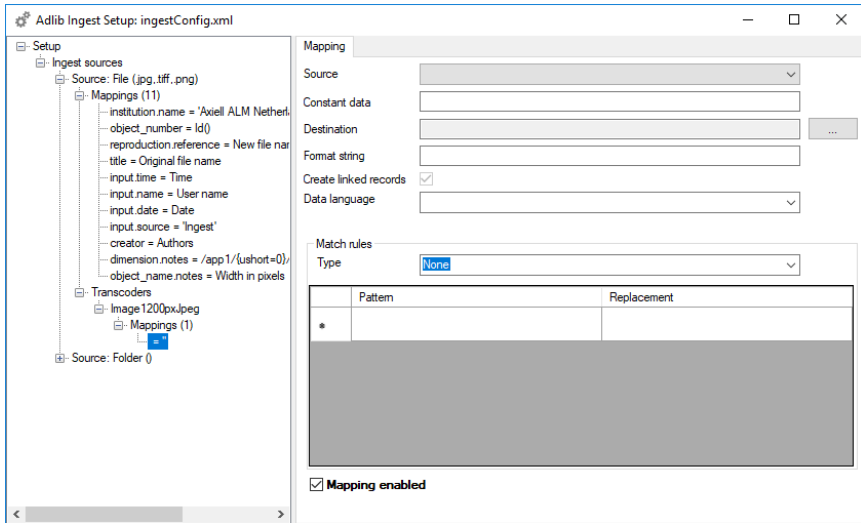


- **Transcoder** allows you to pick one of the transcoders specified in the `imagertranscoder.xml` file.
- **Destination folder** allows you to set the same or a different storage folder for the transcoded (and automatically renamed) image files.
- **Destination field** must be set to an English field name in which the reference to the transcoded image must be stored. (In 7.6.18227.1 you can't use a linked field that you've already used in the current source mapping, like the `reproduction.reference` field in our examples.)

Optionally, right-click the *Mappings* node underneath the transcoder reference you just specified and select *Add Mapping* in the pop-up menu to add one or more *Constant value* or *XPath query* mappings. Select the blank node underneath *Mappings* to set the properties of the mapping.



## Setting up Axiell Ingest

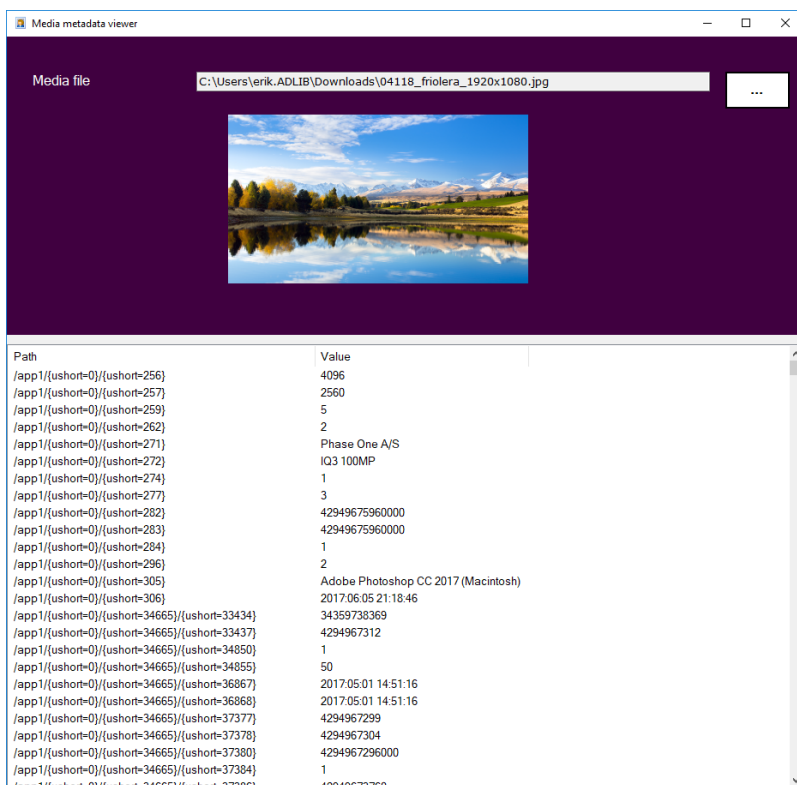


Any node in the tree view can be deleted by right-clicking it and selecting the appropriate option in the pop-up menu.

# Media metadata viewer

Accompanying the Ingest module is the *Media metadata viewer*. You can use it to find so-called metadata queries for image, audio and document metadata you'd like to map in the Ingest configuration. From Adlib Ingest 7.5, supported MS Office document file types are *.docx* (Word documents stored in the Office Open XML format), *.xlsx* (Excel documents) and *.pptx* (Powerpoint documents) (The latter three supported from 7.6.18229.1.) (PDF files are not supported yet.) From Adlib Ingest 7.5.18073.1, *.mp3* audio files are supported as well.

1. Simply double-click *ShowImageMetadata.exe* to open it.
2. Then click the ... button to find an example of the file type you wish to explore: some *.jpg* image for instance.



## Media metadata viewer

3. As soon as you've opened the image, its metadata will be listed. The metadata queries you're looking for (replacing actual metadata field names) are listed in the *Path* column, with their current (for this file) value displayed in the column next to it. Some paths have a readable term at the end while others don't, but then you can sometimes tell from the value what it probably represents. In all cases you can just right-click the desired path and select *Copy* in the pop-up menu to copy the entire path to the clipboard.
4. Now paste the path in the desired *Meta data query* field mapping *Tag path* property.

Mapping	
Source	Meta data query
Tag path	/app1/{ushort=0}/{ushort=34665}/{ushort=40962}

5. Repeat this for all desired metadata query field mappings.