



AGETOR®

JOBS GUI
Configuration Guide

Content

1	Preface	3
1.1	Audience	3
1.2	Typographic conventions	3
1.3	Requirements.....	3
1.4	Acknowledgements.....	3
1.5	Additional information.....	3
2	Why configuration may be needed.....	3
3	Configuring L&T property replication to the JOBS (DDS) system.....	4
3.1	Turning on property replication for jobs.....	4
3.2	Controlling what properties are replicated.....	4
3.3	Auto-replication.....	4
3.3.1	Turning auto-replication on/off	4
3.3.2	Auto-replication fetch frequency and fetch pattern	5
3.4	Example: adding info and customizing JOBS GUI presentation	5
3.4.1	Attaching info during transformations	5
3.4.2	Turning on enrichment and specifying properties to replicate.....	6
3.4.3	Configuring display of the additional job properties now added	6
4	Configuring the JOBS GUI.....	8
4.1	Configuring search fields	8
4.1.1	Search fields with leadtext	8
4.1.2	Generic search fields.....	10
4.2	Configuring Job Result columns	10
4.3	Configuring Grouping options	11
5	References.....	12

1 Preface

This document describes how to configure the JOBS GUI to customize the presentation of information. This includes information on how to enrich job information with data from the Log & Trace system.

1.1 Audience

This guide is written for developers and system managers who want to interact with the AGETOR® framework in order to enrich and customize the presentation of delivery jobs.

1.2 Typographic conventions

- Text marked with *italics* refer to other publications or definitions of concepts
- Text marked like `AbstractClassName`, `identifier`, `myCoolFunction()` and `cmd` refer to executable commands, identifiers or literal code excerpts



Issues requiring your special attention are presented like this!

1.3 Requirements

You need an AGETOR® installation with AXT components (AXT Basic and DDS server).

Some initial knowledge about the AGETOR® systems are assumed when reading this guide. It is assumed that the *AXT JOBS GUI Guide* is understood.

1.4 Acknowledgements

The software described in this document includes software developed by the Apache Software Foundation (<http://www.apache.org>).

1.5 Additional information

General information on AGETOR®/AXT may be found in relevant guides on the AGETOR® download center (www.agetor.com).

Please consult the AXT JOBS GUI Guide for information on the JOBS GUI.

2 Why configuration may be needed

The AGETOR® system provides very flexible tools for exchanging and transforming business data. The data (documents) may be of any conceivable type. This means that the information content is not known per se and what information items that may be interesting is very much dependent on the actual usage and installation.

AGETOR® support extracting and storing such relevant information from documents or calculations during the transformation flow and document exchange.

For such data to be presented in a user friendly manner for jobs executed it is necessary to configure

- What information items should be searchable
- What information should be presented in the JOBS result lists

- What information should be extracted from the Log & Trace system (collected during processing) and shown in the JOBS GUI

3 Configuring L&T property replication to the JOBS (DDS) system

The data that may be fetched and persisted with job information is document and trace properties that relate to the specific job. I.e. if properties were added to the document represented by the job, then these can be made searchable and presented in the JOBS GUI.

This section describes how the replication of such properties from the trace system to the DDS jobs system is configured.

3.1 Turning on property replication for jobs

The value of the media property `enrichFromTrace` (boolean) determines if it is possible to replicate data from the trace system to the DDS server. The value is deduced in the same way as other DDS Jobs properties; i.e. by media configuration with the possibility to override the setting on each individual job. For example the default setting for all media may be overridden by settings for each individual media which again may be overridden on the specific job.

3.2 Controlling what properties are replicated

Two additional job properties `enrichDocumentProperties` and `enrichTraceProperties` control what properties are actually fetched to the job system. The two properties may contain a semi-colon (;) separated list of properties or simply an asterisk (*) to fetch all properties.

If the default media states the following:

```
enrichFromTrace="false"  
enrichDocumentProperties="ref"  
enrichTraceProperties=""
```

No properties are fetched from the trace system. If, however the `enrichFromTrace` property is set to true in general or for a specific job, the `ref` document property would be fetched.

3.3 Auto-replication

The DDS service will automatically fetch document and trace properties for terminated jobs if they have `enrichFromTrace=true` set. It does so by periodically fetching trace properties for all jobs that terminated (and will be kept) if they have the `enrichFromTrace` property set to true.

3.3.1 Turning auto-replication on/off

The DDS service will automatically fetch document and trace properties for terminated jobs if they have `enrichFromTrace=true` set. In some cases you might not want the auto-enrichment to run since it consumes some system resources. Rather you might only want to fetch the additional information explicitly from the JOBS GUI using the refresh button (see JOBS GUI User Guide).

If so, you can change the default behaviour to *not* auto-fetch information by setting the following property in the `docdeliver.properties` file:

```
docdeliver.traceinfo.enrichment.automaticfetch=false
```

3.3.2 Auto-replication fetch frequency and fetch pattern

The auto-replication will run with a fixed period given by the DDS property

```
docdeliver.traceinfo.enrichment.fetchinterval=60
```

That is, if any jobs terminated during the last 60 seconds and they are due to have their properties fetched; a request will be made to update their jobs file.

It is possible to specify how old jobs must be before their trace properties are fetched and also how many times they will be refreshed. This is controlled by the property:

```
docdeliver.traceinfo.enrichment.fetchpattern=30
```

This property states that trace properties are fetched once after 30 seconds. It is possible to give a list of such delayed fetches:

```
docdeliver.traceinfo.enrichment.fetchpattern=30;120;600
```

stating that trace properties are fetched after 30 seconds, 2 minutes and 10 minutes. Note that properties will be fetched all three times!

3.4 Example: adding info and customizing JOBS GUI presentation

3.4.1 Attaching info during transformations

During AXT transformation document and trace properties are added using the `AttachInfo` constructs. These will assign name-value pairs to document or the trace at large and store them in the Log & Trace database.

An example construct from an AXT transformation could look like this:

```
<attach-info scope="trace">
  <param name="type" value="Enrichment example"/>
</attach-info>
<attach-info scope="document">
  <param name="ref" value="XML order #{refno}"/>
</attach-info>
```

This assigns a document property to the current document and a trace property to the corresponding trace. If the document is subsequently delivered to a destination using the DDS service, it will be possible to have the attached property replicated to the job information for display in the JOBS GUI.

The DDS delivery below will store the document in the file system. A description is added to the job and will be visible in the JOBS GUI by default. Also the job will generate checkpoints in the L&T system with texts reflecting the success or failure to deliver.

```
<filter class="dk.bording.axt.client.docdeliver.DocDeliverOutlet" output="input">
  <param name="media" value="file"/>
  <param name="serie" value="myserie"/>
  <param name="mask" value="order_{refno}_%s.txt"/>
  <param name="description" value="File system peristance of XML order"/>
  <param name="folder" value="data/test/output2"/>
  <param name="enrichFromTrace" value="true"/>
  <param name="checkpointTextOnError" value="Error occurred delivering order file {refno}"/>
  <param name="checkpointTextOnSuccess" value="ok"/>
</filter>
```

However, the document and trace properties attached will not appear in the JOBS GUI by default. To have properties from the trace system show there, we need to

- 1) Turn on replication of the trace properties for the job
- 2) Configure a display column that shows the replicated information in the job list of the JOBS GUI

3.4.2 Turning on enrichment and specifying properties to replicate

Turning on enrichment of the properties was explained previously and we use the explicit method of turning it on for the specific job by setting `enrichFromTrace=true`.

Since the default configuration states that the `ref` property should be replicated, that will copy the `ref` property to our job automatically. But we also want the `type` property of the trace replicated! Either we may modify the overall setting of what properties are replicated or we have to specify it for this particular job. We choose to change to generic setting in the `defaultmedia.xml` configuration file to:

```
enrichFromTrace="false"
enrichDocumentProperties="ref"
enrichTraceProperties="type"
```

We could also have stated:

```
enrichFromTrace="false"
enrichDocumentProperties="*"
enrichTraceProperties="*"
```

This would copy all trace and document properties for all jobs that had enrichment turned on (assuming they do not individually override the properties to fetch). Normally this will be a bad solution since we are communicating all properties even those we do not display in the JOBS GUI.

3.4.3 Configuring display of the additional job properties now added

Once the needed data have been configured for replication we need to configure display, and perhaps search fields, for the additional job information now present in the job files.

In section 0 the configuration of the JOBS GUI is detailed, and here we will only describe the configuration need to add a column showing the trace property `type` that we have enriched the job file with.

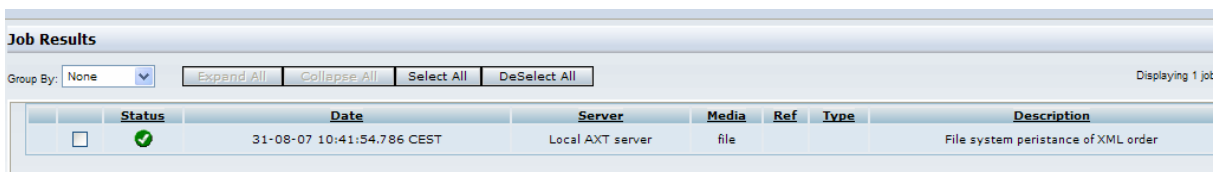
The snippet below from the `axtgui.properties` file shows the addition of a `Type` column (lines 5-7). It is stated that the 6th column should have the name `Type`. The data type of the column is string since we are dealing with plain texts. The `xpath` expression picks out the data item from the underlying job XML structure (see section 0 for details on this).

```

1. agetor.dds.gui.jobs.list.label.5=Ref
2. agetor.dds.gui.jobs.list.type.5=string
3. agetor.dds.gui.jobs.list.xpath.value.5=/job/trace/document/properties/property[@name='ref']/@value
4.
5. agetor.dds.gui.jobs.list.label.6=Type
6. agetor.dds.gui.jobs.list.type.6=string
7. agetor.dds.gui.jobs.list.xpath.value.6=/job/trace/trace/properties/property[@name='type']/@value
8.
9. agetor.dds.gui.jobs.list.label.7=Description
10. agetor.dds.gui.jobs.list.type.7=string
11. agetor.dds.gui.jobs.list.xpath.value.7=/job/@description

```

Submitting our job from the previous example and searching in the JOBS GUI will result in the following entry:

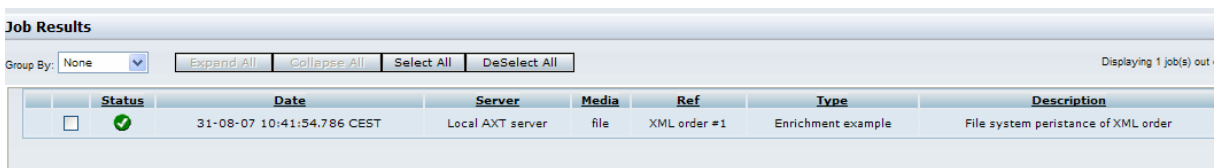


Status	Date	Server	Media	Ref	Type	Description
<input type="checkbox"/>	31-08-07 10:41:54.786 CEST	Local AXT server	file			File system persistence of XML order

Figure 1. Job result list with an additional column *Type*

Note that a Type column is now present. Both the Ref and the Type column have no data initially since the enrichment is slightly delayed by default.

After some seconds a new search will reveal that the data from the L&T system have been copied to the job and they are now displayed in our columns as expected.



Status	Date	Server	Media	Ref	Type	Description
<input type="checkbox"/>	31-08-07 10:41:54.786 CEST	Local AXT server	file	XML order #1	Enrichment example	File system persistence of XML order

Figure 2. Job result list now with the Type and Ref columns enriched by data from the L&T system

It is also possible to group data on our new columns since we have omitted explicit grouping definitions in the configuration (see section 0).

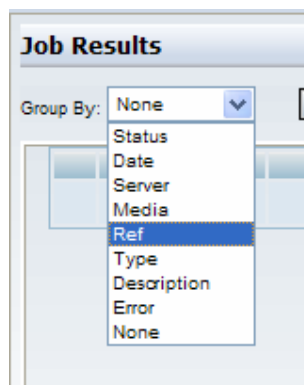


Figure 3. Grouping on new column

4 Configuring the JOBS GUI

The JOBS GUI allows customization of what search fields appear and what data are shown from the jobs in the DDS server(s).

This is necessary since the actual document and information types are very implementation dependant.

The customization is controlled by properties given in the `axtgui.properties` file located in `AGETOR_HOME/conf/properties/`.

4.1 Configuring search fields

Three properties determine the appearance of the default simple search tab. These are:

Property	Description	Values
<code>agetor.dds.gui.jobs.initialsearch</code>	The default status value to use	One of failed, succeeded, delivering, all
<code>Agetor.dds.gui.jobs.initialsearch.results</code>	The number of search results to fetch to the working set	The value to show by default for the number of results to return
<code>Agetor.dds.gui.jobs.search.results</code>	The possible values that may be chosen for the working set size (number of jobs to return on a search)	A semi-colon (;) separated list of values. E.g.: 5;10;20;50;100;200

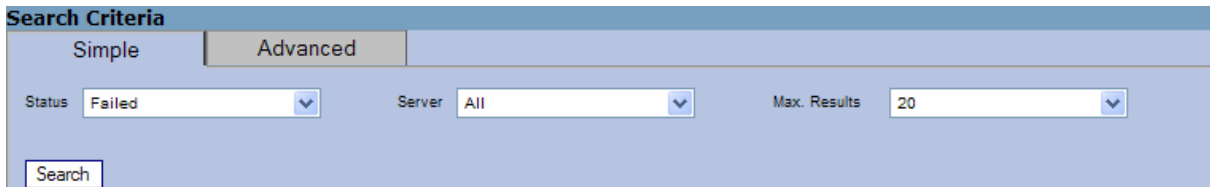


Figure 4. Simple search tab

4.1.1 Search fields with lead text

To configure search fields with a given leading text and limited search operators the following properties are used:

Property	Description	Values
<code>agetor.dds.gui.jobs.searchfield.label.<N></code>	The leading text	e.g. <i>media</i>
<code>agetor.dds.gui.jobs.searchfield.datatype.<N></code>	The type of the property.	string, long, date, boolean
<code>agetor.dds.gui.jobs.searchfield.type.<N></code>	The basic type of the property. Ordinary media (job) properties	job, trace, document

	trace properties or replicated document properties.	
agetor.dds.gui.jobs.searchfield.name.<N>	The name of the property. This is the media attribute name or the document/trace property name	e.g. ref (if a document property called ref exists) or media (to refer to the job property media)
agetor.dds.gui.jobs.searchfield.operators.<N>	A list of operators that is shown in a drop-down for the field	E.g.: like;=;<;>;<=;>=

Note that the <N> represents a number going from 1 and upwards and serve to group related properties together and order the fields in the display.

Search fields configured in this way will appear in the advanced tab in the Common properties section:

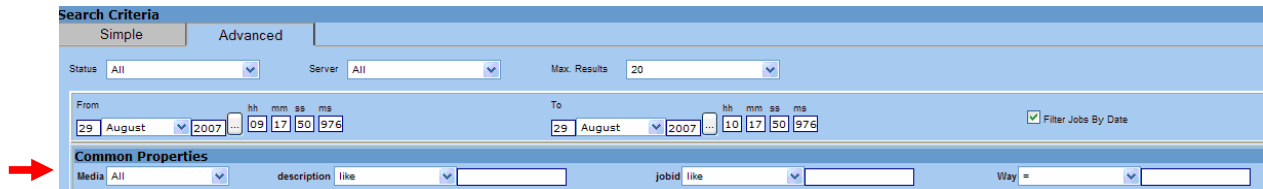


Figure 5. Common properties search fields

The above was configured with the following properties in the `axtgui.properties` file:

```
# common search default properties
agetor.dds.gui.jobs.searchfield.label.1=Media
agetor.dds.gui.jobs.searchfield.datatype.1=string
agetor.dds.gui.jobs.searchfield.type.1=job
agetor.dds.gui.jobs.searchfield.name.1=media
agetor.dds.gui.jobs.searchfield.operators.1=\=;like

agetor.dds.gui.jobs.searchfield.label.2=description
agetor.dds.gui.jobs.searchfield.datatype.2=string
agetor.dds.gui.jobs.searchfield.type.2=job
agetor.dds.gui.jobs.searchfield.name.2=description
agetor.dds.gui.jobs.searchfield.operators.2=like

agetor.dds.gui.jobs.searchfield.label.3=jobid
agetor.dds.gui.jobs.searchfield.datatype.3=string
agetor.dds.gui.jobs.searchfield.type.3=job
agetor.dds.gui.jobs.searchfield.name.3=id
agetor.dds.gui.jobs.searchfield.operators.3=like;\=

agetor.dds.gui.jobs.searchfield.label.4=Way
agetor.dds.gui.jobs.searchfield.datatype.4=string
agetor.dds.gui.jobs.searchfield.type.4=trace
agetor.dds.gui.jobs.searchfield.name.4=way
agetor.dds.gui.jobs.searchfield.operators.4=\=;like
```

To specify a limited number of legal values for a search field it is possible to list such values as in the example below. The example states that the media search field should only allow a choice between the media types *fpt*, *file*, *mail*, *axt*.

```

agetor.dds.gui.jobs.searchfield.value.1.1=
agetor.dds.gui.jobs.searchfield.value.1.2=ftp
agetor.dds.gui.jobs.searchfield.value.1.3=file
agetor.dds.gui.jobs.searchfield.value.1.4=mail
agetor.dds.gui.jobs.searchfield.value.1.5=axt

```

The blank value indicates that any value is allowed. The first value is the default to be shown in the dropdown of the search field:

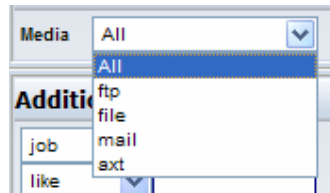


Figure 6. Default values in dropdown

4.1.2 Generic search fields

A number of generic search fields without a lead text or pre-given name allow for quite flexible search on job file properties. It may be required from time to time to search on some job properties that we don't want to appear always in the GUI as search fields.

The configuration of such flexible criteria allow for a number of such fields:

```

#additional search properties
agetor.dds.gui.jobs.searchfield.flexcriteria.types=job;document;trace
agetor.dds.gui.jobs.searchfield.flexcriteria.operators=like;=;<;>;<=;>=
agetor.dds.gui.jobs.searchfield.flexcriteria.max=5

```

This results in the following input search fields. Searching is done by specifying the basic type (job, trace and document) and entering values in the blank boxes for a) the property name and b) the property value to match.

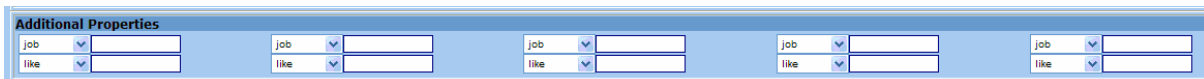


Figure 7. Generic search properties

4.2 Configuring Job Result columns

The columns to display for found jobs are configured by giving a name, a type and an Xpath expression that point to the value in the underlying jobs XML structure. This XML structure has all job properties as XML attributes to the root job tag.

All document properties are held as name/value attributes in the path `/job/trace/document/properties/property` and a value may be referred to by:

```

/job/trace/document/properties/property[@name='docref']/@value

```

This picks out the document property value of the `docref` property. Obviously this is only possible if the value is present in the job file and this depends on whether enrichment with the value has been done.

Similarly trace properties are in the path:

```
/job/trace/trace/properties/property
```

In the example below three job properties are shown (media, description and priority), one document property (ref) and two trace properties (type and sender).

```
# result column default properties

agetor.dds.gui.jobs.list.label.4=Media
agetor.dds.gui.jobs.list.type.4=string
agetor.dds.gui.jobs.list.xpath.value.4=/job/@media

agetor.dds.gui.jobs.list.label.5=Ref
agetor.dds.gui.jobs.list.type.5=string
agetor.dds.gui.jobs.list.xpath.value.5=/job/trace/document/properties/property[@name='ref']/@value

agetor.dds.gui.jobs.list.label.6=Description
agetor.dds.gui.jobs.list.type.6=string
agetor.dds.gui.jobs.list.xpath.value.6=/job/@description
.
.
.
agetor.dds.gui.jobs.list.label.9=type
agetor.dds.gui.jobs.list.type.9=string
agetor.dds.gui.jobs.list.xpath.value.9=/job/trace/trace/properties/property[@name='type']/@value

agetor.dds.gui.jobs.list.label.10=Sender
agetor.dds.gui.jobs.list.type.10=string
agetor.dds.gui.jobs.list.xpath.value.10=/job/trace/trace/properties/property[@name='sender']/@value
.
.
.
agetor.dds.gui.jobs.list.label.14=Priority
agetor.dds.gui.jobs.list.type.14=long
agetor.dds.gui.jobs.list.xpath.value.14=/job/@priority
```

Note that the priority job property is of type long in order to sort correctly on the column.

4.3 Configuring Grouping options

To state the allowed grouping options the following properties may be used:

```
agetor.dds.gui.jobs.list.grouping.columns.<N>=<M>
```

The `<N>` is an increasing number from 1 and the `<M>` refers to the number of a column previously defined.

E.g. the following configuration states that it should be possible to group by column 1, 5, 3 or 4. The final blank value indicates no grouping.

```
# groupBy columns properties
agetor.dds.gui.jobs.list.grouping.columns.1=1
agetor.dds.gui.jobs.list.grouping.columns.2=5
agetor.dds.gui.jobs.list.grouping.columns.3=3
agetor.dds.gui.jobs.list.grouping.columns.4=4
agetor.dds.gui.jobs.list.grouping.columns.5=
```

These configured columns will appear in a dropdown.



To allow grouping on *any* configured column you simply omit these property definitions.

5 References

The AGETOR guides are all available at the AGETOR® download center located at: <http://www.agetor.com/>