

# Ingestion Lab

XML with File Connector



*This lab is designed to provide some hands-on experience for designing and implementing ingestion logic, as discussed in the Ingestion lesson. We will walk through ingestion configuration for an XML file in detail.*

## Prerequisites:

- [Installing Attivio](#)
- [Quick Start Lab \(Factbook\)](#)

## Table of Contents:

- [Getting Started](#)
- [Configuring a New Connector](#)
- [Creating a New Workflow](#)
- [Using the New Workflow](#)
- [Ingesting the Data](#)
- [Exploring the Data](#)
- [Adding a Document Title \(optional\)](#)
- [Shutting Down Attivio](#)



# GETTING STARTED



# The Data Set

*We are going to use a basic XML file in this lab. You can download the file from this link:*

## [Literacy.xml](#)

*This file contains information about the male and female literacy rates for each country in the world over a range of years. Open the file and observe the structure.*

*Place this file in a convenient location; for example, **C:\attivio\input***



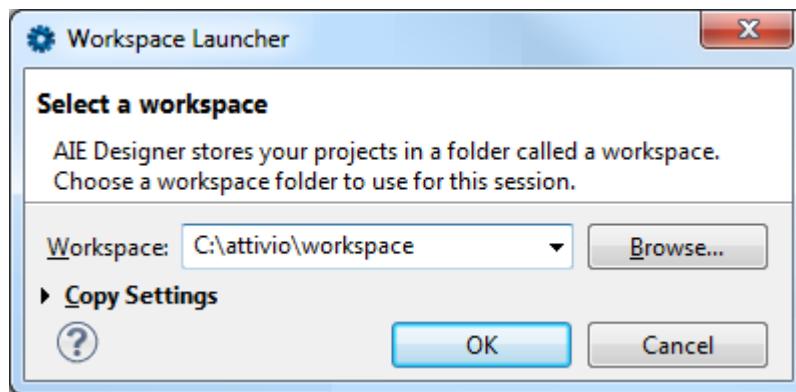
# Starting Attivio Designer

- Start the **Attivio Designer** application

*C:\attivio\install\designer\Designer*

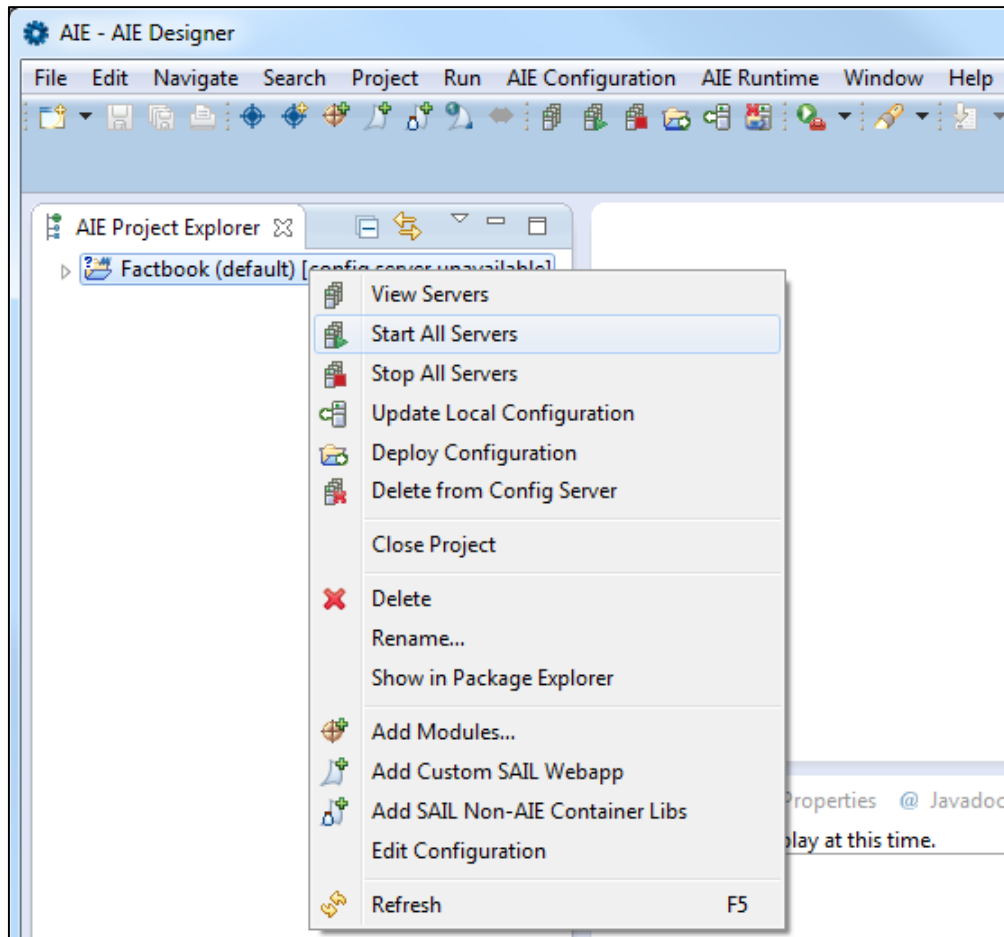
*Attivio Designer is an application intended for system architects and developers. It provides the means to easily and quickly configure, view, and edit every element of the Attivio platform.*

- If prompted, specify the workspace location



# Starting Attivio

- In the **Project Explorer** pane, right-click the **Factbook** project



- Click **Start All Servers**

*The project will need 45-90 seconds to start up.*





# **CONFIGURING A NEW CONNECTOR**

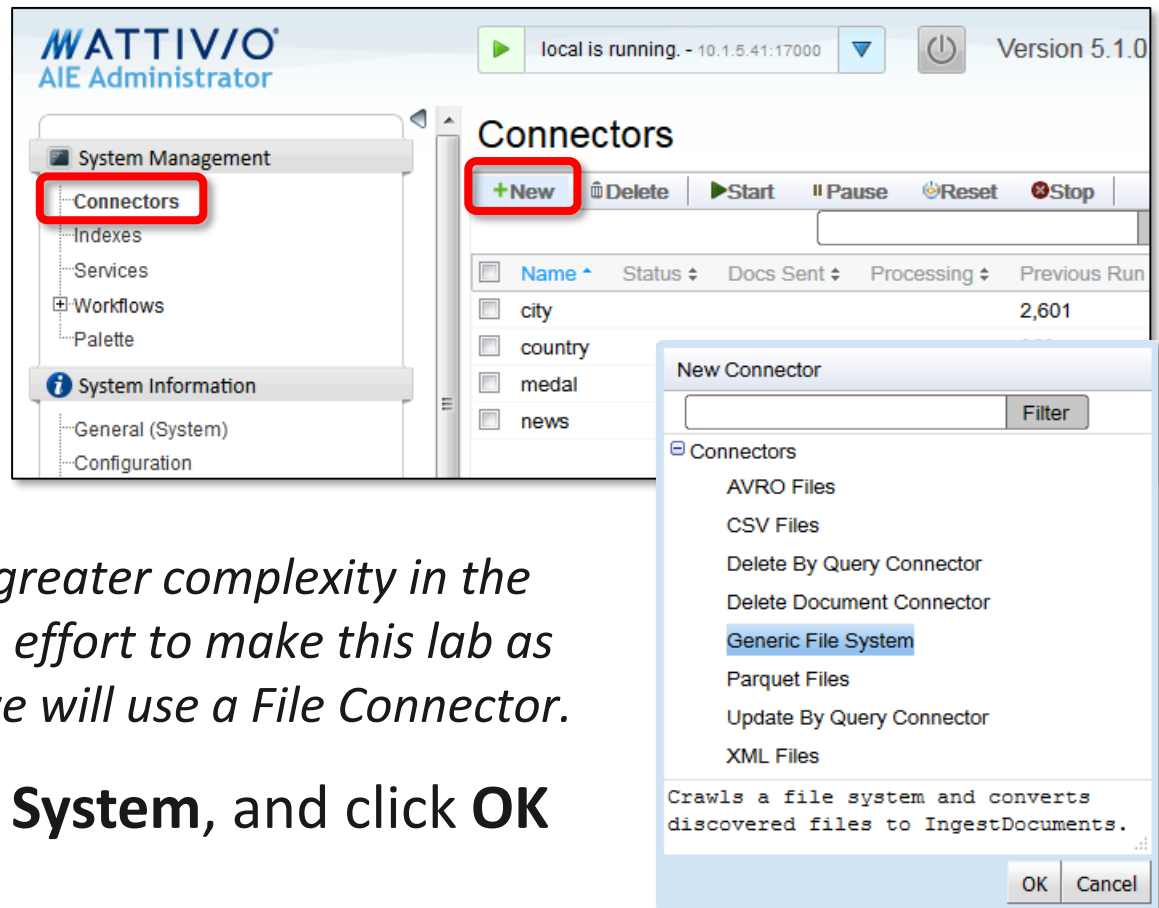


# Configuring a New Connector

- In **Admin UI**, go to the **Connectors** page
- Click **New**

*There are two viable options: we could use a specialized XML connector, or a vanilla File Connector. Because the File Connector is not specialized for XML data, using it will require greater complexity in the following workflow. In an effort to make this lab as informative as possible, we will use a File Connector.*

- Select **Generic File System**, and click **OK**





# Configuring a New Connector

- Name the connector **literacy**  
*This name can be anything, as long as it's unique*
- Enter the directory path to the **Literacy.xml** file  
*e.g., `C:\attivio\input\Literacy.xml`*
- Delete all of the existing **Include Filters**, and add one for **\*.xml**  
*The Literacy file is an XML file, so we need to crawl files with an XML extension and no others*

The screenshot shows the 'FileScanner: New Connector' configuration window. The 'Scanner' tab is selected. The following fields are highlighted with red boxes and arrows:

- \*Connector Name:** A text input field.
- \*Start Directory:** A text input field.
- Wildcard Include Filter:** A list of file extensions with delete icons. The existing filters are: \*.doc, \*.docx, \*.htm, \*.html, \*.pdf, \*.ppt, \*.pptx, \*.txt, \*.xls, \*.xlsx. A new wildcard include filter button (+) is visible at the bottom of the list.

Other visible fields include:

- Namespace Id: (empty)
- Node Set: (use default)
- File System URI: (empty)
- Follow symbolic links: true
- Maximum directory depth: -1
- Maximum File Size (MB): 5
- Minimum File Size (MB): -1
- Wildcard Exclude Filter: \$\*, \*.tmp, \*~, .\*, ~\* (with a new wildcard exclude filter button (+) at the bottom)
- Directory Listing Timeout: -1
- Document ID Prefix: (empty)
- Ingest Workflow: fileIngest

Buttons at the bottom: Save, Save & Test, Cancel.



# Configuring a New Connector

*The finished Scanner tab should look like this.*

*Note that the connector is still feeding the standard fileIngest workflow. We will change that once we create our custom workflow.*

- Click on the **Field Mappings** tab

The screenshot shows the 'FileScanner: New Connector' dialog box with the 'Field Mappings' tab selected. The 'Field Mappings' section has two input fields, each with a '+' button. The 'Static Field Values' section also has two input fields, each with a '+' button. At the bottom are 'Save', 'Save & Test', and 'Cancel' buttons. A red box highlights the 'Field Mappings' tab label.

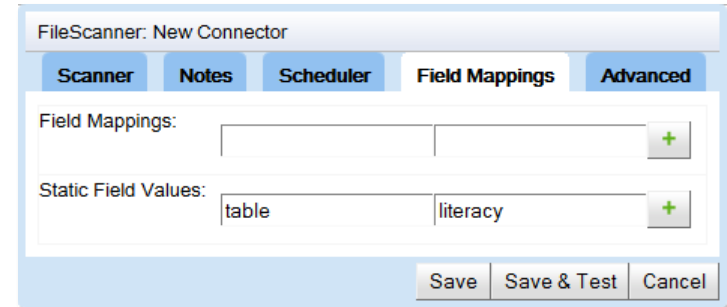
The screenshot shows the 'FileScanner: New Connector' dialog box with the 'Scanner' tab selected. The 'Ingest Workflow' field is highlighted with a red box and contains the value 'fileIngest'. A red arrow points from the 'Field Mappings' tab in the previous screenshot to this field. Other fields include 'Connector Name' (literacy), 'Start Directory' (C:\attivio\input\Literacy.xml), and various file size and depth settings. At the bottom are 'Save', 'Save & Test', and 'Cancel' buttons.



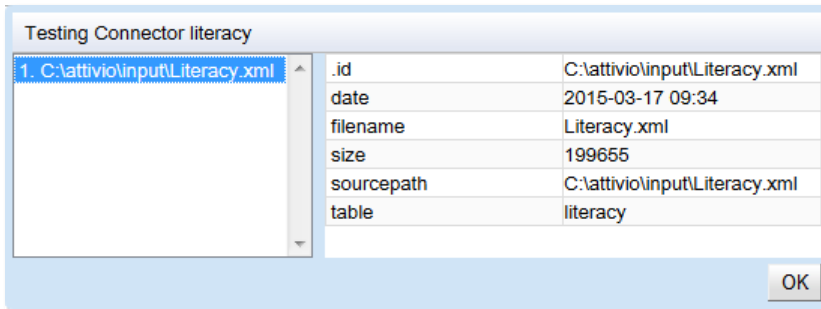
# Configuring a New Connector

- Create a static **table** field mapping with a value of **literacy**

*The finished Field Mappings tab should look like this*



- Click **Save and Test** to verify the connector's behavior



*One document should be produced. It will contain the file metadata, an **id** equivalent to the **sourcepath**, and the **table** field you specified.*

- Click **Ok**
- Save** the connector

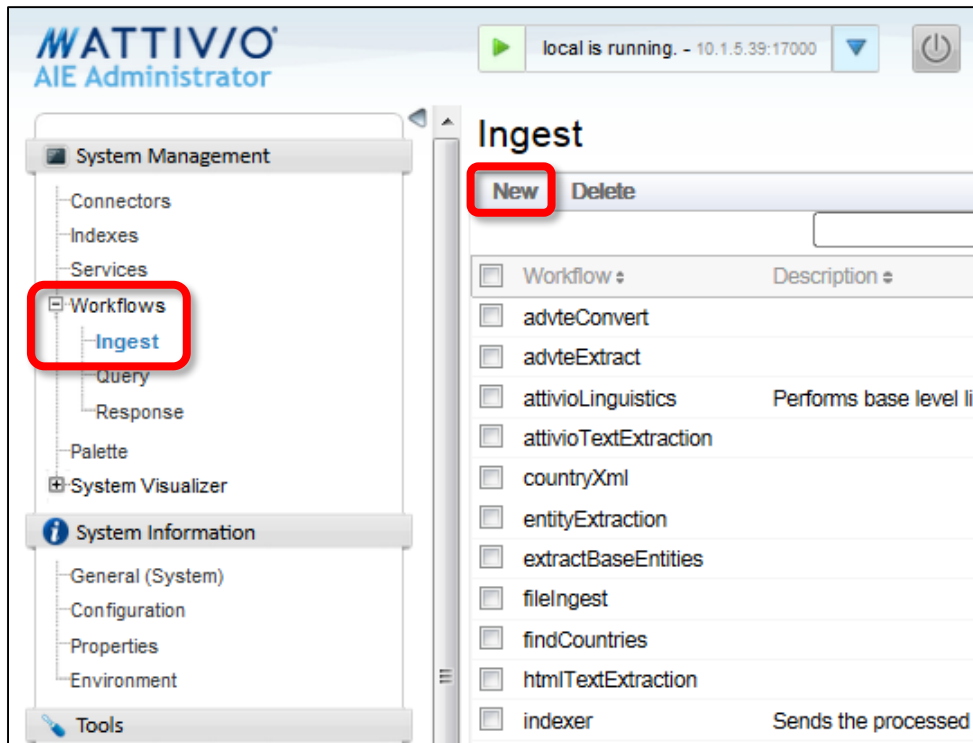


# CREATING A NEW WORKFLOW

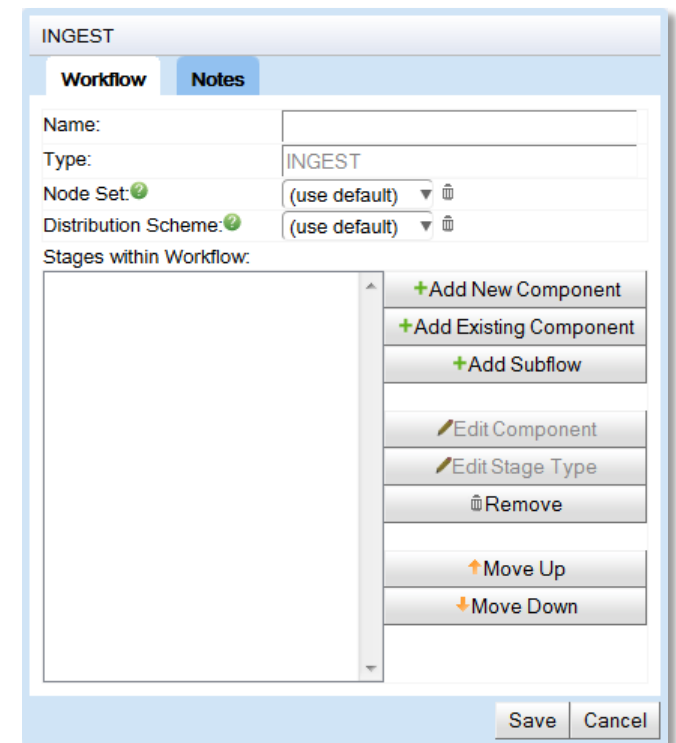


# Creating a New Workflow

- In **Admin UI**, go to the **Ingest Workflows** page
- Click **New**



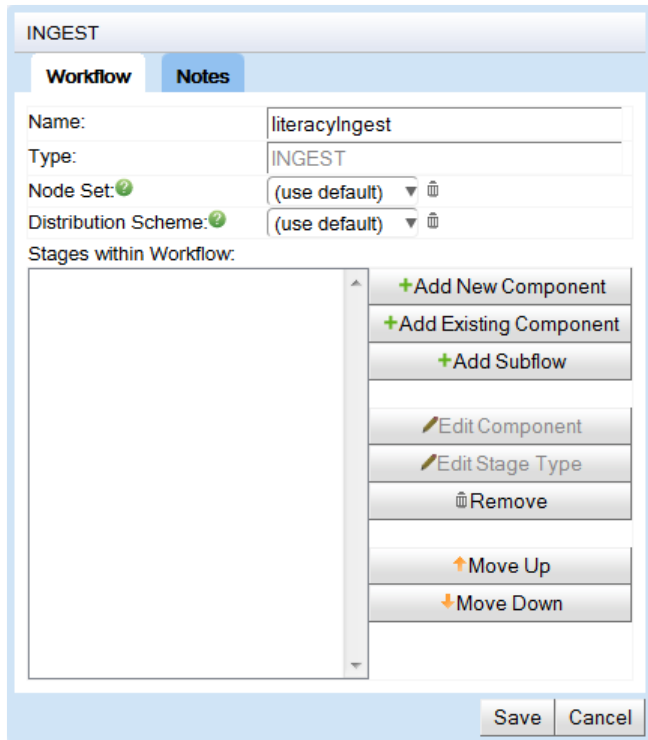
*A blank Workflow Editor window will appear.*



# Creating a New Workflow

- Name the workflow **literacyIngest**, as shown

*As you saw when we tested it, the **literacy** connector ingests the data as a single document. Clearly, though, we'd much rather have many documents, each corresponding to a single record. Since this is an XML file, our ingest workflow should implement the following steps:*



1. *Parse the raw binary into XML*
2. *Split the file into multiple Attivio Documents*
3. *Map XML tags to Attivio Document fields*
4. *Feed the prepared documents through the standard ingestion and indexing process*





# Creating a New Workflow—Parse XML

*Parsing binary into XML is a very common requirement. As such, it is a component that has already been defined and configured in other workflows.*

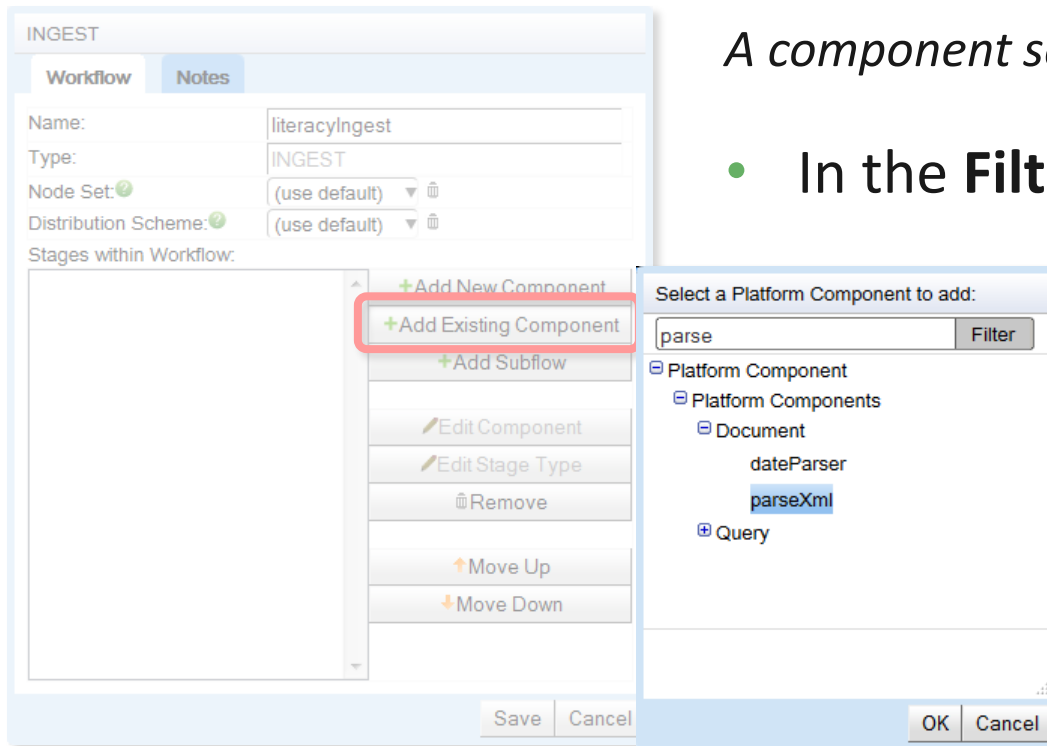
- Click **Add Existing Component**

*A component selection window will appear.*

- In the **Filter** field, search for **parse**

- Select **parseXML**

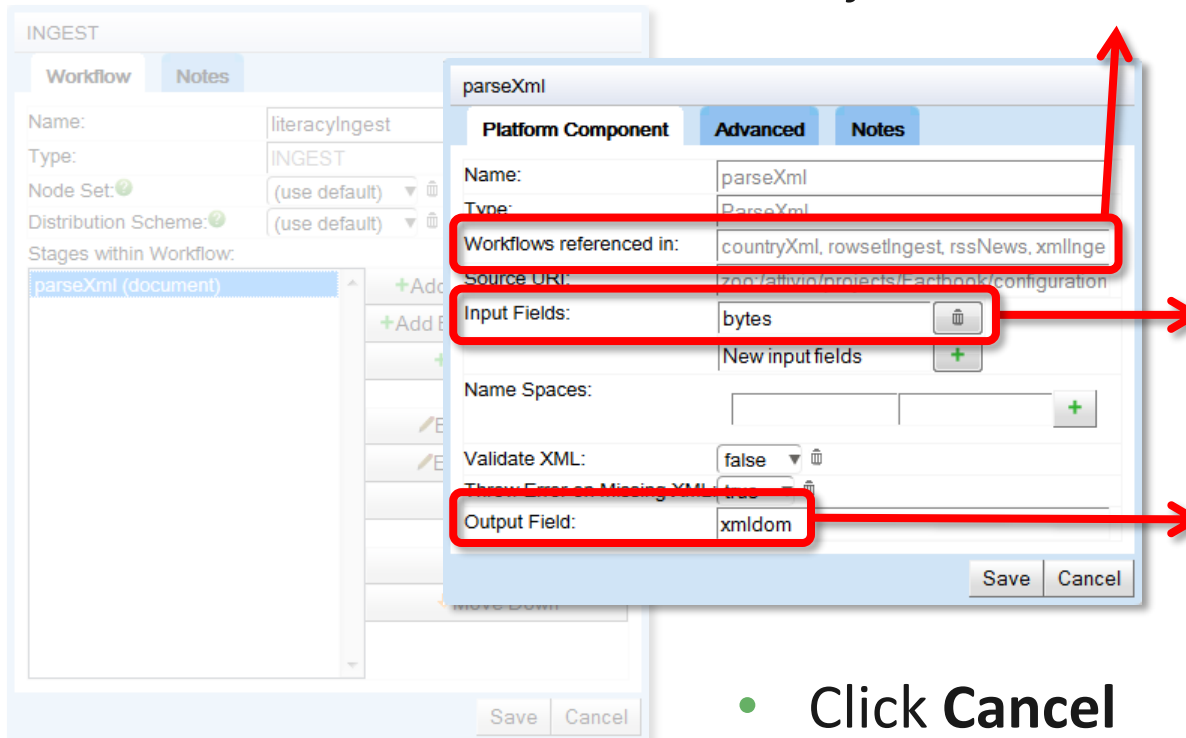
- Click **OK**



# Creating a New Workflow—Parse XML

- Double-click the **parseXml** component

*As mentioned, there are a large number of workflows that use this process.*



*This component reads data from the **bytes** field...*

*...and writes the XML formatted version into the **xmldom** field.*

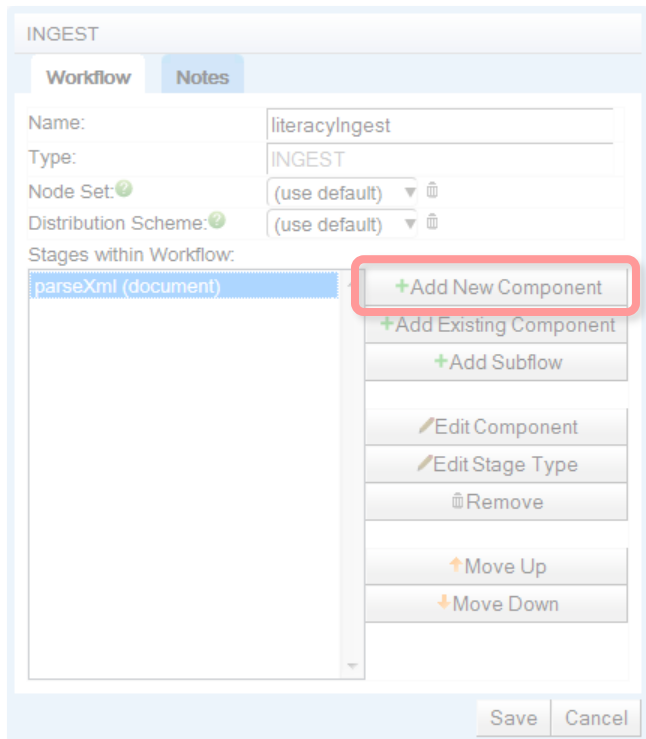
- Click **Cancel**



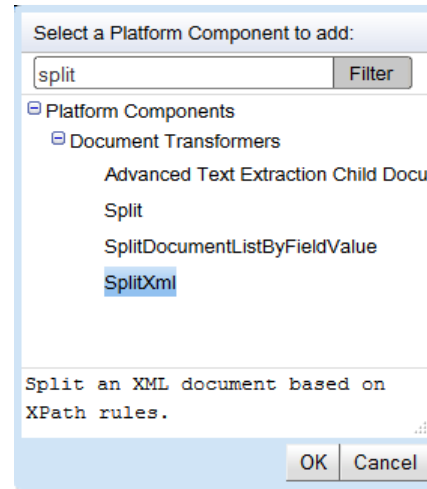
# Creating a New Workflow—Split XML

*Next, we need so split the file into multiple Attivio Documents, one for each **<record>** tag*

- Click **Add New Component**



- Locate and select **SplitXML**



- Click **OK**

*A new component editor window will appear*



## Creating a New Workflow—Split XML

- Name the component **splitLiteracy**
- Fill out the component parameters according to the following table:

Parameter	Value	Description
Name	splitLiteracy	The component name. This can be whatever you wish, as long as it's unique.
Xpath to Document	/ROOT/data/record	This defines which XML tag will be used to define the boundaries of the new Attivio Documents.
Xpath to ID	//field[@name="Year"]	This defines which element of the XML should be used as the ID for the new documents. Unfortunately, none of the XML tags contains something that will be unique. We'll use the <b>Year</b> tag as a temporary ID, and create a truly unique ID later on in the workflow.

Save Cancel



# Creating a New Workflow—Split XML

*The finished component should look like this:*

The screenshot shows the 'New SplitXml' dialog box in the ATTIVO workflow editor. The dialog has four tabs: 'Platform Component', 'Advanced', 'Notes', and 'Other'. The 'Advanced' tab is selected. The fields are as follows:

- Name: splitLiteracy
- Type: SplitXml
- Workflows referenced in: is defined in the following workflows
- Input Fields: xmldom (with a trash icon and a '+' icon for 'New input fields')
- \*Splitting Rules:
  - XPath to Document: /ROOT/data/record
  - XPath to ID: //field[@name='Year'] (with a '+' icon)
- Copy Parent Fields: true (with a trash icon)
- Throw Error on Missing XML: true (with a trash icon)
- Auto Number Child Document Ids: false (with a trash icon)
- Drop parent Document: true (with a trash icon)
- Namespaces: (empty field with a '+' icon)

At the bottom of the dialog are 'Save' and 'Cancel' buttons. A red arrow points to the 'xmldom' input field. In the background, a workflow editor is visible with a workflow named 'INGEST' and a stage named 'parseXml (document)'.

*Note that this component uses the data from the **xmldom** field that was created by **parseXML***

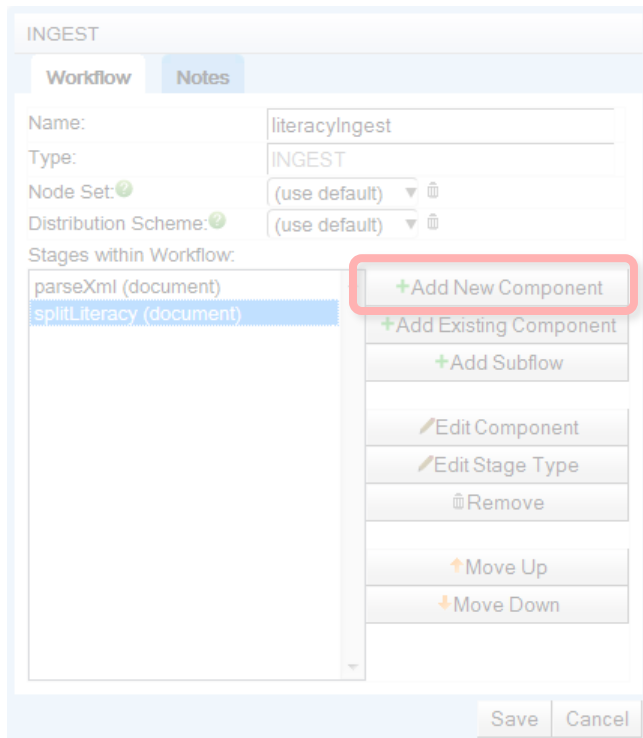
- Click Save



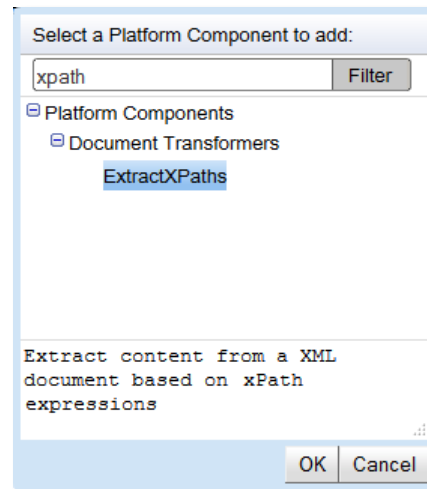
# Creating a New Workflow—Field Mapping

*Next, we need create a mapping from the XML tags in the source to Attivio Document fields*

- Click **Add New Component**



- Locate and select **ExtractXPath**



- Click **OK**

*A new component editor window will appear*





# Creating a New Workflow—Field Mapping

- Name the component **extractLiteracy**
- Fill out the **Extraction Rules** parameters according to the following table:

Output Field Name	Xpath to Extract
country	//field[@name="Country or Area"]
gender	//field[@name="Subgroup"]
literacyRate	//field[@name="Value"]
year	//field[@name="Year"]

Refer to the [Literacy.xml](#) file to understand this configuration.

*Note that the country field is already in use in the Factbook project, but the **gender**, **literacyRate**, and **year** fields do not exist. We can address this by either adding those fields to the schema, or by making these into dynamic fields by adding the appropriate suffix (e.g., **\_s** or **\_i**). In this exercise, we'll follow the best practice, and add the new fields to the schema.*



# Creating a New Workflow—Field Mapping

*The finished component should look like this:*

**New ExtractXPath**

**Platform Component** | **Advanced** | **Notes** | **Other**

Name:

Type:

Workflows referenced in:

Input Fields:

Throw Error on Missing XML:

Schema Name:

Extraction Rules:

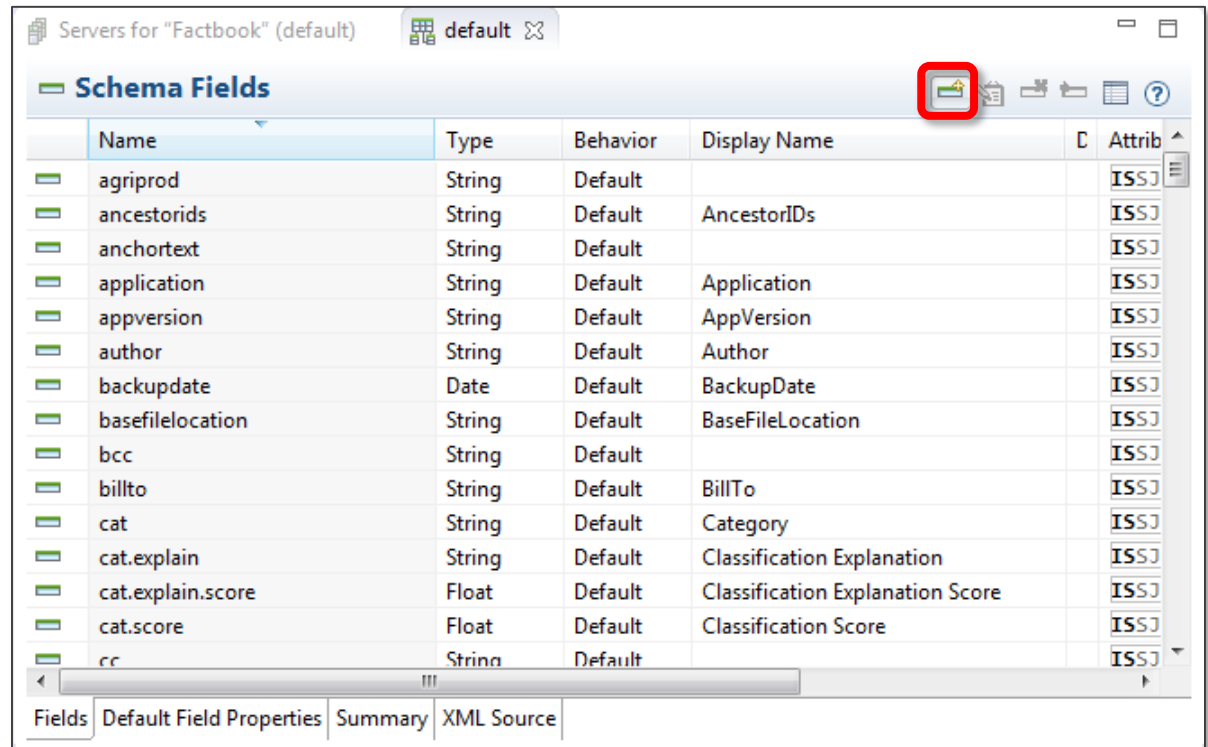
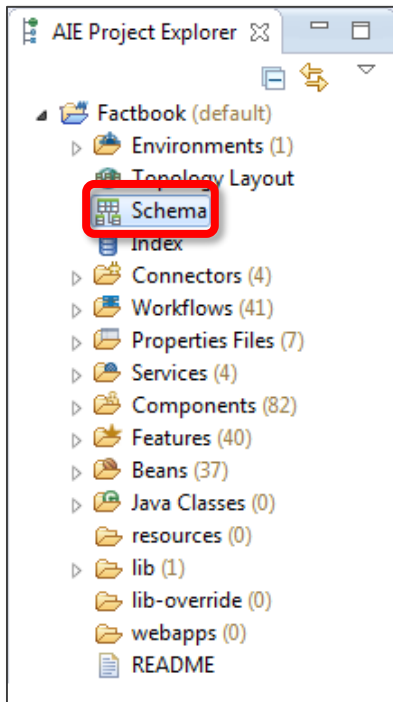
Output Field Name	xPath to Extract	
country	ame="Country or Area"]	<input type="button" value="🗑"/>
gender	!d[@name="Subgroup	<input type="button" value="🗑"/>
literacyRate	//field[@name="Value"	<input type="button" value="🗑"/>
year	//field[@name="Year"]	<input type="button" value="⊕"/>

- Click **Save**



# Adding to the Schema

- In **Designer**, open the **Schema** file
- Click the **New Field** button



*A New Field wizard will appear.*



# Adding to the Schema

- Enter **gender** as the field name

*All of the default attributes are fine for this field. This will be a String type, which is indexed, stored, and facetable. We don't need to implement sorting or JOIN capabilities on **gender**.*

- Check the box **Create another field after this one**
- Click **OK**

The screenshot shows the 'New Schema Field' dialog box. The 'Field Name' is 'gender' and 'Field Type' is 'String'. The 'Create another field after this one' checkbox is checked. The 'OK' button is highlighted.

**New Schema Field**

Enter values to create a new schema field.

Field Name:

Field Type:

☐ Real-time field

Display Name:

Default Value:

Attributes:

- ☒ Indexed (will be searchable)
- ☒ Stored (appears in result lists)
- ☐ Sort (allow use as a key for sorting)
- ☐ Joinable (can be used in joins, won't be tokenized)
- ☒ Facet (can be used in facet queries)
- ☐ Multi-value (supports multiple values per document)

Tokenize:

Lowercase:

Properties:

Included Fields:

☒ Create another field after this one



# Adding to the Schema

- Enter **year** as the field name
- Change the Field Type to **Integer**

*This will enable number-specific capabilities, like “greater than / less than” and value range searches*

- Check the **sort** attribute

*This will enable results sorting based on year*

- Click **OK**

The screenshot shows the 'New Schema Field' dialog box. The 'Field Name' is 'year' and the 'Field Type' is 'Integer'. The 'Sort' attribute is checked. The 'OK' button is highlighted.

**New Schema Field**  
Enter values to create a new schema field.

Field Name: year  
Field Type: Integer  
☐ Real-time field

Display Name:   
Default Value:

Attributes: ☒ Indexed (will be searchable)  
☒ Stored (appears in result lists)  
☒ Sort (allow use as a key for sorting)  
☐ Joinable (can be used in joins, won't be tokenized)  
☒ Facet (can be used in facet queries)  
☐ Multi-value (supports multiple values per document)

Tokenize: Auto (tokenized if it's an indexed field of type String)  
Lowercase: Auto (lowercase if tokenized, case sensitive)

Properties: No properties Edit...  
Included Fields: No includes Edit...

☒ Create another field after this one

? OK Cancel



# Adding to the Schema

- Enter **literacyRate** as the field name
- Change the Field Type to **Float**

*Just like Integer, the Float type will enable number-specific processing and search capabilities*

- Check the **sort** attribute
- Uncheck the **Create another field** box
- Click **OK**

**New Schema Field**

Enter values to create a new schema field.

Field Name: literacyRate

Field Type: Float

☐ Real-time field

Display Name:

Default Value:

Attributes: ☒ Indexed (will be searchable)  
☒ Stored (appears in result lists)  
☒ Sort (allow use as a key for sorting)  
☐ Joinable (can be used in joins, won't be tokenized)  
☒ Facet (can be used in facet queries)  
☐ Multi-value (supports multiple values per document)

Tokenize: Auto (tokenized if it's an indexed field of type String)

Lowercase: Auto (lowercase if tokenized, case sensitive)

Properties: No properties [Edit...](#)

Included Fields: No includes [Edit...](#)

☐ Create another field after this one

[?](#) [OK](#) [Cancel](#)

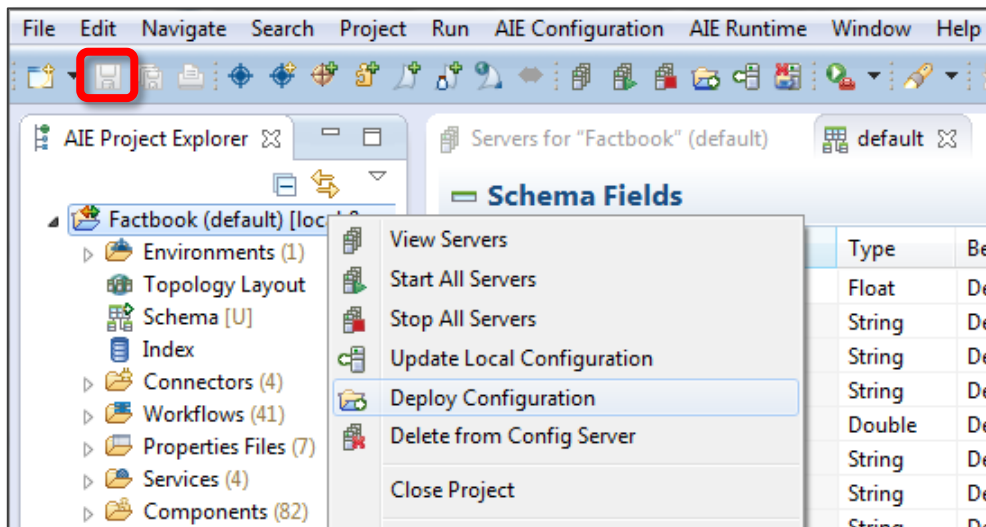




# Deploying the Schema Changes

*Recall that, in order for changes made in Designer to take effect, we need to deploy the new configuration.*

- **Save** the schema file
- Right-click the Factbook folder, and click **Deploy Configuration**



*This will shut down the **local** node, update the configuration on the server, and restart the node.*

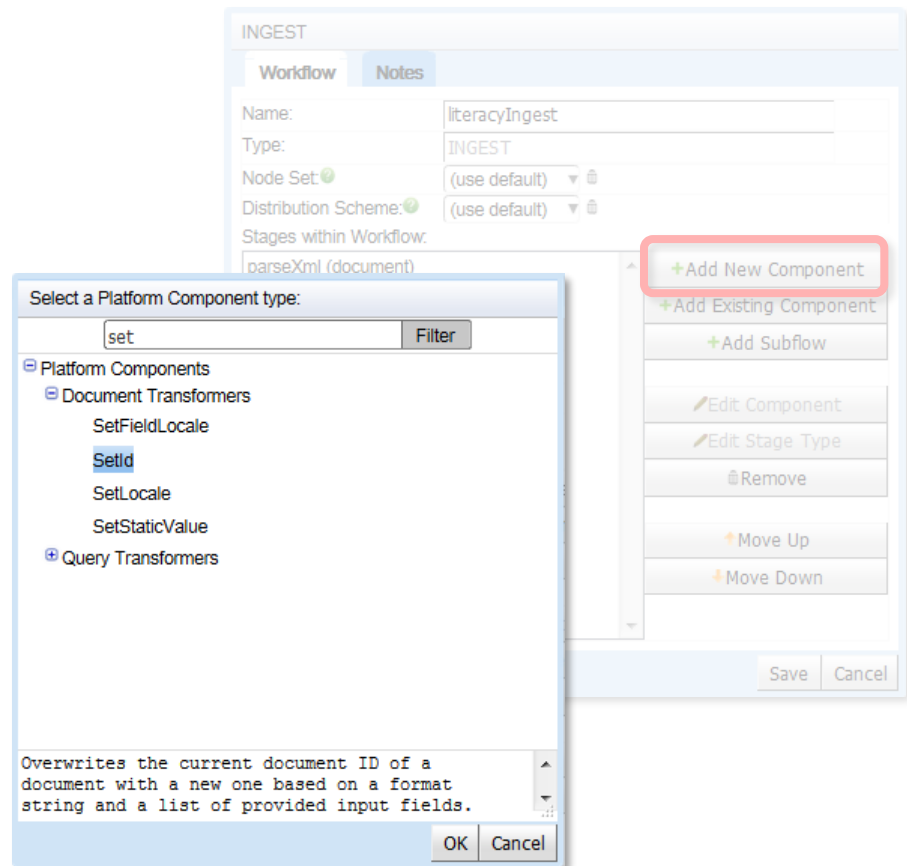
*If the node does not restart automatically, you can do so manually with the **Start All Servers** command.*



# Creating a New Workflow—Set ID

*Now that the source data has been properly mapped to Attivio Document fields, we can address the non-unique ID problem that arose during the SplitXML stage.*

- Click **Add New Component**
- Locate and select **SetId**
- Click **OK**



## Creating a New Workflow—Set ID

- Name the component **setLiteracyId**
- Fill out the parameters according to the following table:

Parameter	Value
Input Fields	country
	gender
	year
Format String	LITERACY-%s-%s-%s

***SetId** creates a new ID by concatenating multiple document fields' values into a single string.*

***Input Fields** specifies which fields to concatenate, and in what order.*

***Format String** defines how the concatenation will be formatted. The %s symbols are variables, each of which captures the values from one of the input fields list. For example, a literacy document for women in Germany in 1995 would have an ID of **LITERACY-Germany-Female-1995***



# Creating a New Workflow—Set ID

*The finished component should look like this:*

The screenshot shows the 'New SetId' dialog box in the ATTIV/O interface. The dialog is titled 'New SetId' and has three tabs: 'Platform Component', 'Advanced', and 'Notes'. The 'Platform Component' tab is active. It contains the following fields and values:

- Name: setLiteracyId
- Type: SetId
- Workflows referenced in: is defined in the following workflows
- Input Fields: country, gender, year, New input fields
- Format String: LITERACY-%s-%s-%s
- Multi-Value Separator: ,
- Default Value for missing fields:

At the bottom right of the dialog are 'Save' and 'Cancel' buttons. In the background, the 'INGEST' workflow editor is visible, showing a list of stages: 'parseXml (document)', 'splitLiteracy (document)', and 'extractLiteracy (document)'.

- Click **Save**



# Creating a New Workflow—Drop DOM

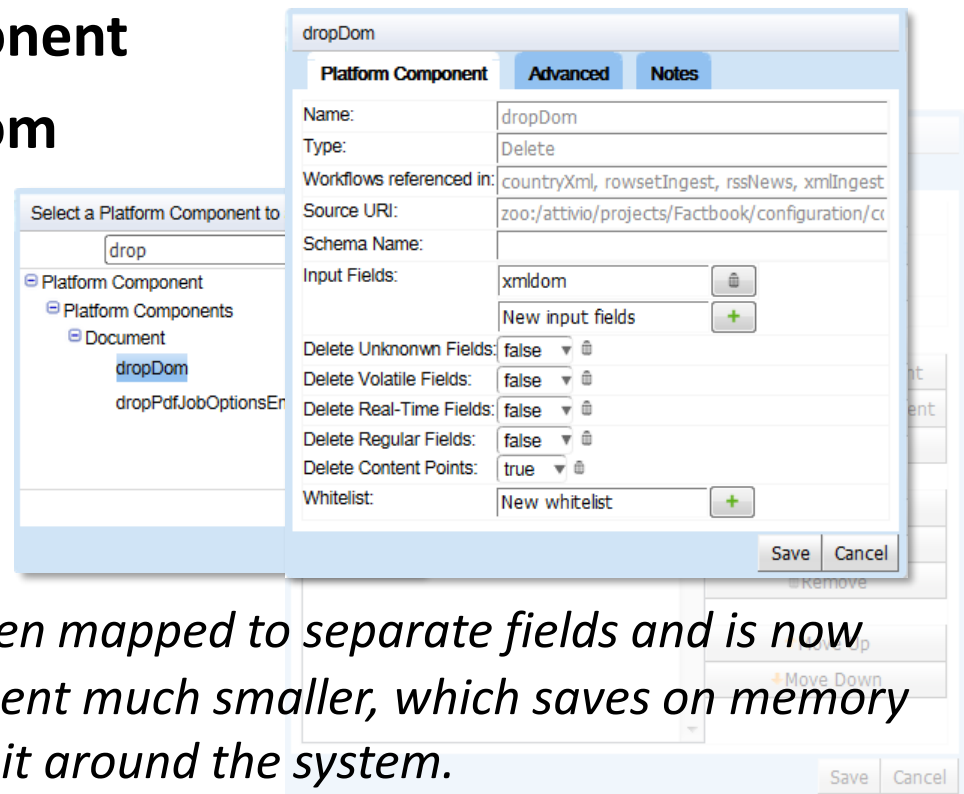
*At this point, the workflow is doing all of the things we initially wanted. All that remains is to send the documents to the index.*

*But first, we'll add one last step for performance optimization.*

- Click **Add Existing Component**
- Locate and select **dropDom**
- click **OK**

***dropDom** is a Delete component, which deletes every input field from the document. In this case, we're deleting the **xmlDom** field created by the **parseXml***

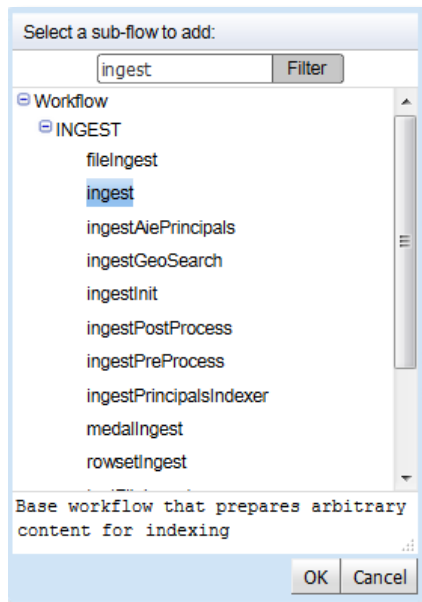
*component, since the XML has been mapped to separate fields and is now redundant. This makes the document much smaller, which saves on memory and network traffic while passing it around the system.*



# Creating a New Workflow—Indexing

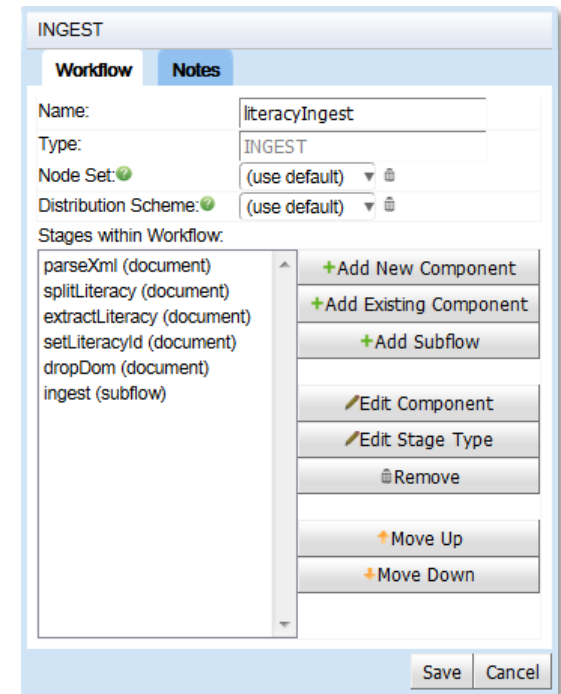
*Finally, the workflow will send the documents through the standard cleanup and indexing process.*

- Click **Add Subflow**
- Locate and select **ingest**
- Click **OK**



*The finished workflow should look like this:*

- Click **Save**





# **CONFIGURING THE CONNECTOR TO USE THE NEW WORKFLOW**



# Implementing the New Workflow

- In **Admin UI**, go to the **Connectors** page
- Click on the **literacy** connector
- Change the **Ingest Workflow** to **literacyIngest**
- Click **Save**

FileScanner: literacy

Scanner	Notes	Scheduler	Field Mappings	Advanced
Connector Name:		literacy		
Node Set:		processors		
*Start Directory:		C:\attivio_4.3.2\input\Literacy.xml		
Follow symbolic links:		true		
Maximum directory depth:		-1		
Maximum File Size (MB):		5		
Minimum File Size (MB):		-1		
Wildcard Include Filter:		*.xml		
		New wildcard include filter +		
Wildcard Exclude Filter:		\$*		
		*.tmp		
		*~		
		.*		
		~*		
		New wildcard exclude filter +		
Directory Listing Timeout:		-1		
Document ID Prefix:				
Ingest Workflow:		lit		
► Incremental		literacyIngest		
► Advanced				
		Save Save & Test Cancel		

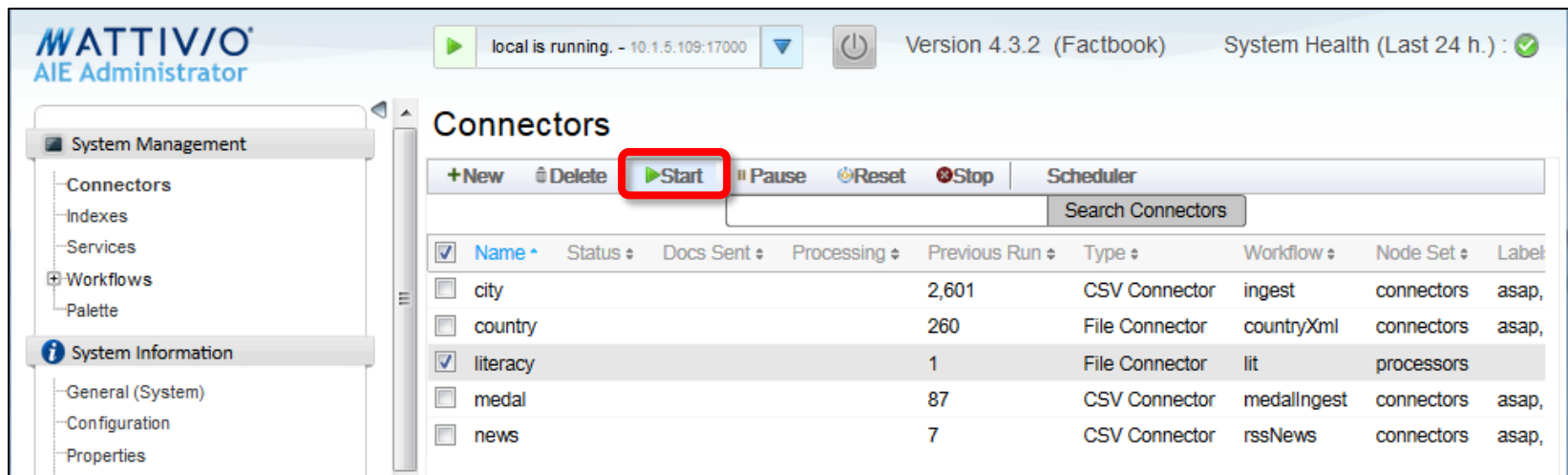


# INGESTING THE DATA



# Ingesting the Data

- On the Connectors page of Admin UI, check the box next to the literacy connector
- Click **Start**



WATTIV/O AIE Administrator

local is running. - 10.1.5.109:17000

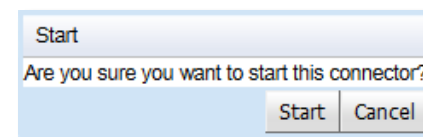
Version 4.3.2 (Factbook) System Health (Last 24 h.): ✔

### Connectors

[+New](#) [Delete](#) **▶Start** [Pause](#) [Reset](#) [Stop](#) [Scheduler](#)

<input checked="" type="checkbox"/>	Name	Status	Docs Sent	Processing	Previous Run	Type	Workflow	Node Set	Label
<input type="checkbox"/>	city				2,601	CSV Connector	ingest	connectors	asap,
<input type="checkbox"/>	country				260	File Connector	countryXml	connectors	asap,
<input checked="" type="checkbox"/>	literacy				1	File Connector	lit	processors	
<input type="checkbox"/>	medal				87	CSV Connector	medalIngest	connectors	asap,
<input type="checkbox"/>	news				7	CSV Connector	rssNews	connectors	asap,

- In the dialog box that appears, click **Start** again

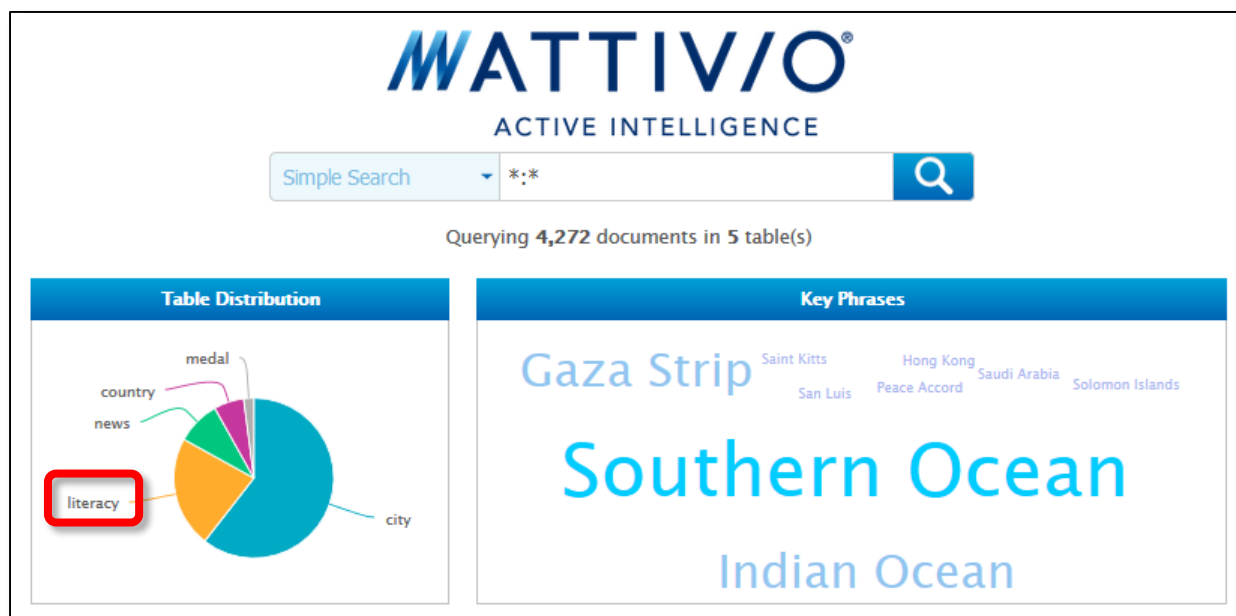


# EXPLORING THE DATA

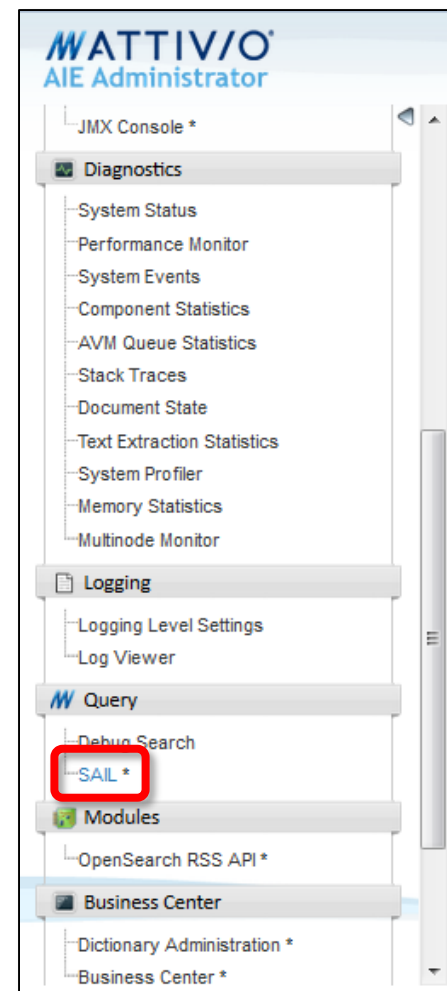


# Exploring the Data

- In **Admin UI**, scroll down to the **Query** section of the left-side menu
- Open **SAIL** in a new tab



*Note that **literacy** is now part of the table field graph*



# Exploring the Data

- Enter the query **table:literacy**

*This will return all of the new literacy documents.*

*Note the very useful facet filters based on the new data.*

The screenshot displays the WATTIV/O search interface. At the top, the search bar contains 'table:literacy' and the search options are set to 'Simple Search'. The results show 960 documents found in 107 milliseconds, sorted by Relevance. The results are faceted by gender, year, and literacyRate.

**Table**

Facet	Count
<b>gender</b>	
Female 15+ yr	480
Male 15+ yr	480
<b>year</b>	
1996 - 2005	560
1991 - 1992	84
1990 - 1991	56
1994 - 1995	48
1989 - 1990	40
1992 - 1993	40
1995 - 1996	20
1988 - 1989	16
1987 - 1988	12
1993 - 1994	12
<b>literacyRate</b>	
90.63 - 99.80	344
81.46 - 90.63	178
72.29 - 81.46	132
63.12 - 72.29	74
53.95 - 63.12	60
44.78 - 53.95	38
35.61 - 44.78	36
26.44 - 35.61	32
17.27 - 26.44	26
8.10 - 17.27	22

**1. literacy**  
Details  
Date: 2015-03-17T13:34:16  
Tags: [Assign Value](#)

**2. literacy**  
Details  
Date: 2015-03-17T13:34:16  
Tags: [Assign Value](#)

**3. literacy**  
Details  
Date: 2015-03-17T13:34:16  
Tags: [Assign Value](#)

**4. literacy**  
Details  
Date: 2015-03-17T13:34:16  
Tags: [Assign Value](#)

**5. literacy**  
Details  
Date: 2015-03-17T13:34:16

*The documents, though, look incomplete! Don't worry—by default, SAIL only displays a small subset of fields. The data is in fact all there, as you will see in a moment.*





# Exploring the Data

- Click the **Search Options** button at the top of the page
- Check the **Debug** box, and click **Search**

*Debug mode displays the entire Attivio Document. Now you can see that each literacy document contains file metadata, as well as the special fields we created.*

The screenshot shows the WATTIV/O search interface. At the top, there is a search bar with the text "table:literacy" and a "Search Options" button highlighted with a red box. Below the search bar, the "Search Options" panel is visible, with the "General" tab selected. The "Debug" checkbox is checked. There are "Cancel" and "Search" buttons. Below the search options, a message states "Your search found 960 results in 81 milliseconds." with a "View Query Feedback" button and a "Sort" dropdown set to "Relevancy".

On the left, there is a table of results:

Table	
literacy	960
gender	
Female 15+ yr	480
Male 15+ yr	480
year	
1996 - 2005	560
1991 - 1992	84
1990 - 1991	56
1994 - 1995	48
	40
	40
	20
	16
	12
	12
	20
	16
	16

On the right, the search results are displayed. The first result is "1. Literacy-Algeria-Female 15+ yr-2002". It shows a score of 0.23933873 and various metadata fields including .id, .zone, Size, Table, Date, sourcepath, filename, parentid, conversionErrorCode, country, gender, year, and literacyRate. The score explanation is also provided. The second result is "2. Literacy-Armenia-Male 15+ yr-2001" with a score of 0.23933873.

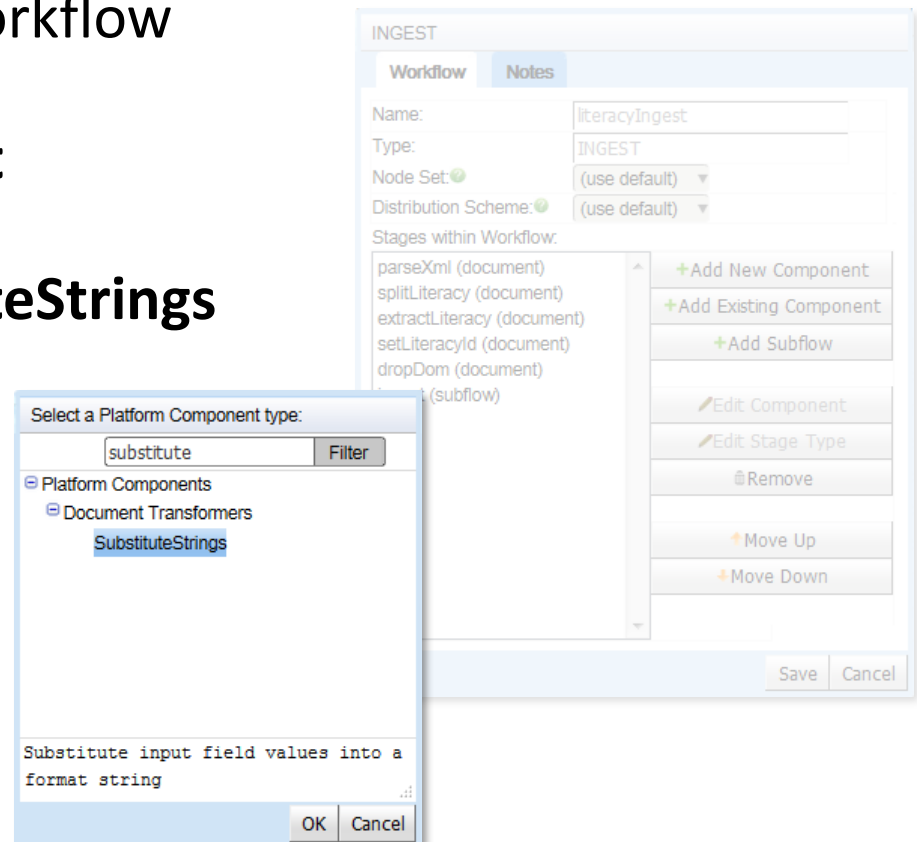


# ADDING A DOCUMENT TITLE (optional)



# Modifying the Workflow—Set Title

- In **Admin UI**, go to the **Ingest Workflows** page
- Open the **literacyIngest** workflow
- Click **Add New Component**
- Locate and select **SubstituteStrings**
- Click **OK**



## Modifying the Workflow—Set Title

- Name the component **setLiteracyTitle**
- Fill out the parameters according to the following table:

Parameter	Value
Input Fields	country
	gender
	year
Format String	%s %s Literacy, %s
Output Field	title

*Just like **SetId**, **SubstituteStrings** concatenates multiple document fields into a single string. The only difference is that you can write that compound string into ANY non-ID field.*

*Similar to **setLiteracyId**, this component will create a **title** field with values like **Germany Female Literacy, 1995***



# Modifying the Workflow—Set Title

*The finished component should look like this:*

INGEST

**Workflow** **Notes**

Name: literacyIngest

Type: INGEST

Node Set: (use default)

Distribution Scheme: (use default)

Stages within Workflow:

- parseXml (document)
- splitLiteracy (document)
- extractLiteracy (document)
- setLiteracyId (document)
- dropDom (document)
- setLiteracyTitle (document)**
- ingest (subflow)

+ Add New Component

+ Add Existing Component

+ Add Subflow

Edit Component

Edit Stage Type

Remove

Move Up

Move Down

Save Cancel

- Click **Save**

New SubstituteStrings

**Platform Component** **Advanced** **Notes**

Name: setLiteracyTitle

Type: SubstituteStrings

Workflows referenced in: is defined in the following workflows

Input Field: country

gender

year

New input field

String Format: %s %s Literacy, %s

Output Field: title

Save Cancel

- In the Workflow window, select **setLiteracyTitle** and click **Move Up** until it is above the **ingest** subflow
- Save** the workflow

*If you don't do this re-ordering, the title won't be created until after the documents have already been sent to the index, which would be pointless.*



# Re-ingesting the Data

- In **Admin UI**, go to the **Connectors** page
- Start the **literacy** connector again.
- In **SAIL**, re-issue the query **table:literacy**

The screenshot shows the WATTIV/O SAIL interface. At the top, there's a search bar with 'Simple Search' and a dropdown menu, and a search input field containing 'table:literacy'. Below the search bar, it says 'Search Options'. The main content area displays search results for 'table:literacy'. On the left, there's a table with three sections: 'Table', 'gender', and 'year'. The 'Table' section shows 'literacy' with 480 results. The 'gender' section shows 'Female 15+ yr' with 240 results and 'Male 15+ yr' with 240 results. The 'year' section shows '1996 - 2005' with 280 results, '1991 - 1992' with 42 results, and '1990 - 1991' with 28 results. On the right, there are two result cards. The first card is titled '1. Afghanistan Female 15+ yr Literacy, 2000' and shows 'literacy' with details: 'Date: 2015-03-17T13:34:16' and 'Tags: Assign Value'. The second card is titled '2. Afghanistan Male 15+ yr Literacy, 2000' and shows 'literacy' with details: 'Date: 2015-03-17T13:34:16' and 'Tags: Assign Value'.

Table	Count
literacy	480

gender	Count
Female 15+ yr	240
Male 15+ yr	240

year	Count
1996 - 2005	280
1991 - 1992	42
1990 - 1991	28

*Now all of the documents have titles, improving the search experience.*



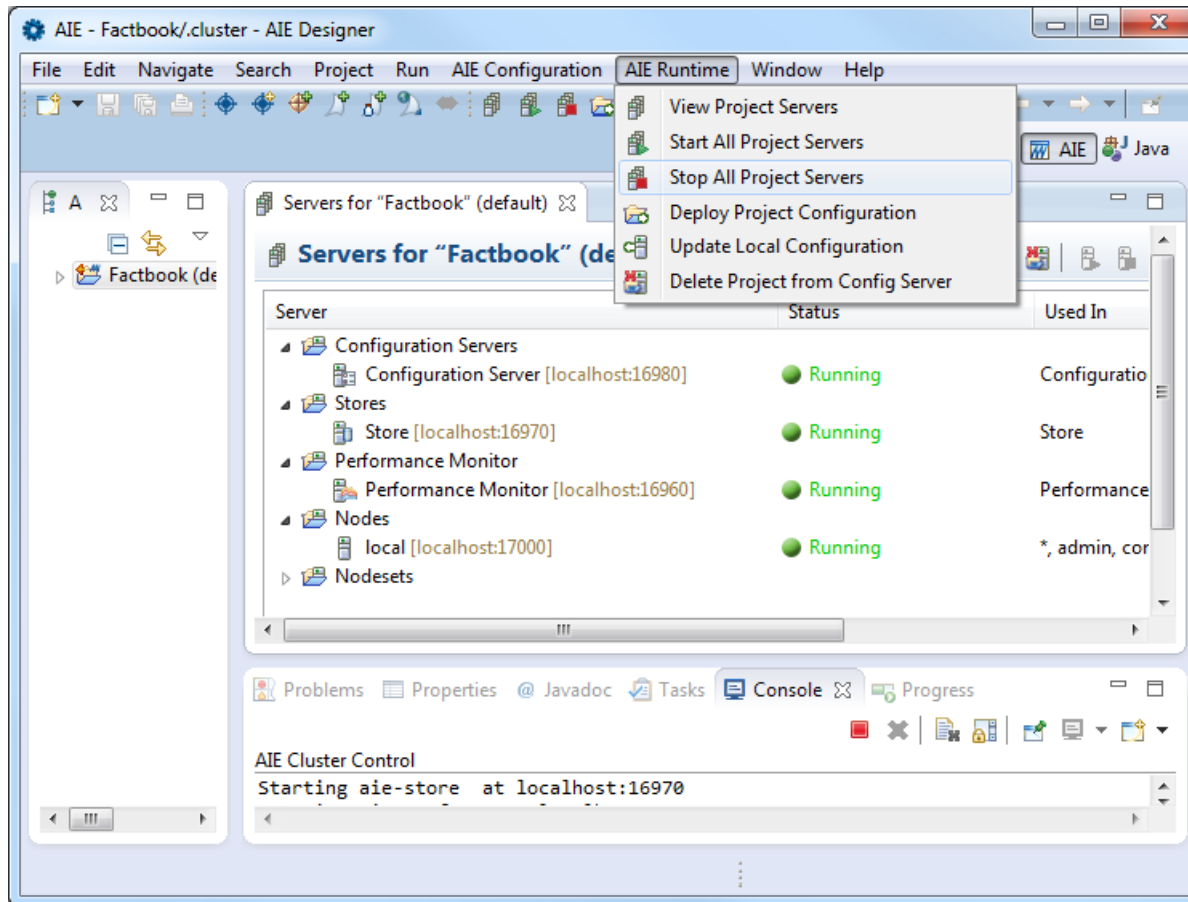
# SHUTTING DOWN ATTIVIO





# Shutting Down Attivio

- In Designer, click **AIE Runtime > Stop all project servers**



*Always shut down the Attivio engine when you're finished! If you shut down or sleep your computer with Attivio running, it can corrupt some configuration files and/or take a very long time for the machine to wake up.*

