Ingestion Lab

XML with File Connector

Lab Overview

ATTIVIO
The Data Dexterity Company

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

The Data Set

ATTIVIO
The Data Dexterity Company

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

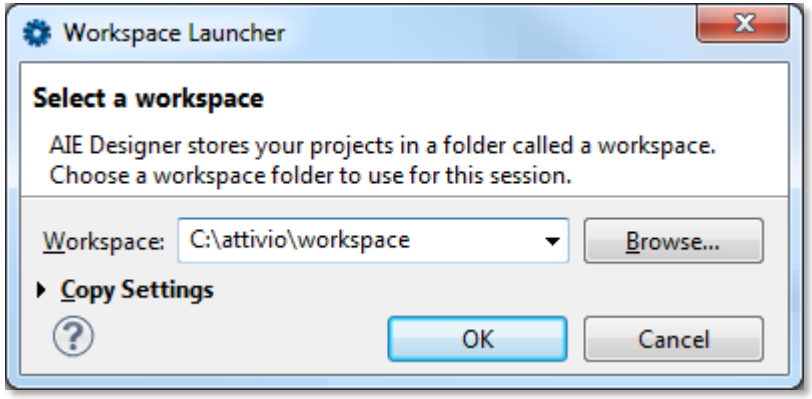
ATTIVIO
The Data Dexterity Company

- Start the **Attivio Designer** application

  C:\attivio\install\designer\Designer

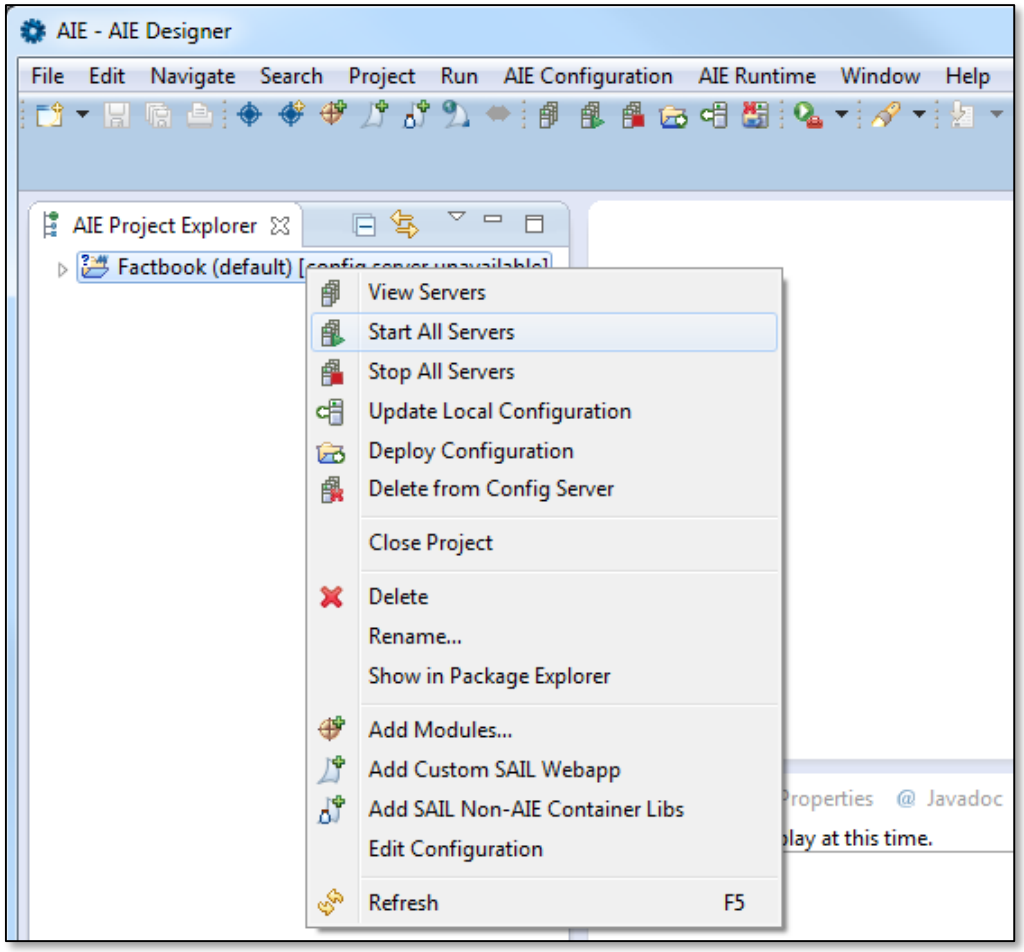
  Attivio Designer is an application intended for system architects and developers. It provides the means to easily and quickly configure, view, and edit every element of the Attivio platform.

- If prompted, specify the workspace location



Starting Attivio

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



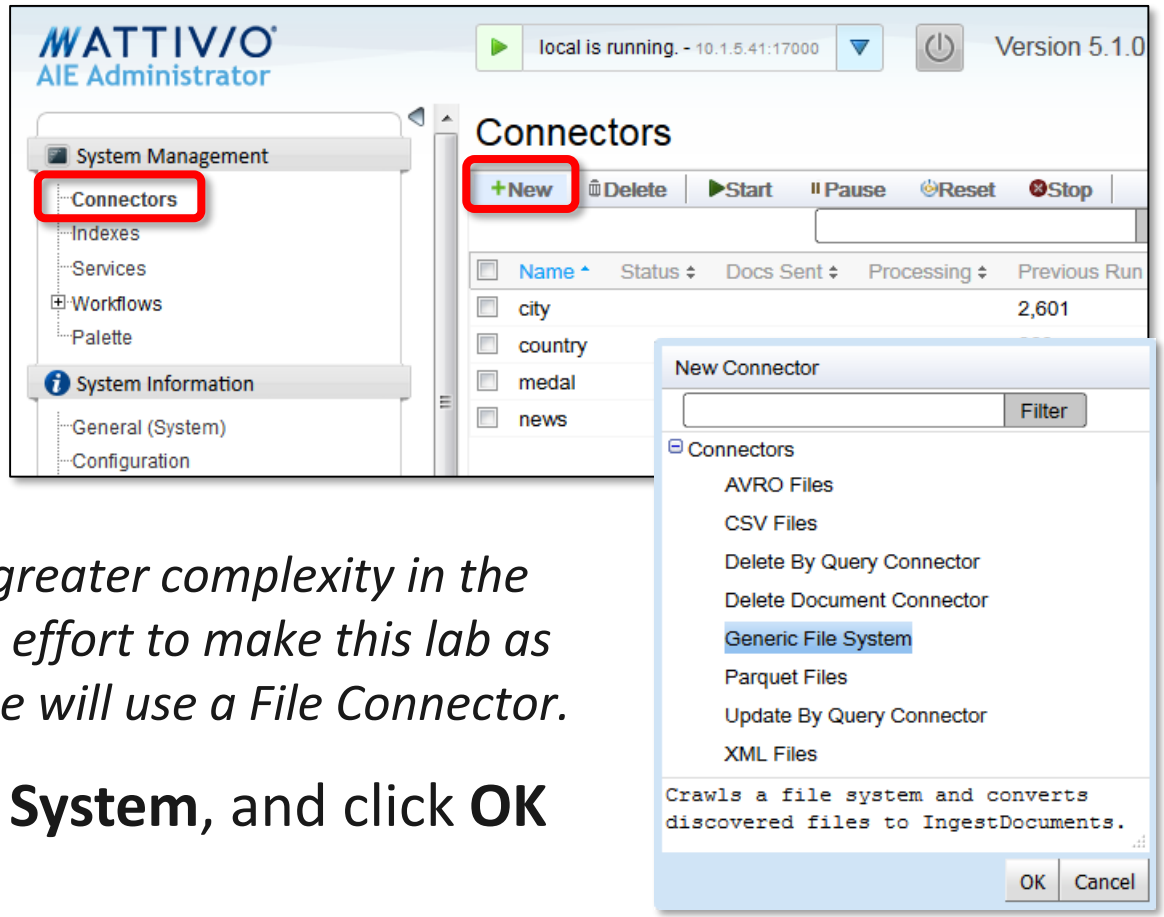
- Click **Start All Servers**

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.

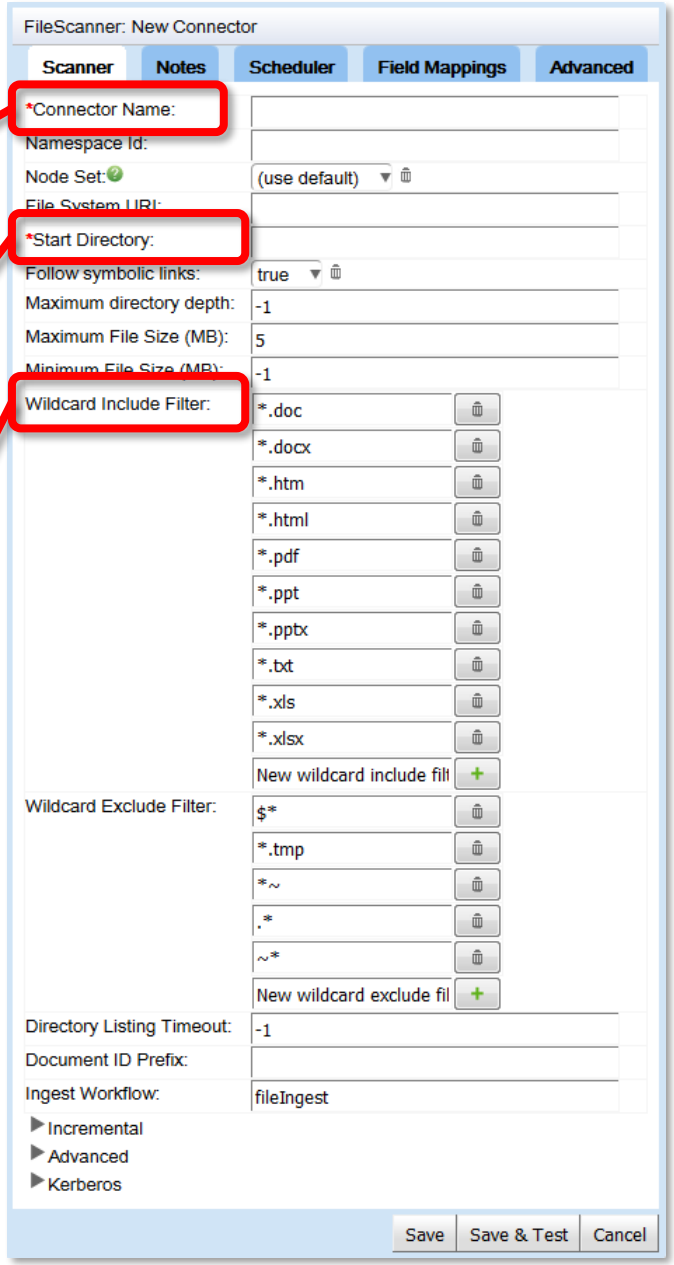
- Select **Generic File System**, and click **OK**

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

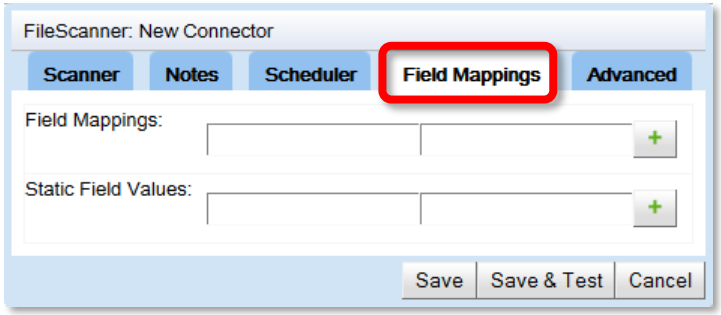


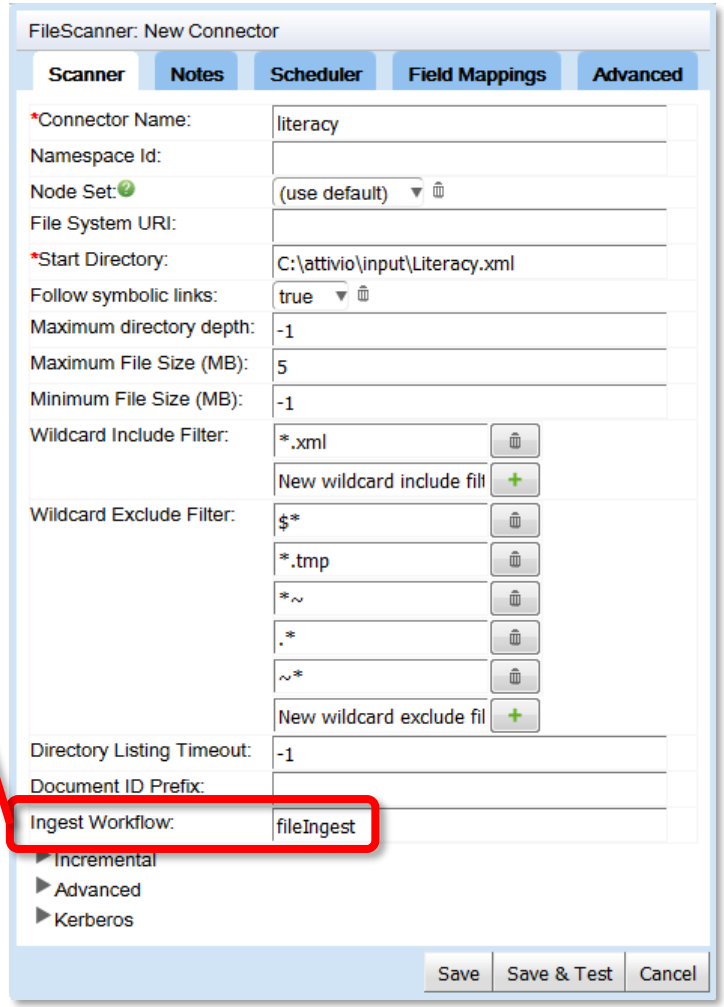
Configuring a New Connector

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

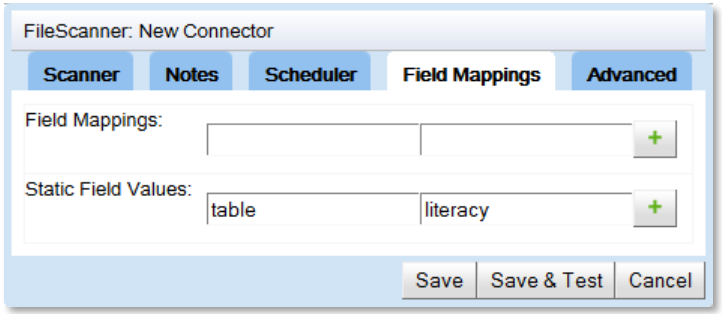




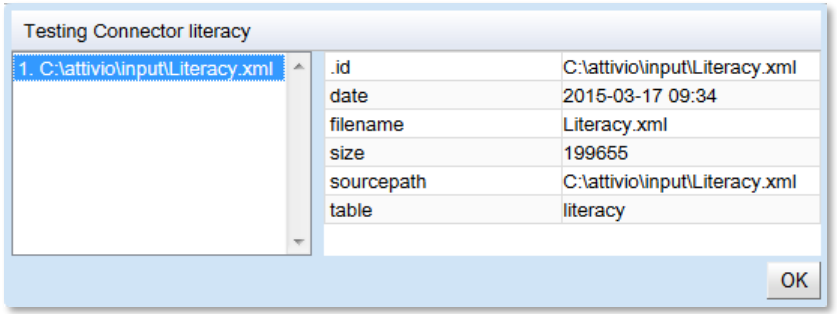
Configuring a New Connector

- 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



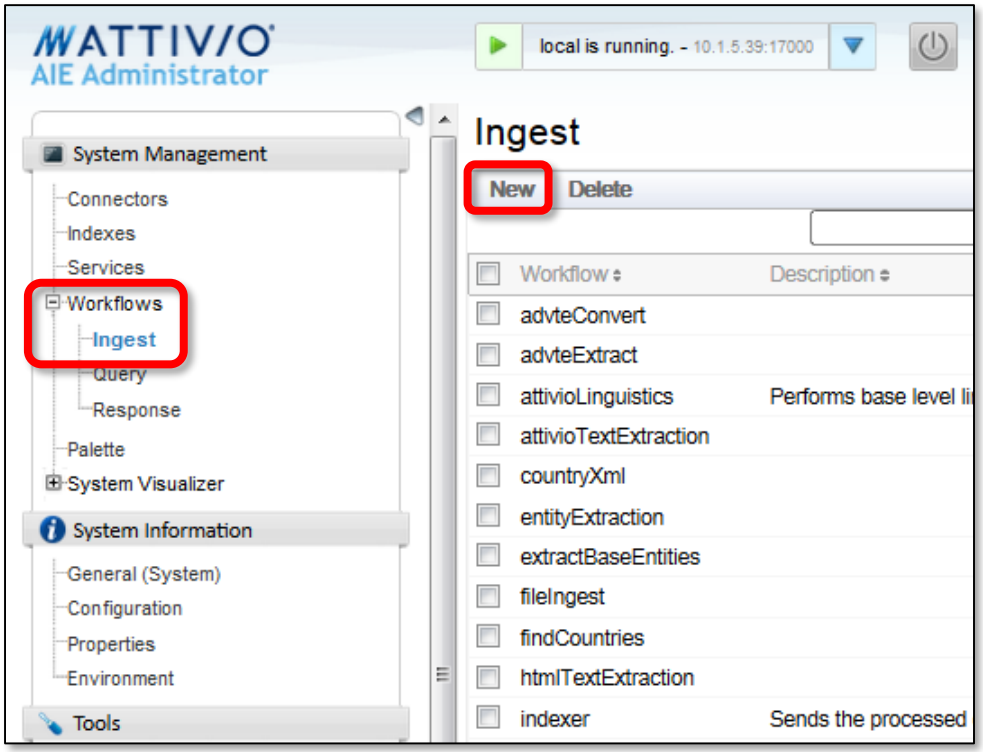
*One document should be produced. It will contain the file metadata, an **id** equivalent to the **sourcepath**, and the **table** field you specified.*

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

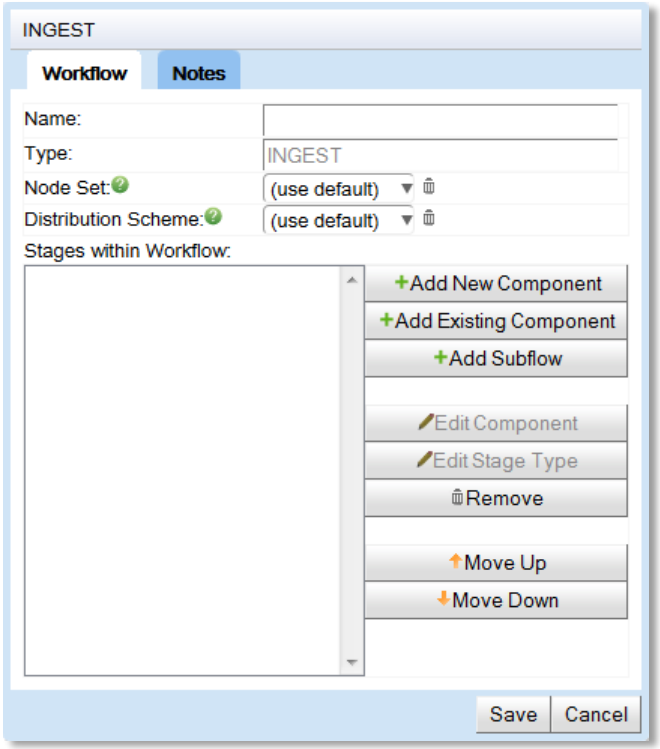
CREATING A NEW WORKFLOW

Creating a New Workflow

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



A blank Workflow Editor window will appear.

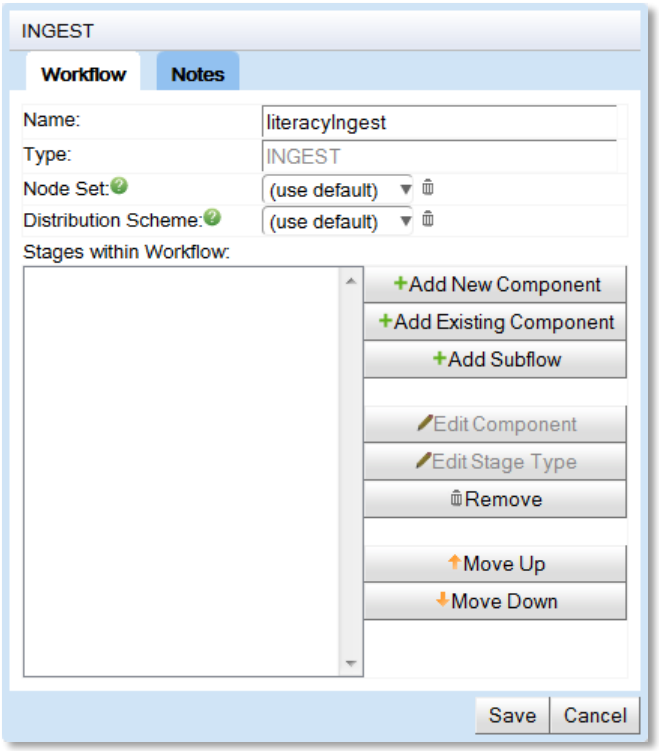


Creating a New Workflow

ATTIVIO
The Data Dexterity Company

- 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:*

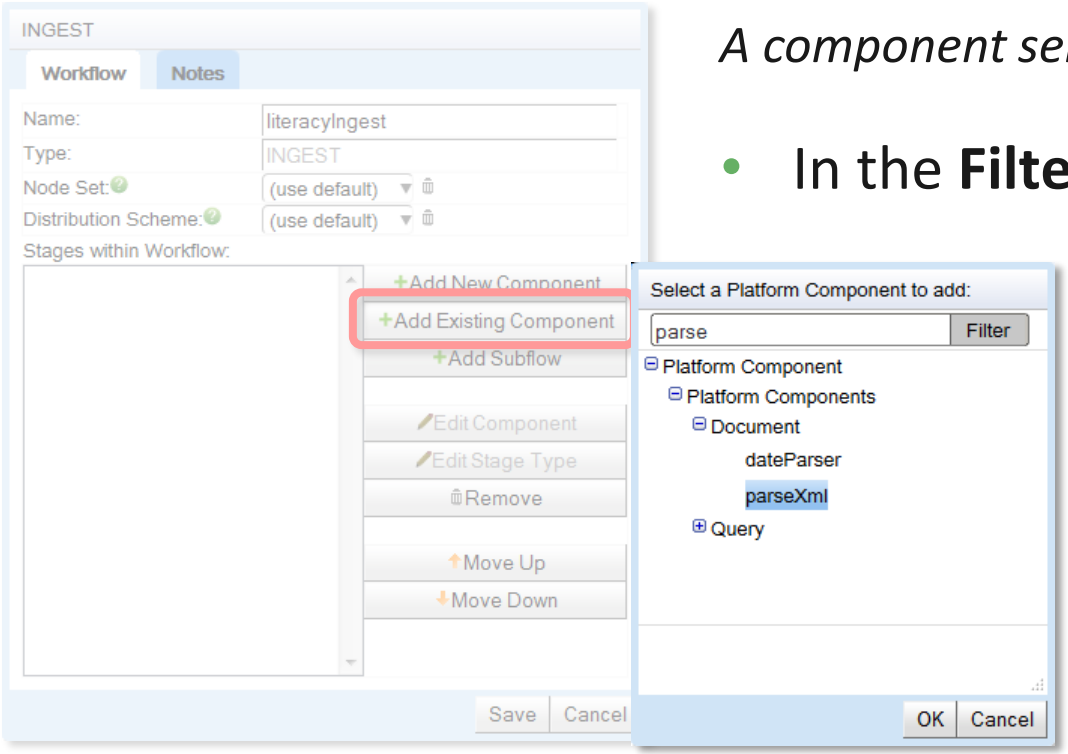


1. *Parse the raw binary into XML*
2. *Split the file into multiple Attivio Documents*
3. *Map XML tags to Attivio Document fields*
4. *Feed the prepared documents through the standard ingestion and indexing process*

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**



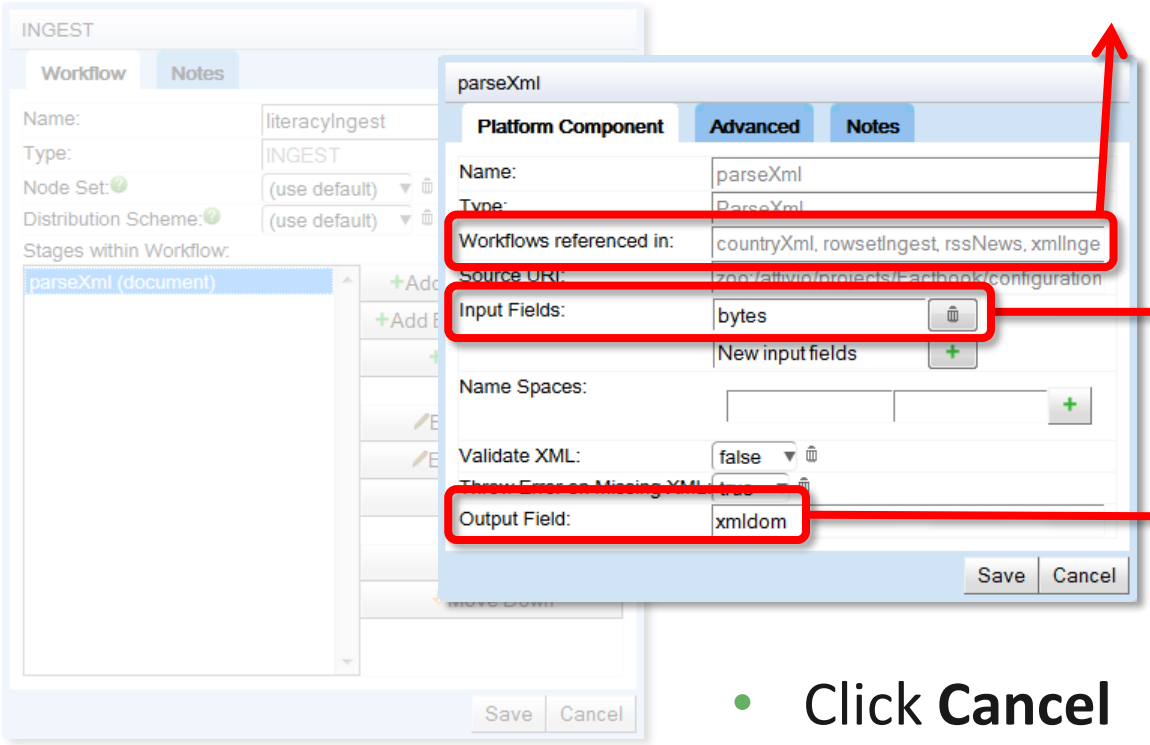
A component selection window will appear.

- 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…*

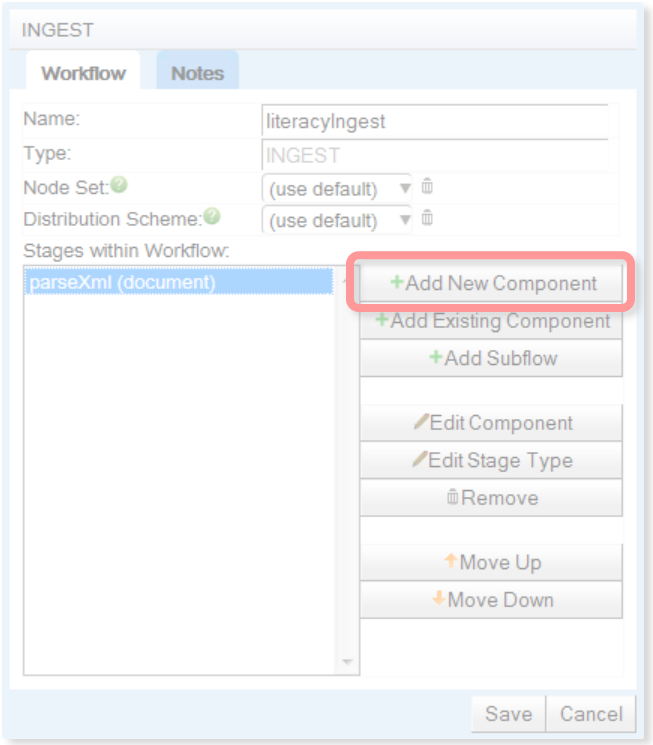
*…and writes the XML formatted version into the **xmldom** field.*

- Click **Cancel**

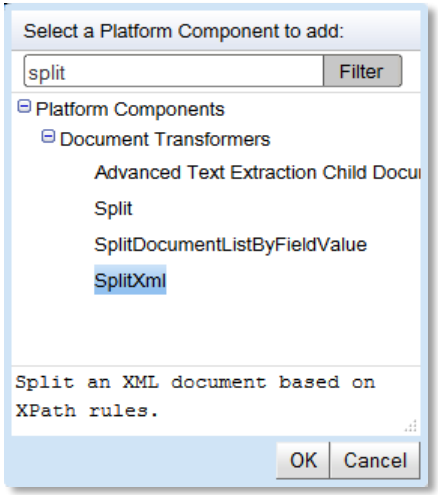
Creating a New Workflow—Split XML

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

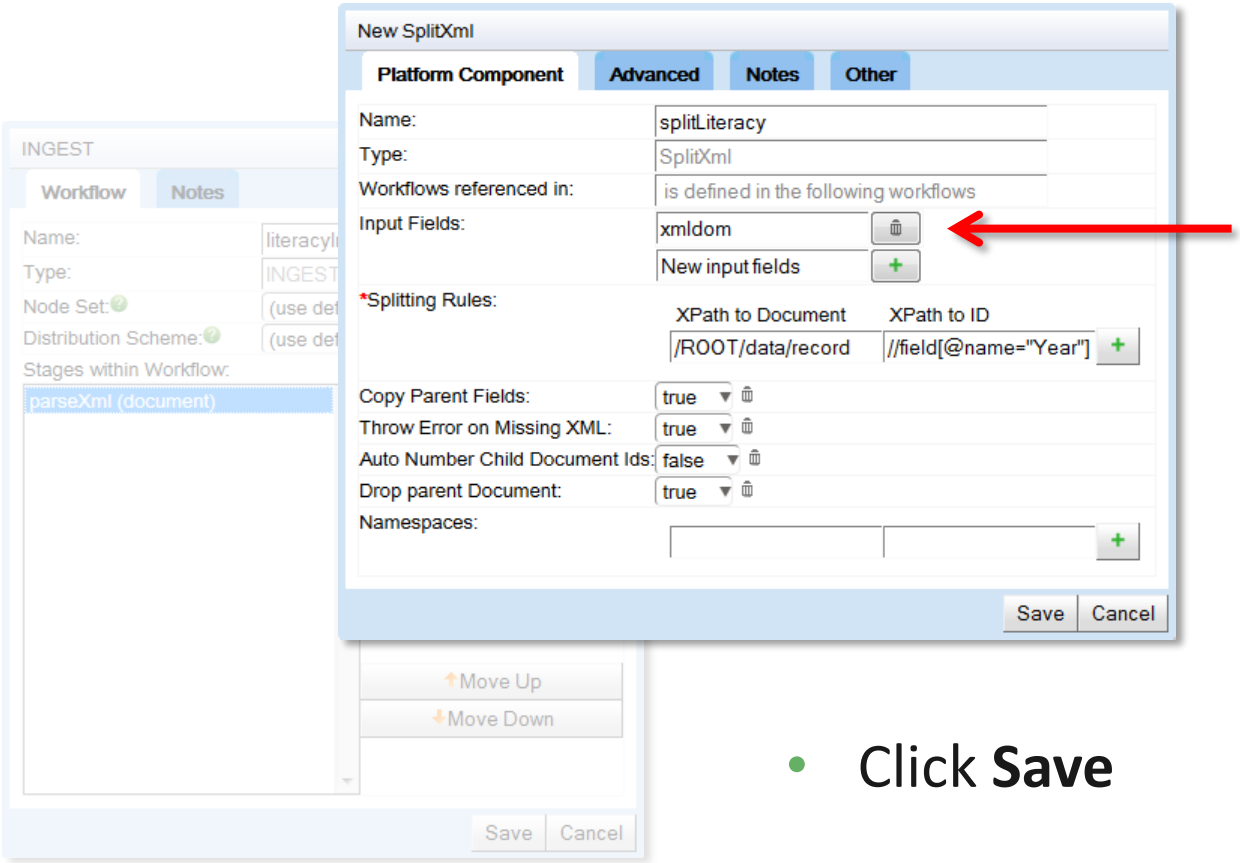
ATTIVIO
The Data Dexterity Company

- 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 **Year** tag as a temporary ID, and create a truly unique ID later on in the workflow.

Creating a New Workflow—Split XML

The finished component should look like this:



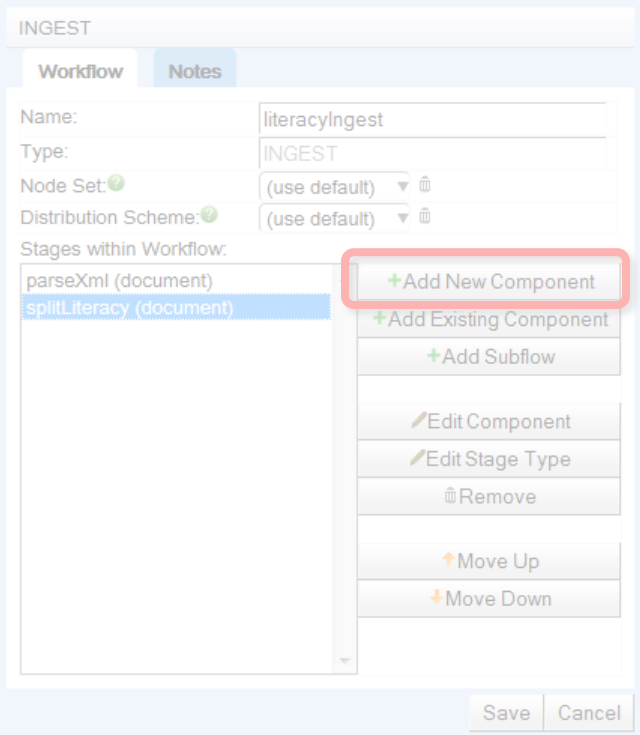
*Note that this component uses the data from the **xmldom** field that was created by **parseXML***

- Click **Save**

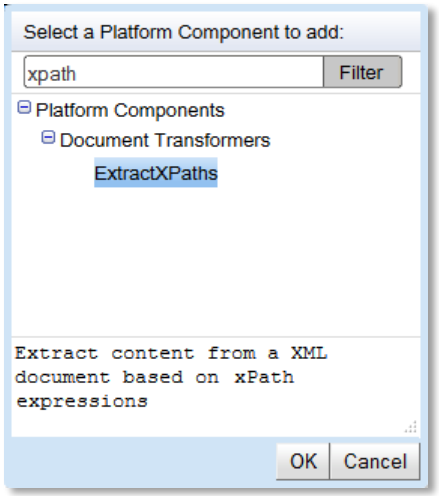
Creating a New Workflow—Field Mapping

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

- Click **Add New Component**



- Locate and select **ExtractXPaths**



- Click **OK**

A new component editor window will appear

Creating a New Workflow—Field Mapping

ATTIVIO
The Data Dexterity Company

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

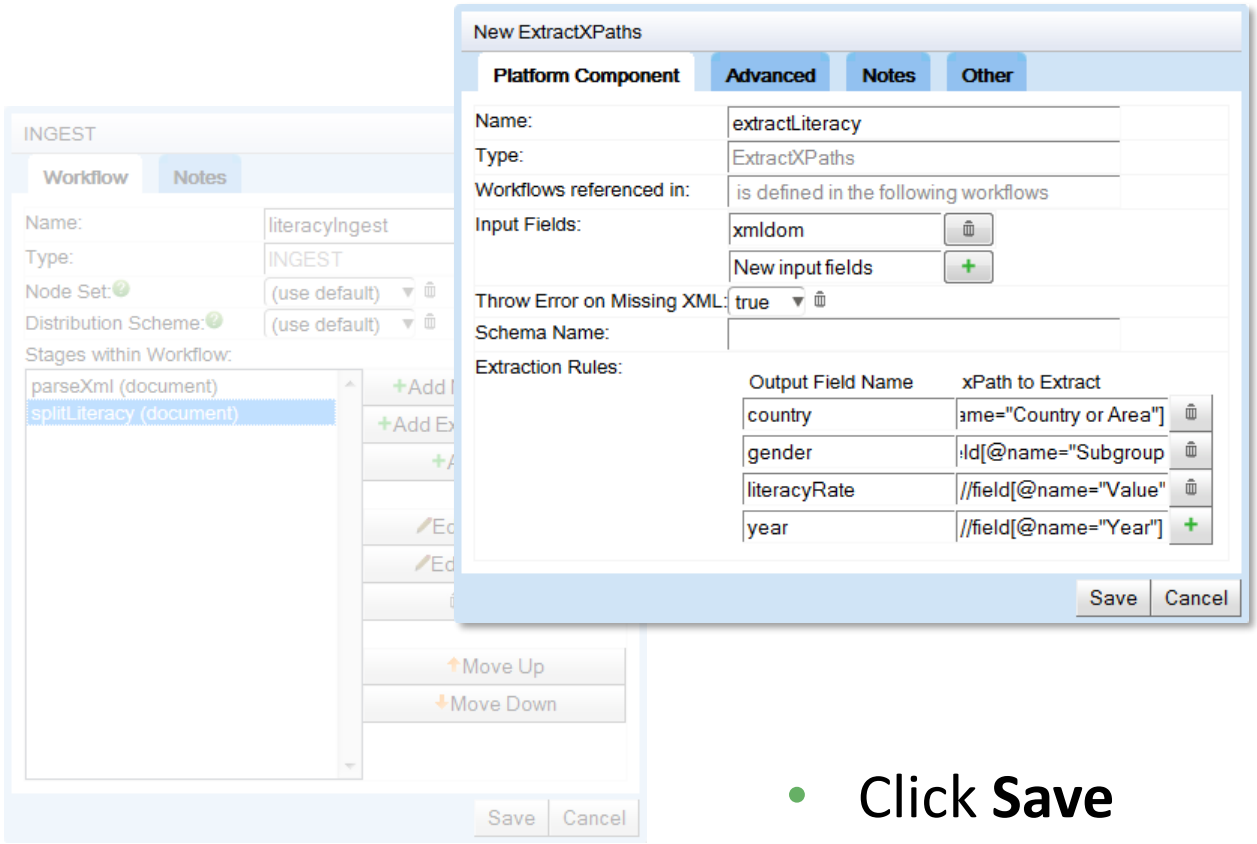
Output Field Name	Xpath to Extract
country	//field[@name="Country or Area"]
gender	//field[@name="Subgroup"]
literacyRate	//field[@name="Value"]
year	//field[@name="Year"]

*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

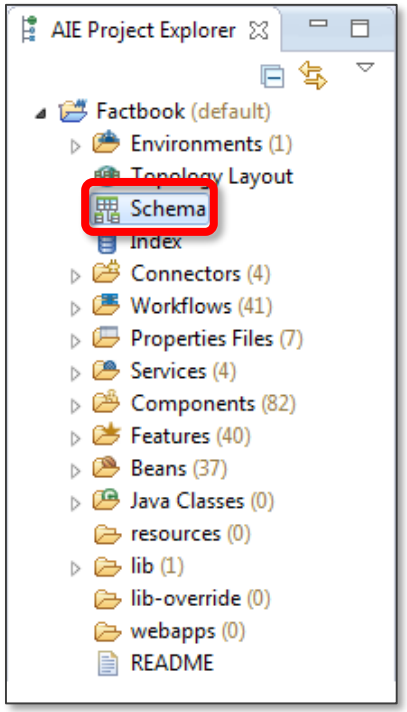
The finished component should look like this:



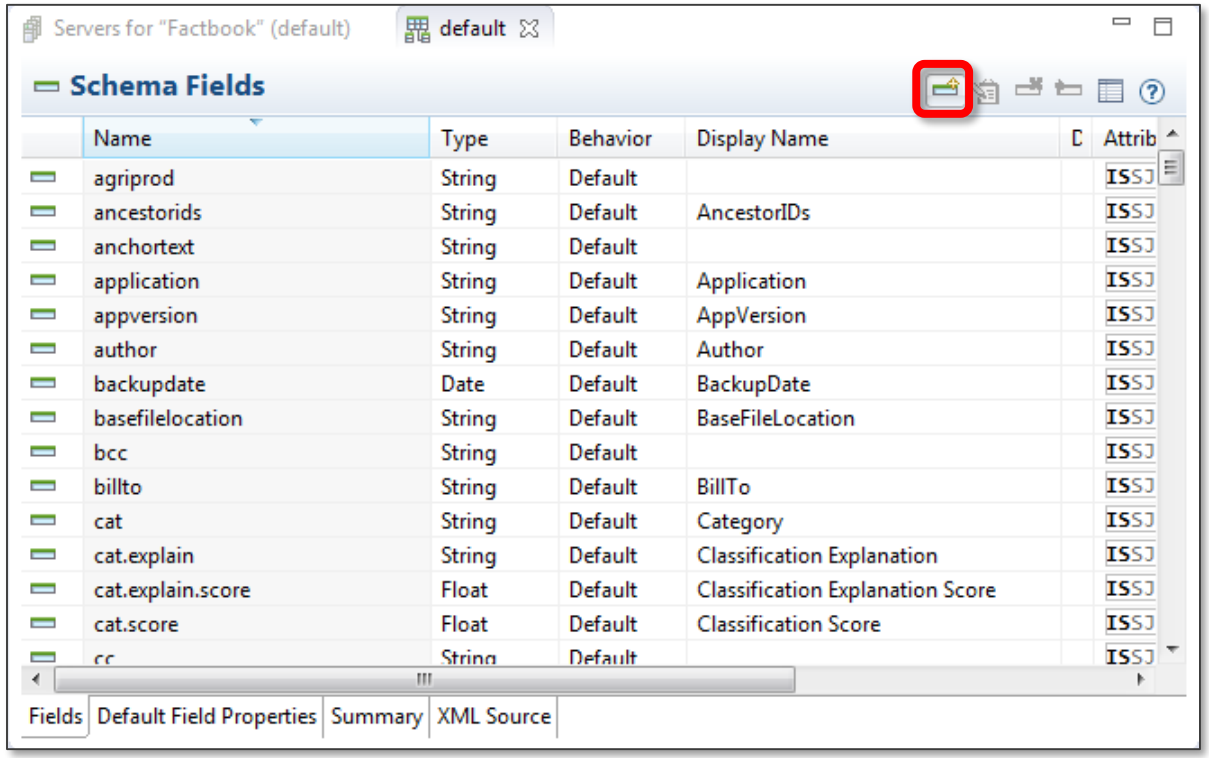
- Click **Save**

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

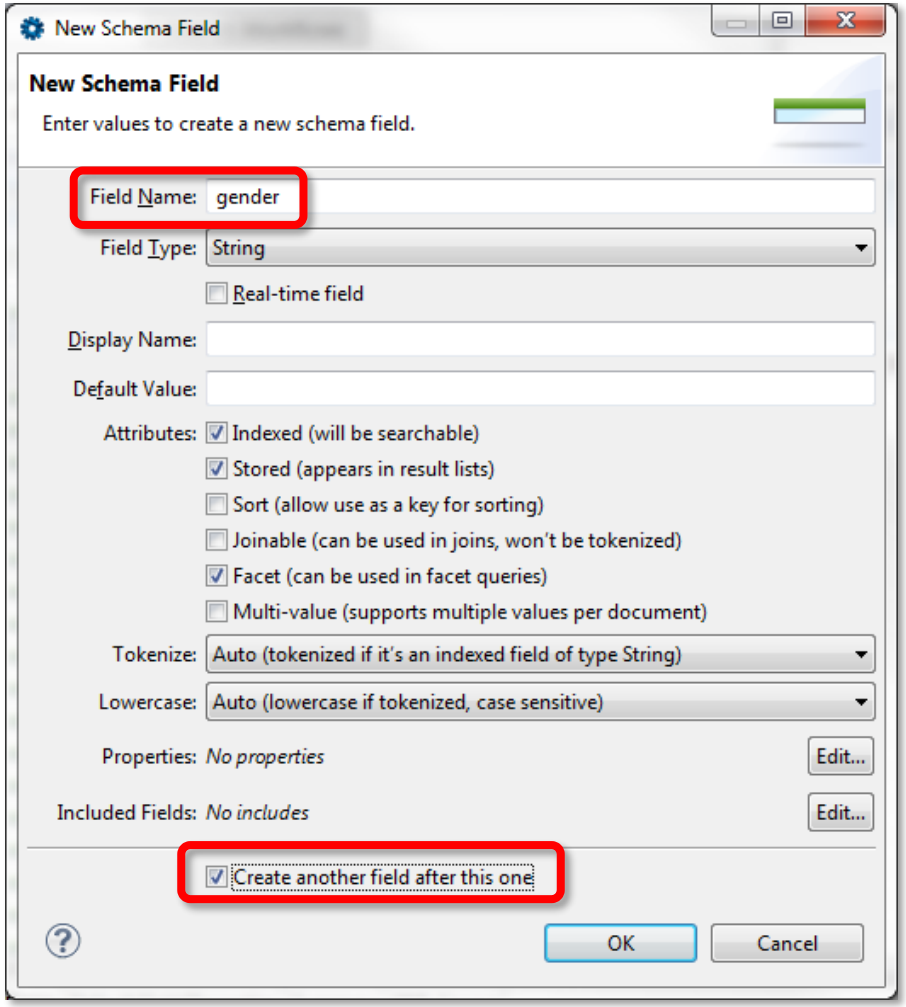
ATTIVIO
The Data Dexterity Company

- 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**



Adding to the Schema

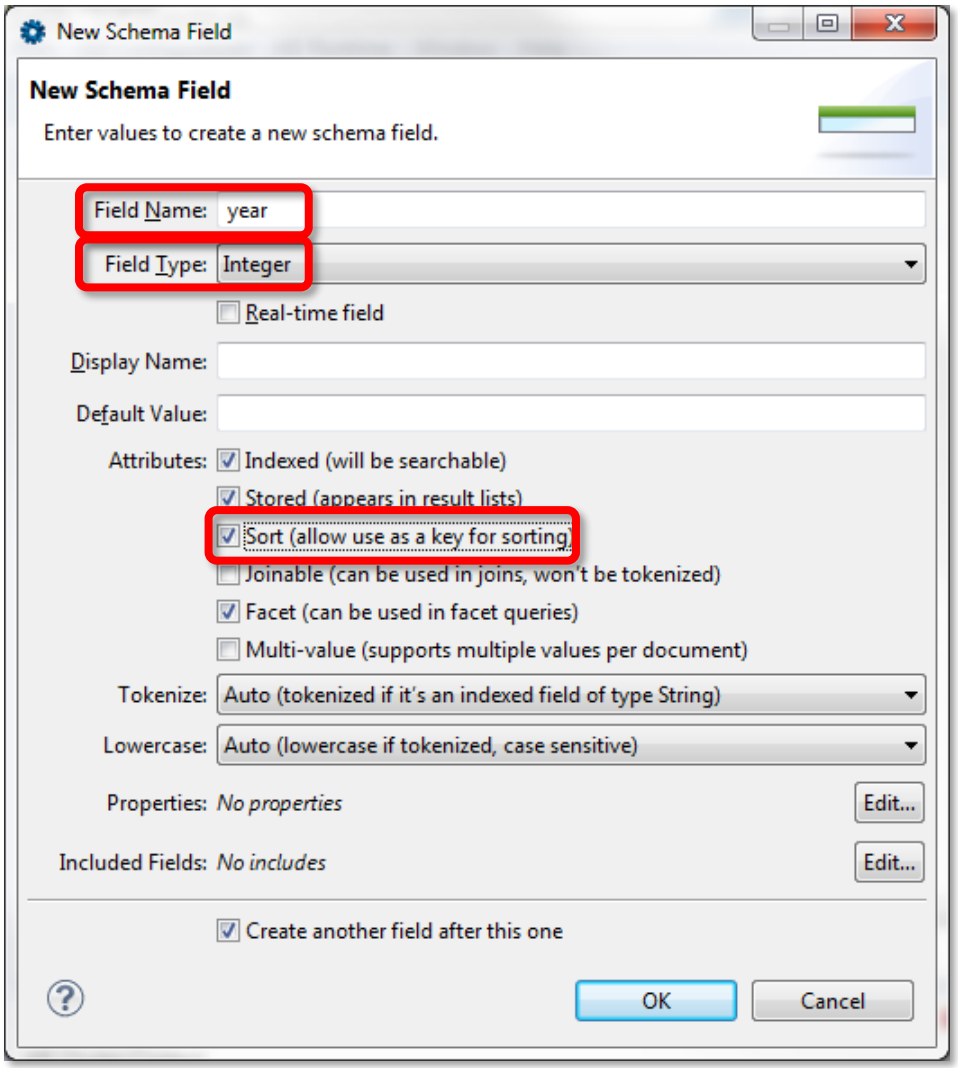
- Enter **year** as the field name
- Change the Field Type to **Integer**

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**



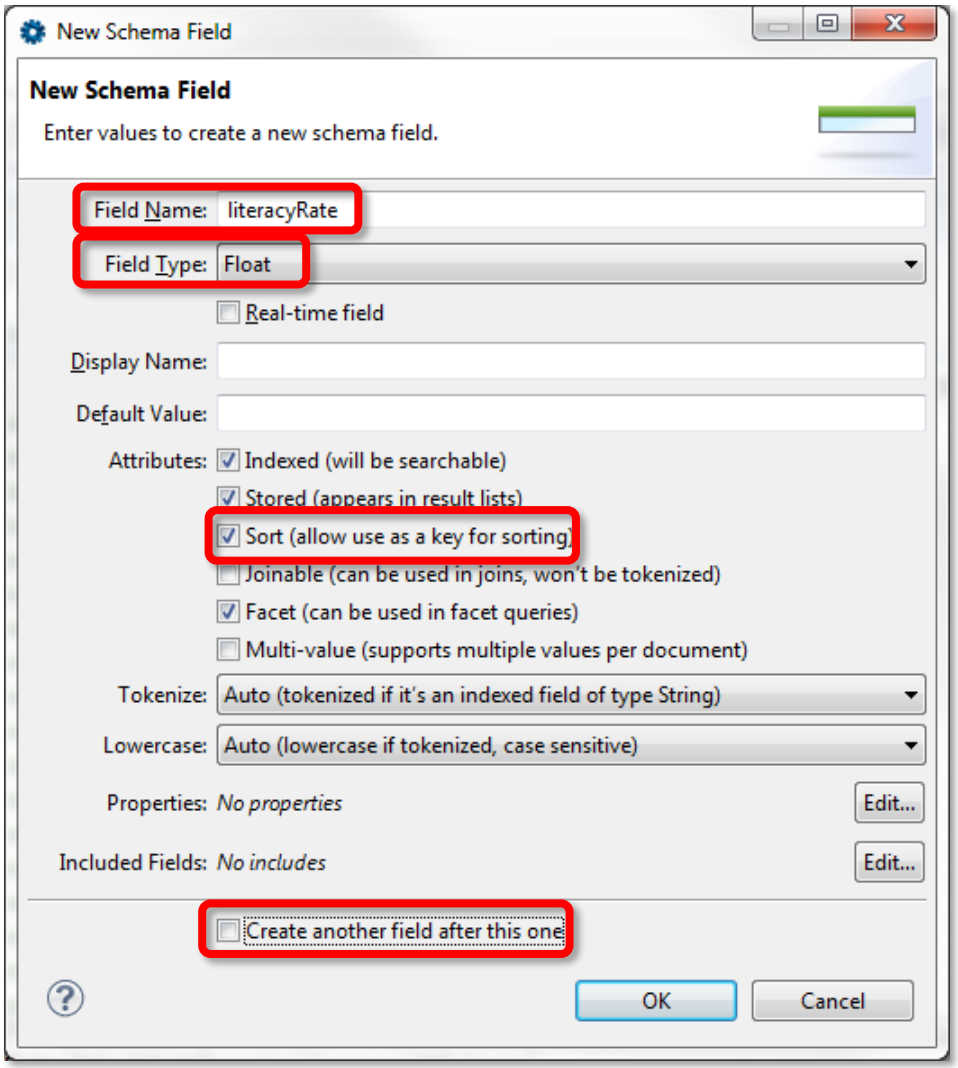
Adding to the Schema

ATTIVIO
The Data Dexterity Company

- 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**

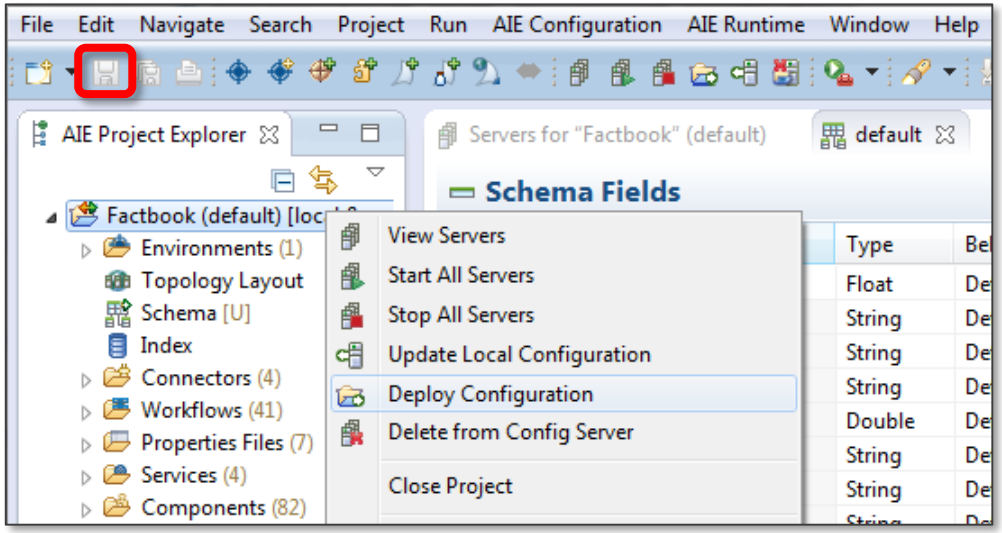


Deploying the Schema Changes

ATTIVIO
The Data Dexterity Company

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**



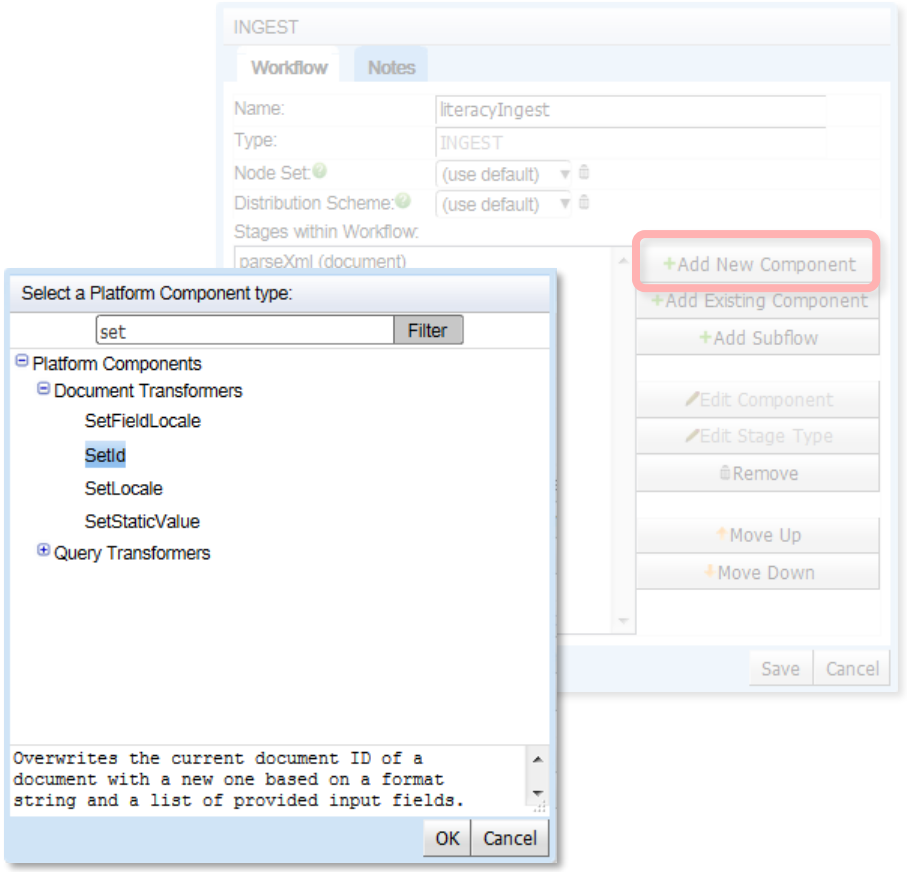
*This will shut down the **local** node, update the configuration on the server, and restart the node.*

*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

ATTIVIO
The Data Dexterity Company

- 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.*

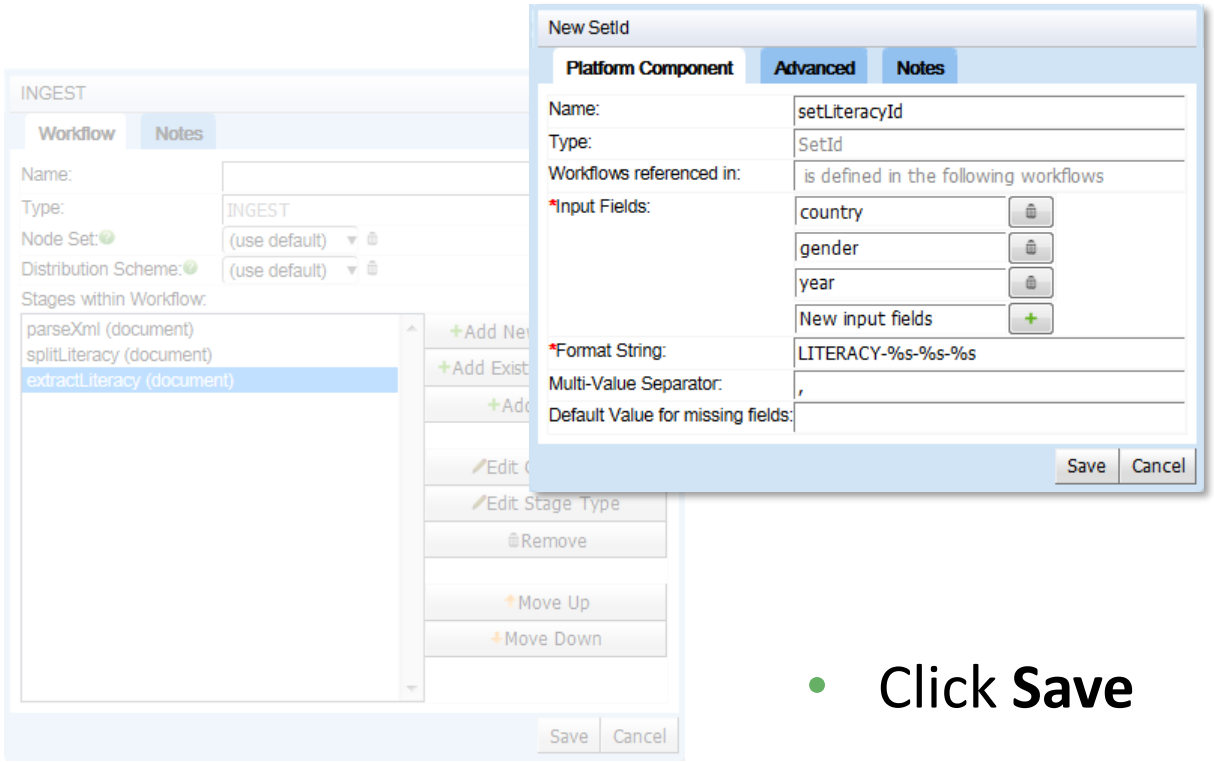
Input Fields *specifies which fields to concatenate, and in what order.*

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:



- Click **Save**

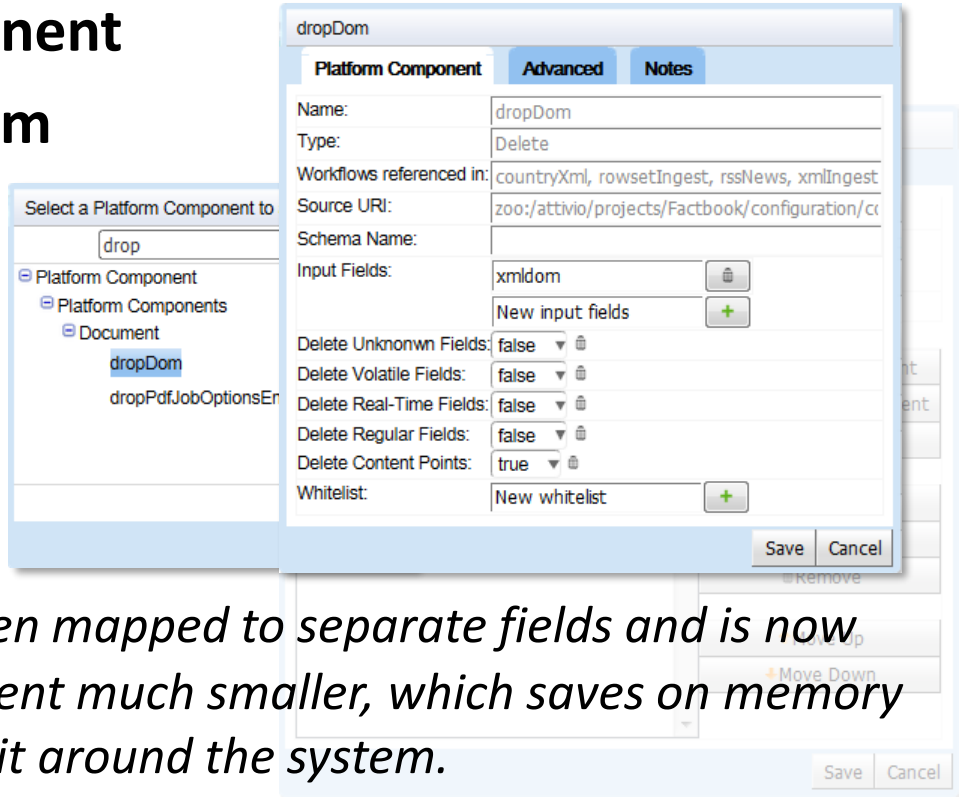
Creating a New Workflow—Drop DOM

ATTIVIO
The Data Dexterity Company

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**

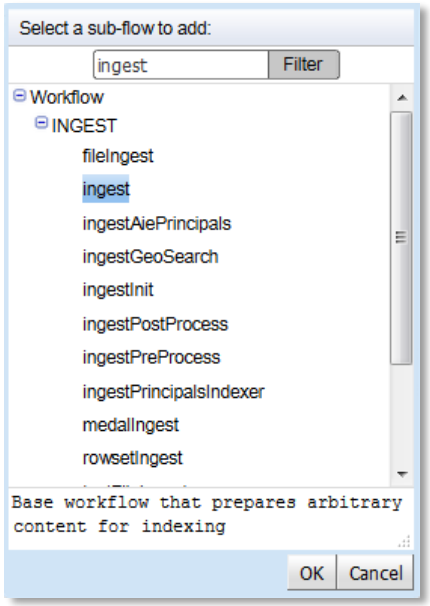


***dropDom** is a Delete component, which deletes every input field from the document. In this case, we're deleting the **xmldom** field created by the **parseXml** component, since the XML has been mapped to separate fields and is now redundant. This makes the document much smaller, which saves on memory and network traffic while passing it around the system.*

Creating a New Workflow—Indexing

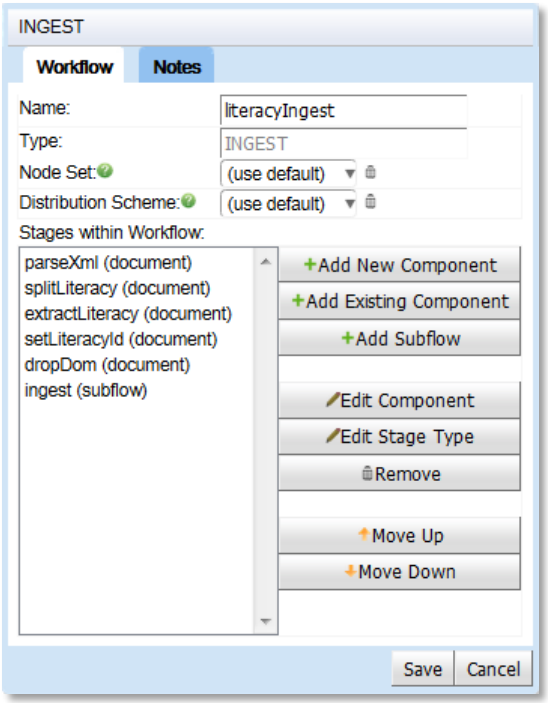
Finally, the workflow will send the documents through the standard cleanup and indexing process.

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



The finished workflow should look like this:

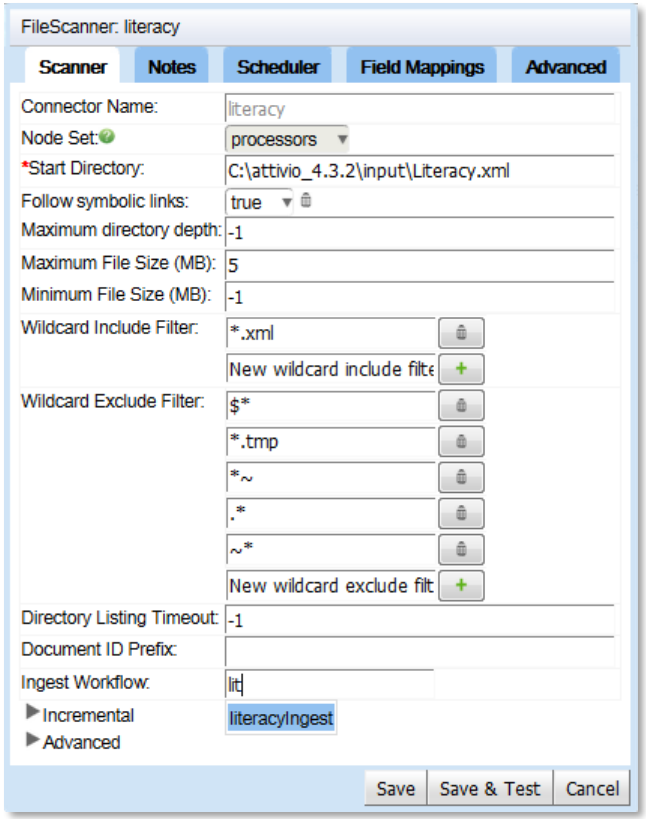
- Click **Save**



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**

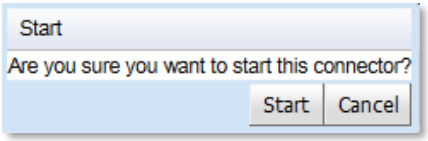


INGESTING THE DATA

Ingesting the Data

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

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

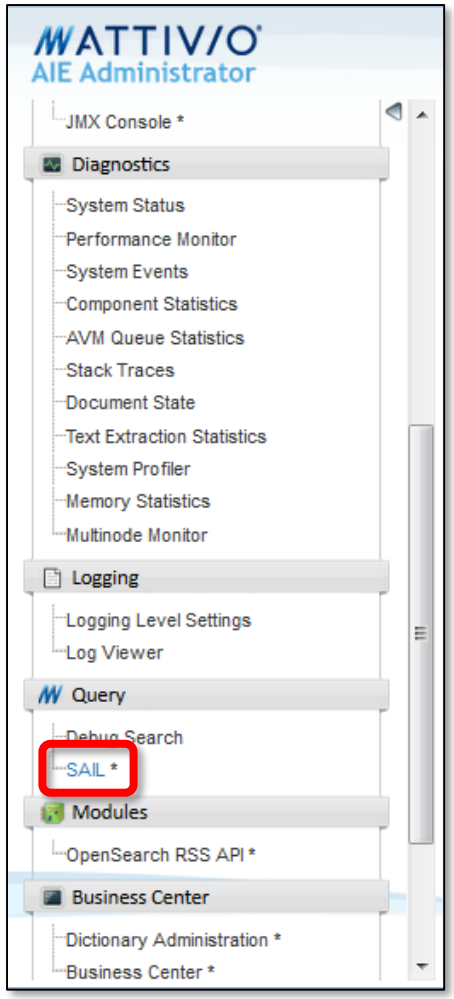


EXPLORING THE DATA

Exploring the Data

ATTIVIO
The Data Dexterity Company

- 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*

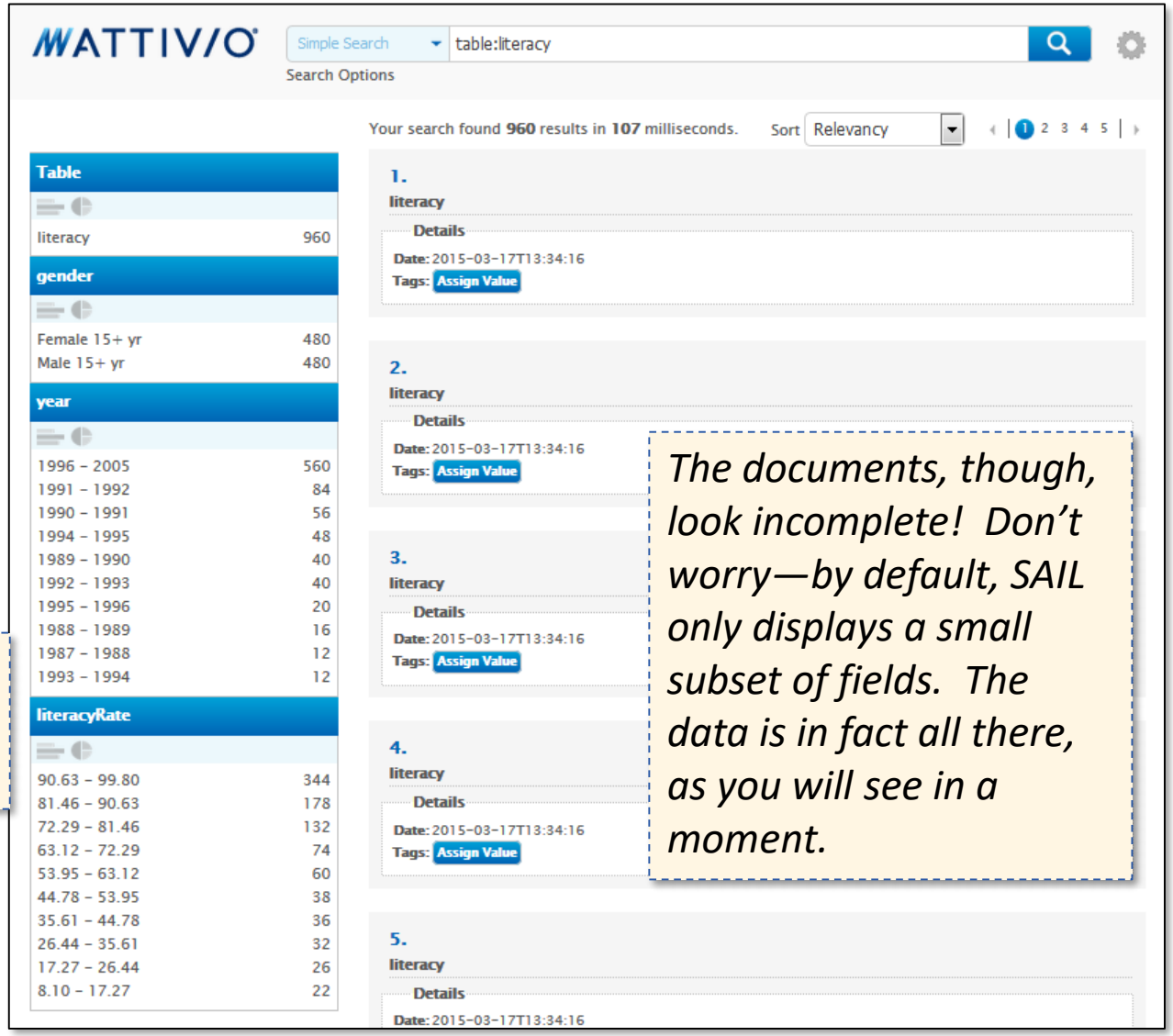
Exploring the Data

ATTIVIO
The Data Dexterity Company

- 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.

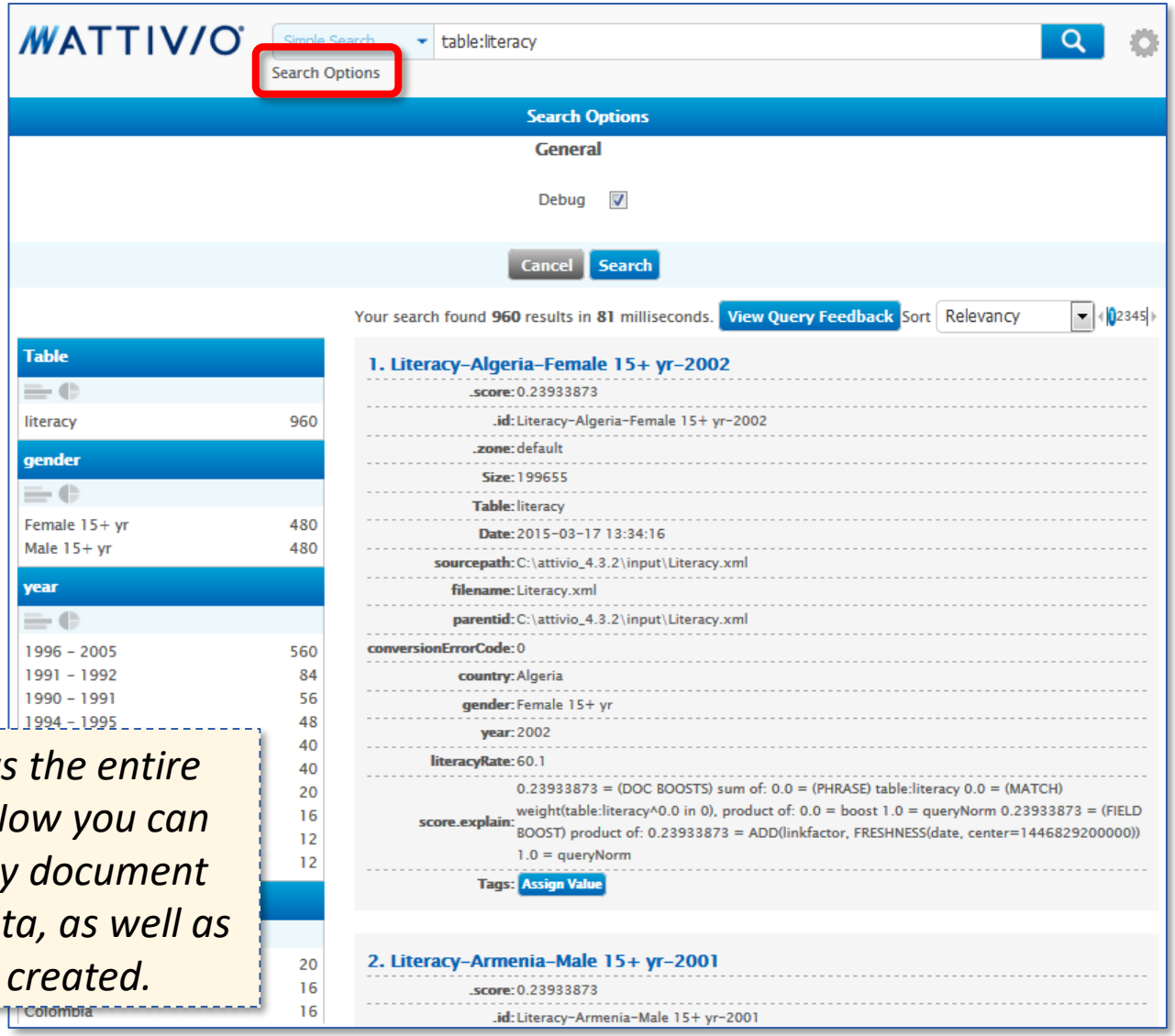


The documents, though, look incomplete! Don't worry—by default, SAIL only displays a small subset of fields. The data is in fact all there, as you will see in a moment.

Exploring the Data

ATTIVIO
The Data Dexterity Company

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

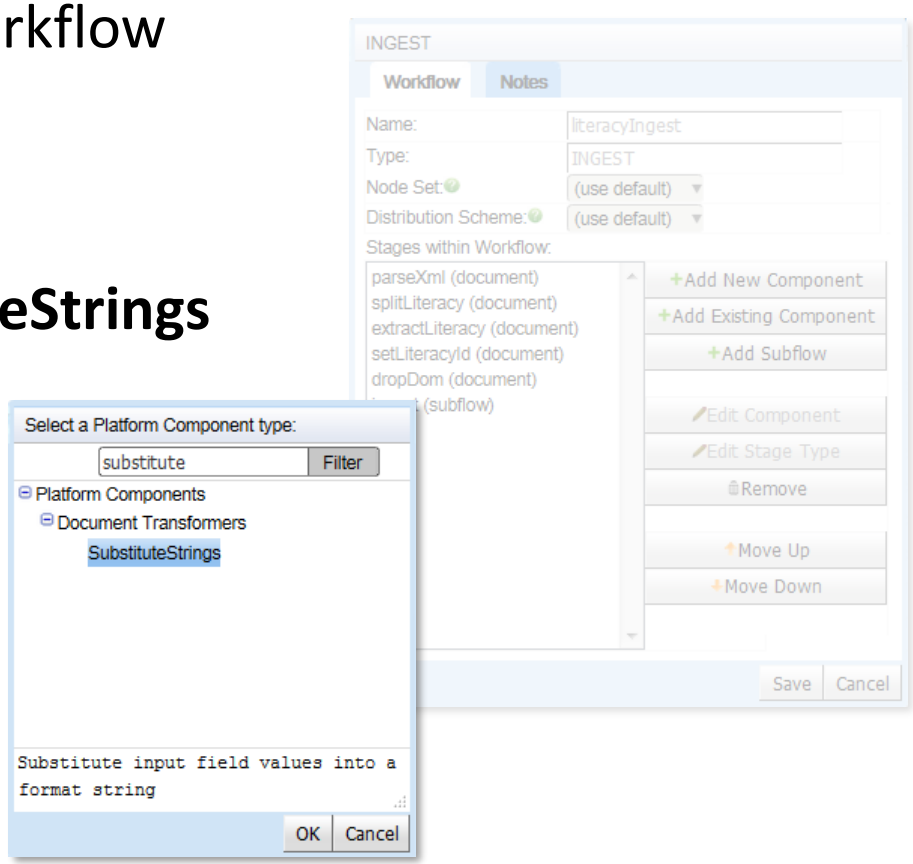


Debug mode displays the entire Attivio Document. Now you can see that each literacy document contains file metadata, as well as the special fields we created.

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

ATTIVIO
The Data Dexterity Company

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

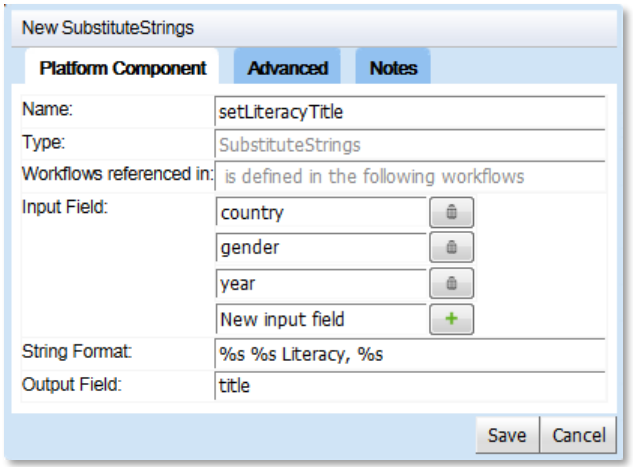
Parameter	Value
Input Fields	country
	gender
	year
Format String	%s %s Literacy, %s
Output Field	title

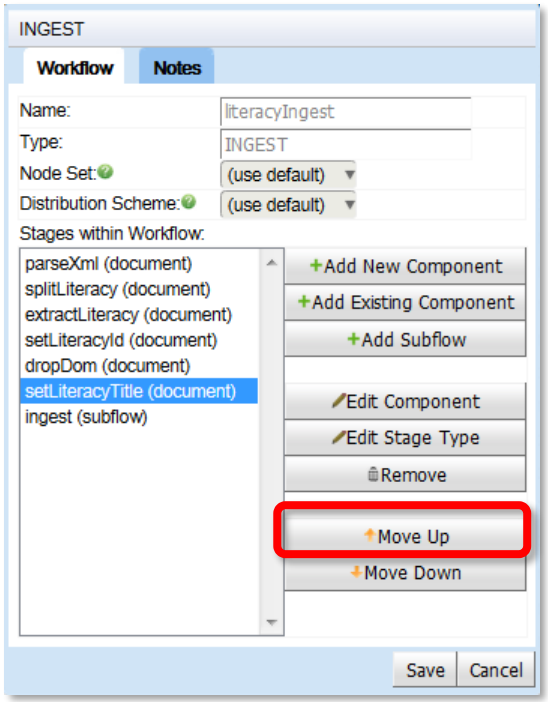
*Just like **SetId, SubstituteStrings** concatenates multiple document fields into a single string. The only difference is that you can write that compound string into ANY non-ID field.*

*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:



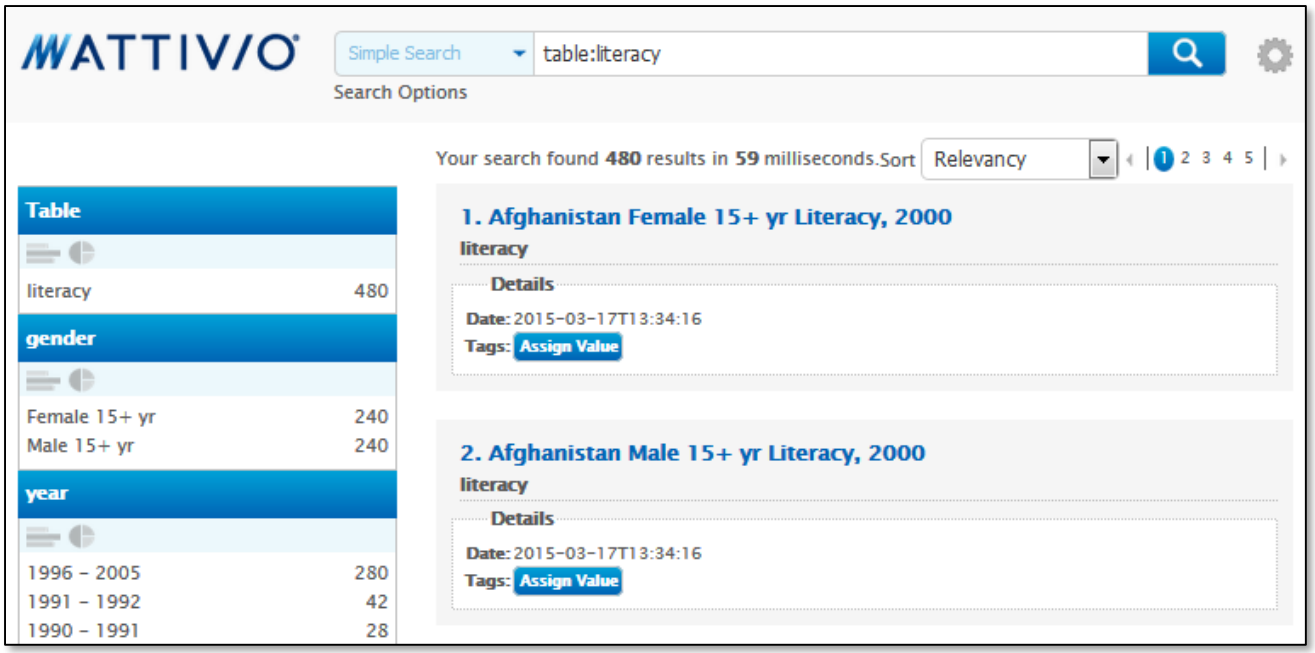


- Click **Save**
- In the Workflow window, select **setLiteracyTitle** and click **Move Up** until it is above the **ingest** subflow
- **Save** the workflow

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**

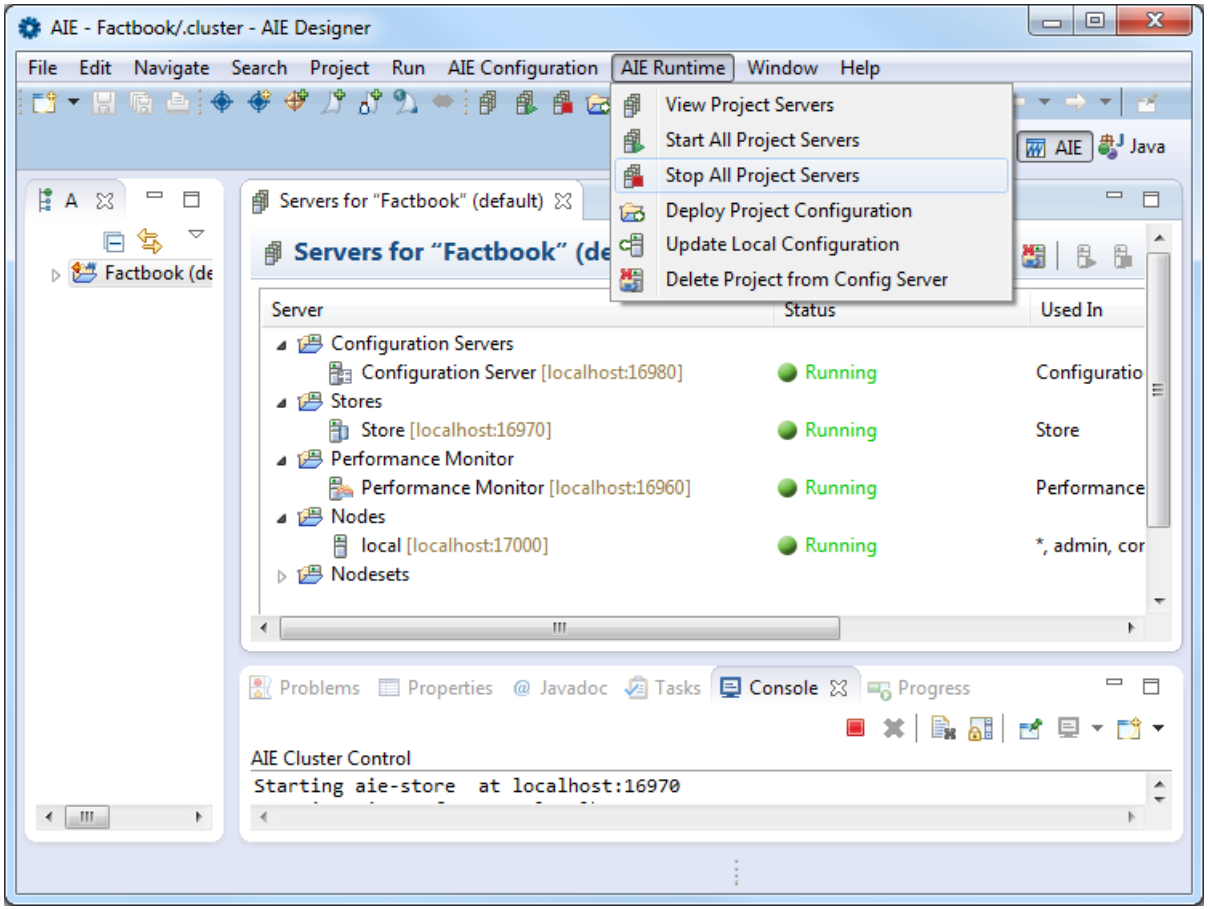


Now all of the documents have titles, improving the search experience.

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.