

# **Dell Statistica Big Data Analytics % .%**

User Guide



© 2016 Dell Software Inc.

## ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Dell Inc.

The information in this document is provided in connection with Dell products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Dell products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, DELL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF DELL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Dell makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Dell does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Dell Inc.  
Attn: LEGAL Dept.  
5 Polaris Way  
Aliso Viejo, CA 92656

Refer to our Web site ([software.dell.com](http://software.dell.com)) for regional and international office information.

## Patents

This product includes patent pending technology.

## Trademarks

Dell, the Dell logo, and Statistica are trademarks of Dell Inc. and/or its affiliates. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims any proprietary interest in the marks and names of others.

## Legend



**CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.



**WARNING:** A WARNING icon indicates a potential for property damage, personal injury, or death.



**MOBILE:** A MOBILE icon indicates that the functionality is available in a mobile application.



**VIDEO:** A VIDEO icon indicates that an instructional video is available.

Dell Statistica Big Data Analytics User Guide  
Updated - 2016  
Software Version - 13.1

# Contents

<b>Getting started</b>	<b>5</b>
Conventions and terms used in this document	5
Interface at a glance	5
Introduction and overview	6
Quick-start tour	9
Importing Statistica scoring algorithms	15
Using the HDFS browser	15
General strategy and important concepts	17
<b>Installation and administration</b>	<b>18</b>
Requirements and planning	18
Installation	20
Starting and stopping services	21
Amazon S3	22
Machine learning and predictive analysis	22
Managing licenses	22
Deploying custom processing Components	23
Supporting Quantum 4D (Q4D) visualization	25
Geospatial names	26
Interacting with Apache Solr	26
<b>Advanced analytics and search</b>	<b>27</b>
Terminology	27
Compatibility	28
Requirements	28
Installation	28
A quick tour of a Statistica Big Data Analytics Search user experience	29
Taxonomy of Statistica Big Data Analytics Search widgets	32
Hello Search! with Statistica Analytics Search	36
Using Statistica Big Data Analytics Search	37
<b>Working with Q4D Builder</b>	<b>42</b>
Setup and configuration	42
Creating your Q4D space	42
Linking spaces in Q4D	43
Q4D schema script	46
<b>Appendix A: Processing elements reference</b>	<b>54</b>
Release FHE processing elements	54
Release 2.0 and earlier processing elements	71
<b>Appendix B: Manual configuration of data nodes for GATE</b>	<b>97</b>
<b>About Dell</b>	<b>99</b>
Contacting Dell	99
Contacting Support	99



# Getting started

This chapter is a guide to getting started with Statistica Big Data Analytics. It contains information to help you conquer Big Data problems and leads you through understanding the Statistica Big Data Analytics environment, creating and running jobs, and analyzing results.

## Conventions and terms used in this document

Statistica Big Data Analytics uses a number of conventions and terms that should be explained for the sake of clarity.

### Conventions

The following conventions are used throughout this document:

URL - highlighted in blue

Bolded text - options that appear in the Statistica user interface, or commands to type into the command line, or source code

**NOTE:** - used to identify additional information provided for further clarification

### Terms

The following terms are used in the discussion of Statistica:

Processing Element (PE) - a widget in the user interface that performs some action

Connector - a line between Processing Elements that connects the two together

Sources - a special kind of Processing Element that acts as a starting point by inputting data

Sinks - another special kind of Processing Element that acts as an ending point by depositing final data

Analytics Workflow - the entire set of Processing Elements and connectors that will be converted into one or more Hadoop jobs to perform a processing need

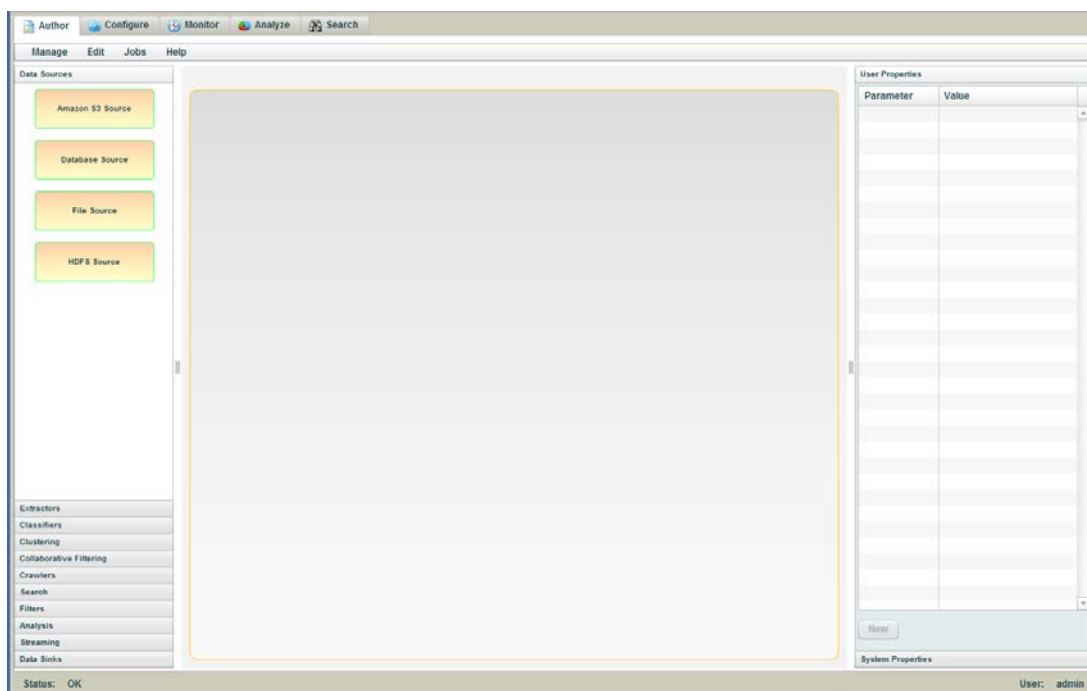
HDFS - Hadoop Distributed File System; the file system used by Hadoop to access very large files

SBDA - Statistica Big Data Analytics

\$STATISTICA\_HOME - a UNIX style representation of the home directory for the user that will run the Statistica Big Data Analytics (i.e.: /home/statistica-big-data-analytics-13.1)

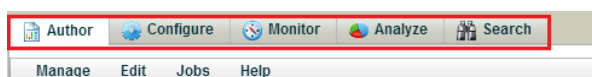
## Interface at a glance

Once the installation and configuration of Statistica Big Data Analytics has been completed and you are ready to run it for the first time (refer to the installation and configuration sections for details), the first page you will see will be similar to what is shown below.



**Figure 1:** Statistica Big Data Analytics user interface

Notice the five tabs at the top of the page. You will interact with the Statistica Big Data Analytics through these five tabs, which group collections of functions logically.



**Figure 2:** Statistica Big Data Analytics tabs

Each of these tabs provides important functionality, as summarized below:

- **Author** - create Workflows by dragging and dropping a set of Processing Elements and connecting them with Connectors
- **Configure** - browse or manage files on HDFS and manage Statistica user accounts and site configuration attributes
- **Monitor** - provides an overall view of your cluster(s) health
- **Analyze** - generates visual representations and reports of the processed data
- **Search** - provides a convenient interface into the embedded instance of Solr

You do not need all of the functionality provided by these tabs immediately. Initially, start with the **Author** tab where you can perform the processing to convert your multiple sources of data into something that is meaningful and useful. Next, consider the **Analyze** tab, where you can bring your data to life and make it presentable and truly substantive. The other tabs are indeed valuable, but the utility of them will become apparent as you become more familiar with Statistica. These will all be explained in greater detail in later sections.

## Introduction and overview

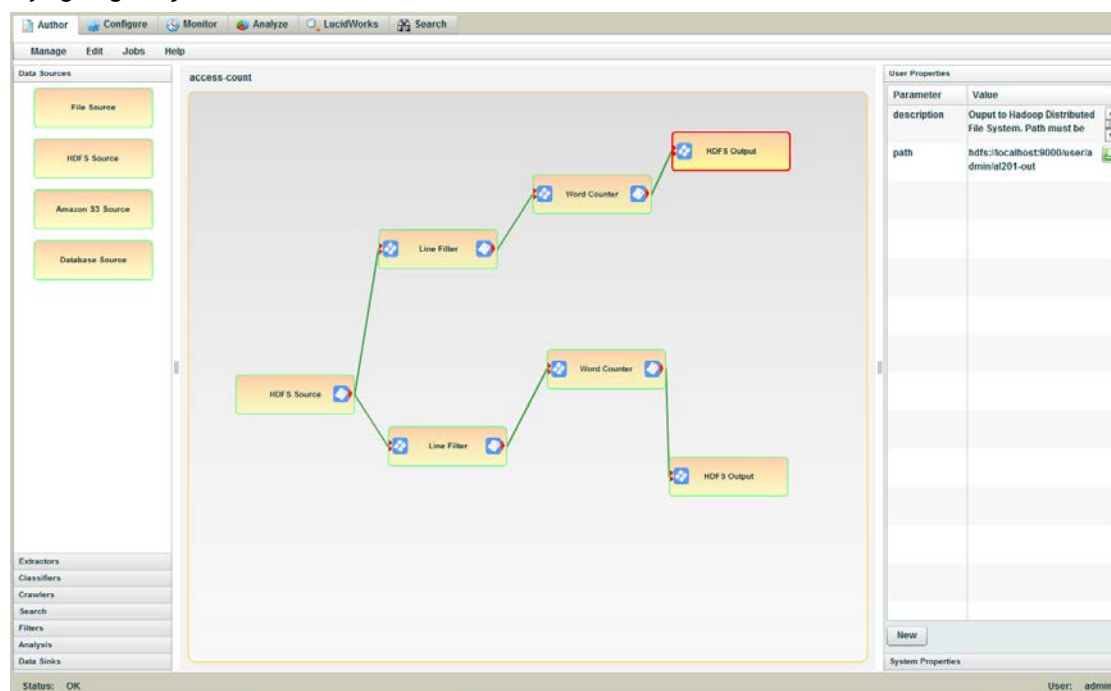
Statistica Big Data Analytics is designed to reduce or eliminate the complexity and costs associated with scaled-out content mining. SBDA supports combining ready-to-run content processing elements into full-fledged content mining solutions and then distributing those solutions over cloud and cluster-based computing assets. The entire process can be monitored and batched, and the results analyzed using sophisticated visualization and analytics tools.

With no special configuration, SBDA supports:

- Crawling remote or local sites and scraping content

- Crawling Twitter and Facebook public postings related to a topic area
- Extracting people, places, organizations, and dates, and normalizing
- Extracting links from HTML documents
- Extracting biological entities like cell lines, protein names, and DNA/RNA mentions from biological information sources
- Performing social network analysis of people, places, and organizations in document mentions
- Monitoring cluster health, nodes, jobs, and tasks
- Counting and filtering words, extracted entities, and documents
- Using relational databases as content sources and results repositories
- Charting, graphing, and visualizing content mining results
- Full text search with embedded visualizations using a faceted search engine
- Integration of the ability of a computer to learn without being programmed and the predictive analytics capabilities based on Weka and Mahout
- The ability to use multilingual language analysis components from other vendors

Figure 3 shows an example of a simple content mining application during authoring. On the left are pallets of authoring components that can be dragged and dropped onto the stage. Connectors indicate data flow between components. The panel on the right of the authoring stage shows the properties and parameters of each component. The components come preconfigured at the system level, while user-specific data paths and other properties can be set at run time. Ultimately, Statistica Big Data Analytics is not primarily a programming environment but an authoring system designed for analysts and knowledge professionals who are trying to get a job done.

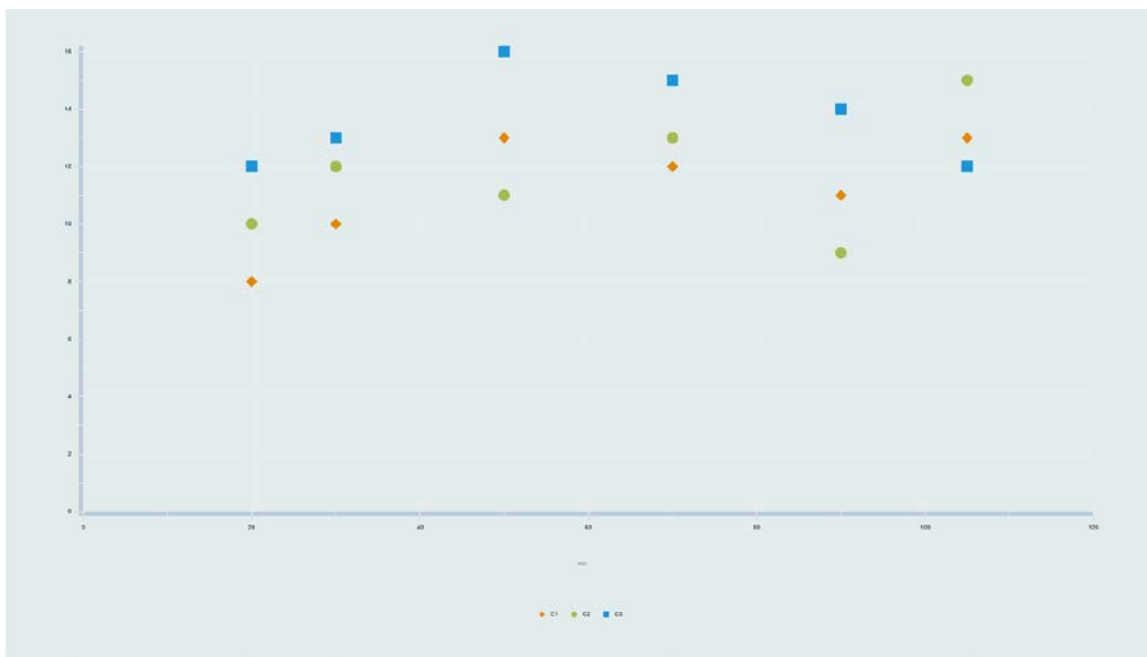


**Figure 3:** Statistica Big Data Analytics authoring user interface, showing the content mining workflow that extracts and counts IP addresses and destination pages from log files. The left panels show the available processing elements that can be added to the layout, while the right panel shows properties of the selected element. Analytics workflows can be created through configuration—without the need for programming.

Figure 4 shows a treemap visualization of biological information extracted from text. Treemaps allow for visualizations of large amounts of quantitative information so you can get an at-a-glance understanding of the relative sizes of the labeled quantities. In this case, more than 300 mentions of cell lines, genes, and proteins have been pulled out from an authored extraction workflow and the results visualized without ever leaving the Statistica Analytics environment.







**Figure 6:** Scatterplot visualization

Figure 7 shows monitoring of SBDA clusters during runs from a centralized dashboard. The gauges along the top row of the screen show at-a-glance overviews of Statistica Analytics.



**Figure 7:** Dashboard for monitoring clusters, jobs, tasks and nodes from within the Statistica Big Data Analytics application UI. Gauges provide at-a-glance understanding of ongoing processes, while detailed graphs of performance on a per-cluster and per-node basis are available below

## Quick-start tour

Getting up and running with Statistica Big Data Analytics is as easy as dragging-and-dropping components to create an analytics workflow. In this section, you will learn how to use the different types of analysis components and how to connect them together to solve a problem.

First, you must author an analytics workflow. Generally speaking, an analytics workflow consists of one or more data sources, one or more processing elements, and at least one data sink. A data source is where data comes from and is generally a database or a distributed file system. Processing elements have many different

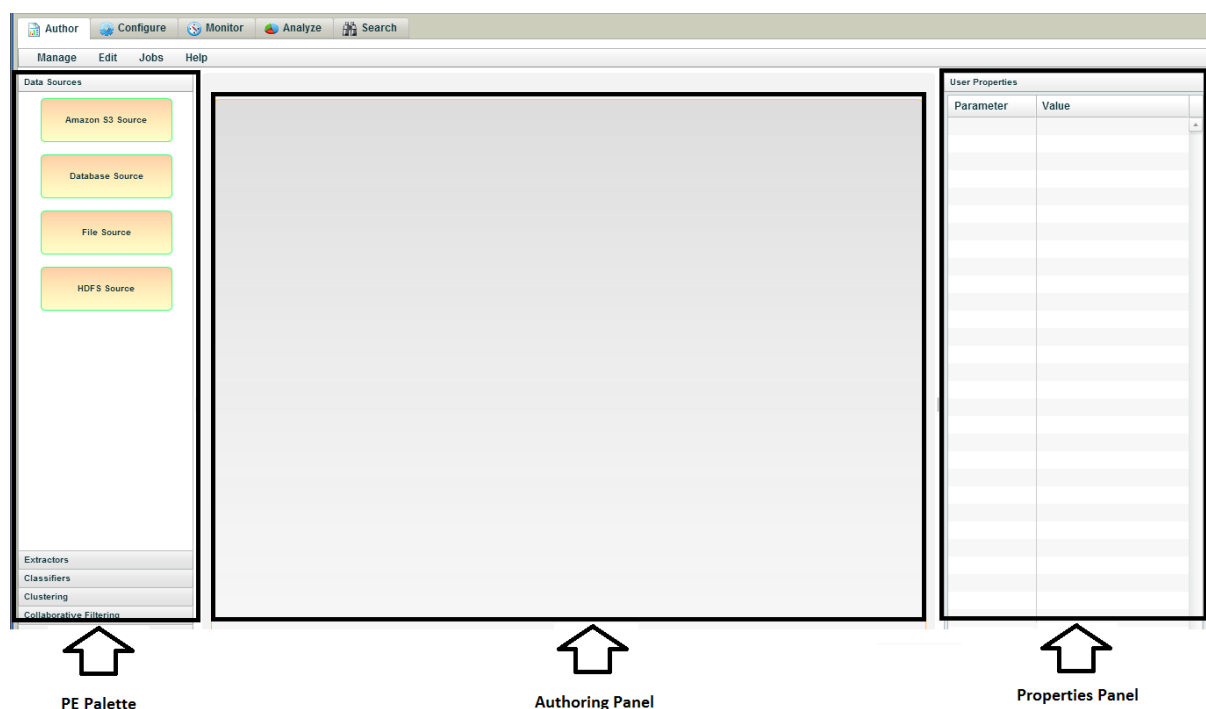
forms and types and include information extraction for text files, indexing by search engines, word counters, crawlers, and much more.

Next, after the desired data has been generated, you will then be ready to analyze that result. Here you would present the data in one or more different ways depending upon your needs. Statistica Big Data Analytics provides several different options here, such as pie charts, bar charts, column charts, line graphs, scatter plots, tree maps, and so on.

In order to get the full benefit of this section, you need to have an operating Statistica Big Data Analytics installation. Consult the installation section to learn more about how to install Statistica Big Data Analytics or contact your information technology representative.

## Authoring interface

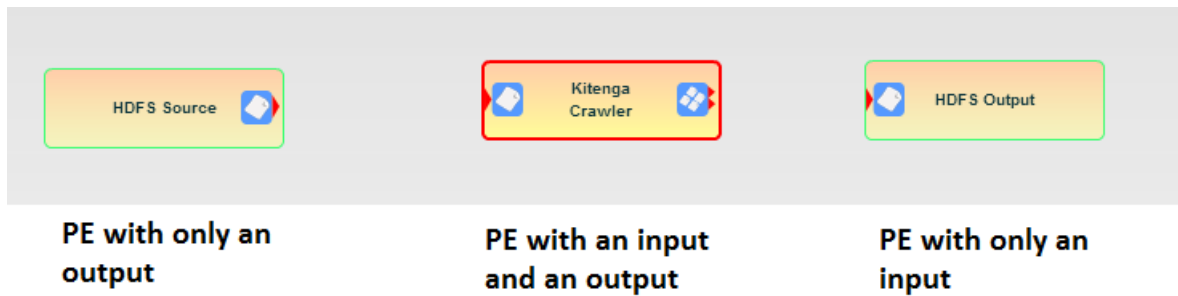
The **Author** tab is divided into three panels (Figure 8). On the left is the Processing Element palette. In the middle is the Authoring panel, and on the right is the Properties panel. Each of these panels is explained below.



**Figure 8:** Three panels on the **Author** tab

Drag Processing Elements (PE) from the Processing Element Palette to the Authoring Panel. Since Statistica provides more PEs that can possibly fit on a single page, they are divided into functional groups that are presented as collapsible subpalettes. To expand the set of PEs in one of the subpalettes, click the header, and the complete set of PEs will be exposed. Take some time to browse the different subpalettes and PEs within each and to familiarize yourself with all of them.

Once there are two or more PEs in the Authoring Panel, connect them together with a Connector. What this means is that Statistica Big Data Analytics will take the output from the first PE and direct that as input into the second PE. Looking closely at the different PEs, you will notice they have blue boxes on the left side or the right side. The blue boxes and red triangles on the left side are used for inputs, and those on the right are used for outputs (Figure 9).



**Figure 9:** Examples of PE inputs and outputs

**NOTE:** Some of the input or outputs have a single white flag and a single red triangle, indicating they use regular files, and others have four small flags and two red triangles, indicating they use sequence files.

Create a Connector between two PEs by clicking on the output side of the first PE. A red line terminated with a red circle is displayed. Continue holding down the mouse button, drag the red circle over to the input side of the second PE, and release the mouse button. The two PEs will be connected.

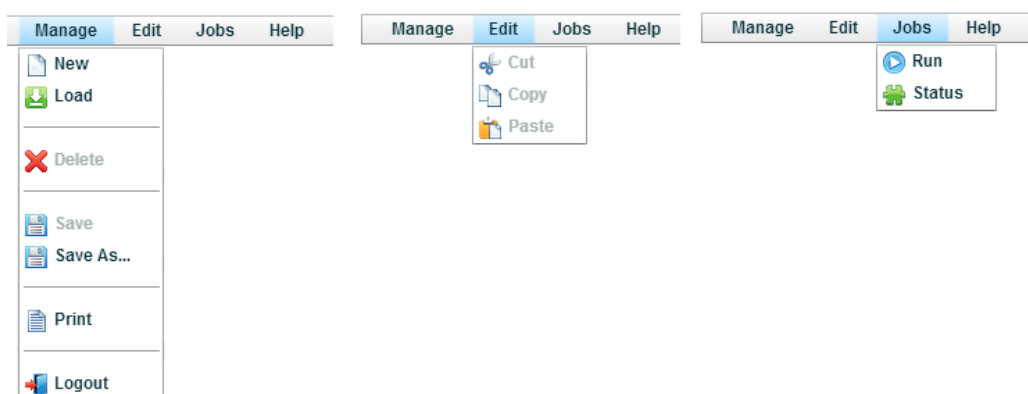
Most of the PEs have additional configuration parameters that are specific to that particular PE. This is where you use the Properties Panel. It is important to understand that the PEs will be converted into Hadoop map/reduce jobs, and the properties will be used in the submission of those jobs. Looking at the Properties panel, you see two kinds of properties: User and System. These are also organized as collapsible subpanels. Generally speaking, the User properties are the required properties that you must specify. These are required in order for the PE to operate correctly.

However, it is possible to insert additional parameters to the Hadoop job by clicking the **New** button and adding a new property. For example, it might be desirable to increase the buffer size by setting the property `io.file.buffer.size` to a much higher value, such as 131,072 (128 KB).

The System properties are for the most part configured for you and should not be changed, unless there is a good reason to do so. These are the parameters the PE requires for the operations that Statistica Big Data Analytics does for you. However, there are a few notable exceptions where you need to modify the System properties. For example, with the Database Source, you must provide your own database driver JAR, copy that into the /zetta-cache directory on HDFS, and add the JAR file name to the dependencies property.

## Operations on analytics workflows

Once the analytics workflows have been created, there is a set of operations provided. These are offered via the menus on the toolbar: **Manage**, **Edit**, and **Jobs** (Figure 10).



**Figure 10:** Three menus contain operations for analytics workflows

The **Manage** menu contains a command to create a **New** analytics workflow and clear whatever is currently in the Authoring Panel providing a clean slate. You can **Load** an existing analytics workflow from the persisted storage. The **Delete** command removes the analytics workflow from persisted storage and clears the Authoring Panel. The **Save** and **Save As** commands are used to store the current analytics workflow to persisted storage.

The commands on the **Edit** menu are used to **Cut**, **Copy**, or **Paste** one of the PEs. A PE can be cut or copied within the same analytics workflow or between two different ones, as long as the copy buffer remains active. It is limited to a single PE at this time.

The **Jobs** menu is used to submit and to monitor the progress of your analytics workflow to the Hadoop cluster. To submit your analytics workflow, use the **Run** command. This interrogates all of the PEs, converts them into Hadoop Map/Reduce jobs, and submits the jobs to the cluster in the appropriate order. An analytics workflow does not need to be saved to be run and allows for flexibility during the early stages of the analysis. The **Status** command displays a window that provides a list of the jobs that Statistica has started as well as their current status. In addition, if the job is still running, you will have the option to **Kill** it by clicking the Hand symbol on the right.

## Creating your first analytics workflow

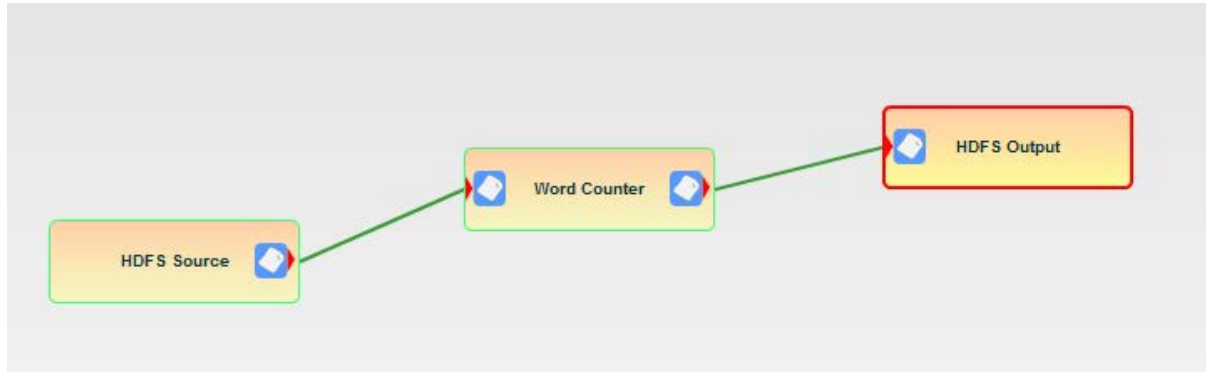
Now that you have an understanding of how to author an analytics workflow, it is time to work through an example. We will use a canonical example for counting words in a collection of sample documents. After the initial task is complete, we will discuss some important additional concepts that can help you as you begin to experiment with your own processing jobs.

### The data

For the data inputs, we can use the eight plain-text abstracts that are included in installation and that can be found in the bio subdirectory, all drawn from PubMed abstracts. PubMed is a clearinghouse for medical research abstracts run by the United States National Library of Medicine.

### Authoring an analysis


This first analytics workflow consists of at least one HDFS Data Source, one Word Counter Processing Element, and one HDFS Data Sink. The goal is to create the workflow shown in Figure 11.



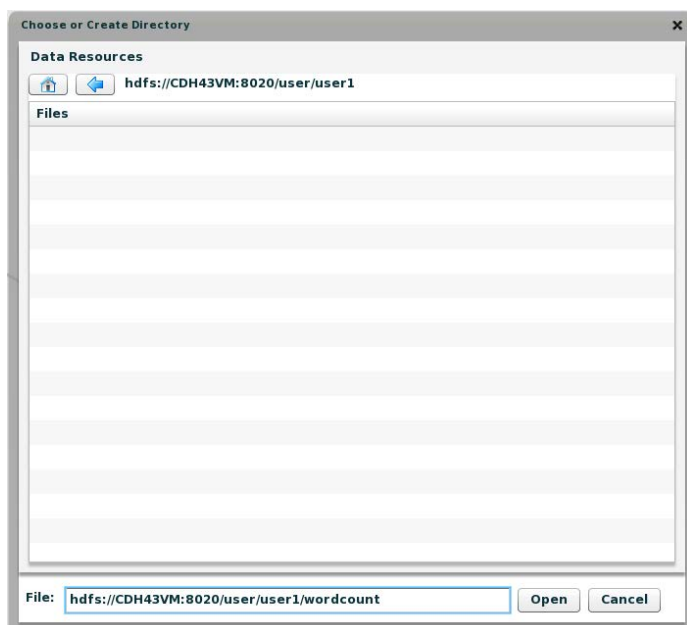
**Figure 11:** Example of an analytics workflow

The HDFS Source is in the Data Sources collection, while the HDFS Output is in the Data Sinks collection. The Word Counter is under the Analysis components.

### Create a new analytics workflow

1. From the **Manage** menu, select **New**, and assign the job a name.
2. Drag the individual components onto the Authoring Panel.
3. To connect the individual elements together, grab the document on the right side of one element (the output of the element) and connect it to the input on the next element as shown.  
**NOTE:** If you make a mistake, you can highlight the connector, select **Edit>Cut** to eliminate it, and try again.
4. After creating the analytics workflow, you need to specify which files from the Hadoop Distributed File System will be the source and which files will be the sink. Select the **HDFS Source** element, and then choose the file system browser from the **User** panel at the right: .

- Using the pop-up file browser (Figure 12), choose the bio subdirectory from the directory for the source.
- NOTE:** The **Word Counter** will work on either individual files or, in this case, directories of files. Different processing components have different input and output characteristics. Labeling conventions help to keep the types distinguishable. This will be discussed further into the guide.
- After configuring the **HDFS Source**, select the **HDFS Output** and choose the file system browser again.
  - Navigate to your user directory (/user/admin if you are logged in as admin; user1 in this example), but this time enter a new directory name such as wordcount as an extension to the path to user1 (Figure 12).




**Figure 12:** File browser for the Hadoop Distributed File System (HDFS)

This will create the word count subdirectory for the results of the process. You are now ready to execute your job. You may also save your analytics workflow at this time by selecting **Manage>Save** prior to executing your job.

**NOTE:** The file system that you are interacting with is the Hadoop Distributed File System (HDFS). A distributed file system means that the files are spread out over multiple computers and are replicated to several places in case any one computer fails. For the Getting Started implementation, there may be only one computer in your Cloudera cluster, but if there were more, the files would be replicated.

### Execute and terminate your job

- Select **Jobs > Run**, and then select the **Hadoop cluster** to execute your job.
- NOTE:** Generally there will only be one **Hadoop cluster** listed, but the system supports more than one cluster).
- The job status tracker is displayed. It will take a few moments for the system to initialize your job. This is due to Hadoop. During this time, Hadoop will create the destination directory and create a series of tracking and logging files on the system. This is part of Hadoop's commitment to restarting failing jobs and working around failing compute resources within the cluster.
  - Once running, you will see something like the following as the job progresses.
  - Use the stop button on the right as shown in Figure 13 to terminate a job while it is running.

ZettaVox Model Status			
Model	Status	Progress	
new model 1	Running	46 % [75%,17%]	

**Figure 13:** Stop button for terminating jobs

You can also monitor the cluster while the system is running by hiding the **Job Status** popup and selecting the **Monitor** tab. From the **Monitor** tab, you can observe the overall cluster status as well as the number of **Map** and **Reduce** tasks that are running. **Map** and **Reduce** tasks are the two components of the Statistica Analytics execution plan. **Maps** remap data into a different form and **Reduce** tasks collate together the data from multiple sources. In our example, the **Map** task is performing the counts while the **Reduce** task is aggregating the word counts from multiple files.

**NOTE:** If the destination directory (in this case wordcount) exists, you will be prompted to confirm overwrite of the destination directory prior to executing your workflow.

## Analyze the results

The next stage of the process is to make use of your results in an analysis. In this Getting Started example, we will look at the results of our word counting process using both a pie chart and a special visualization tool called a treemap.

### Create a pie chart

12. Select the **Analyze** tab and choose a pie chart by double-clicking the pie chart icon.

**NOTE:** Your chart is now in a subtab of the **Analyze** tab.

13. Click the **Choose** button to load the results from the word counting analysis into the pie chart. Choose the part-r-00000 file from the wordcount subdirectory of your user directory using the file system browser. The part-r-00000 file is the result of Hadoop processing. If you have a larger cluster, you may have several part-r-00000 files. These can be combined together by modifying the Hadoop settings.

14. Modify the **Field Separator** to use a **Tab** character. This ensures that spaces within the word fields are not misinterpreted as separators.

15. Type `\t` into the field next to the **Other** option button, and then select the option button.

**NOTE:** You will notice that you are already viewing a portion of the word count data, but only the first three items.

16. Change the range of the data by specifying the **Range** to be 220 through 250.

17. Click the adjacent refresh button or press return to refresh the data view.

**NOTE:** A brief popup warns you about the number of data items.

18. Dismiss the popup, and your view should look like the example in Figure 14.

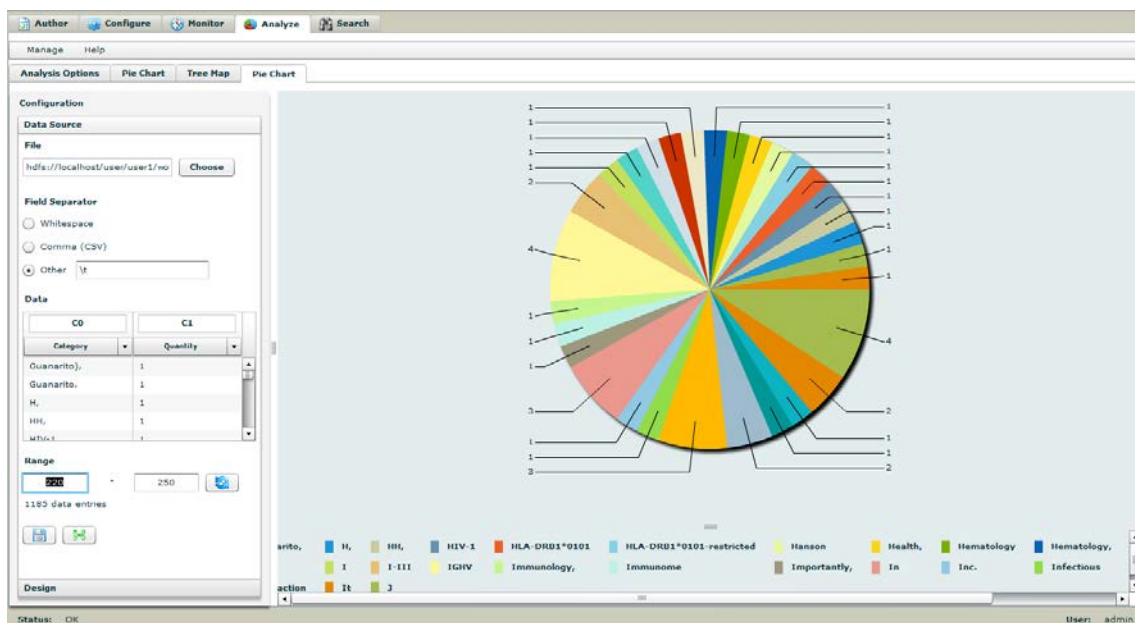
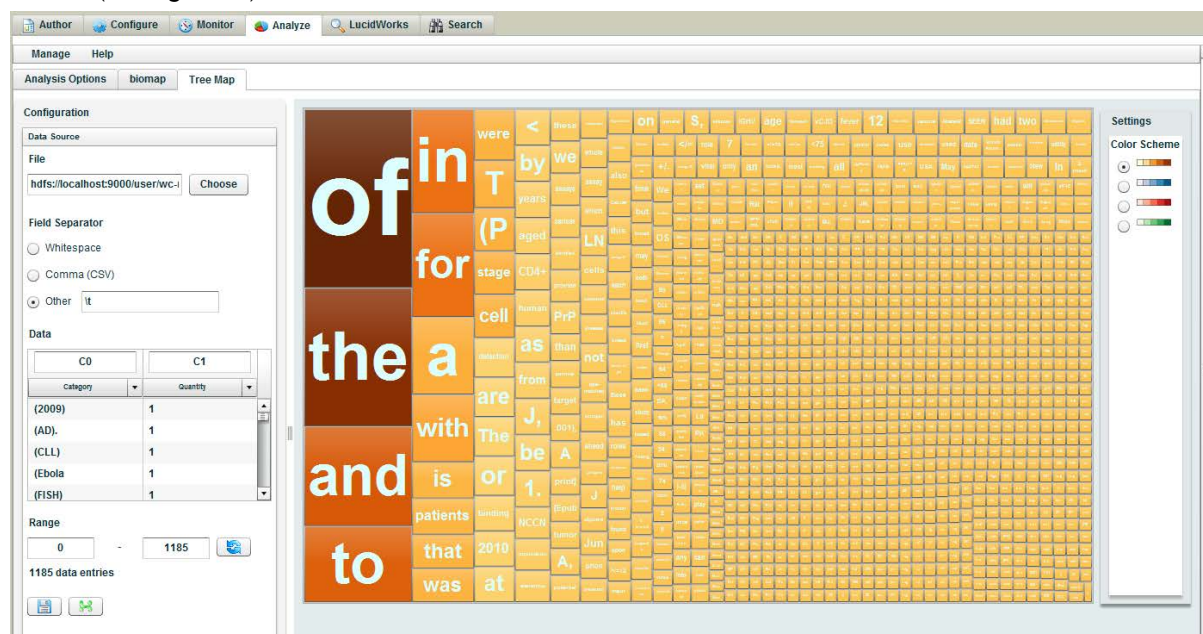


Figure 14: Pie chart example

You can download the same data to an Excel spreadsheet using the **Excel** button in the lower-left or to a local file system.

One of the limitations of pie charts for large amounts of aggregate data is that you cannot visually separate the slices. In this chart, we set our range to only 30 items from the entire set of 1185 items.

To visualize the entire data set, select the **Analysis Options** subtab of the **Analysis** tab and double-click a **Treemap**. Follow the same steps as you do to create pie chart, but now specify the data range to be from 0 to 1185 (see Figure 15).



**Figure 15:** Treemap example.

With a treemap, you can see relative magnitudes of count information at a glance and mouse over the individual cells or filter the data using the guide at the bottom.

## Importing Statistica scoring algorithms

You can easily import Statistica scoring algorithms using the `import-statistica.sh [statistica-mapreduce.java]` import script. After import, a new Processing Element (PE) will be available under the **Analysis** tab.

## Using the HDFS browser

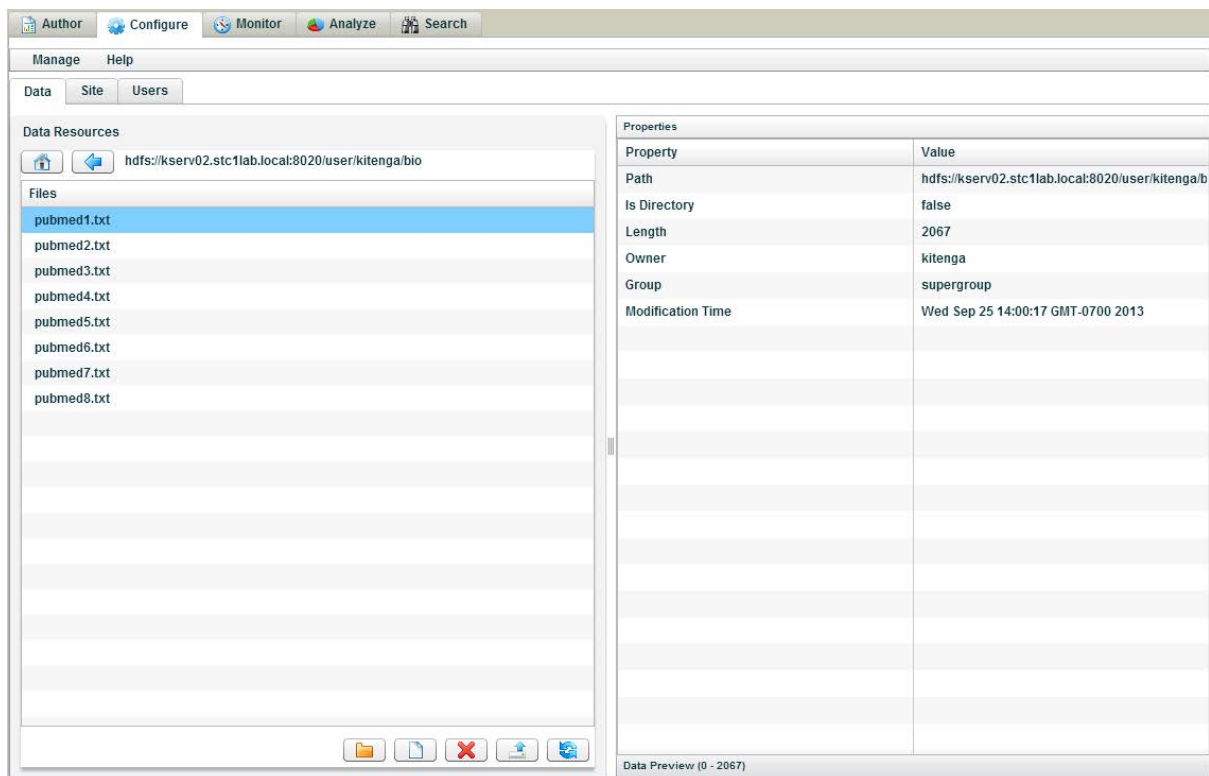
As mentioned in the previous section, Statistica Big Data Analytics provides a convenient file system browser for HDFS. The file browser is accessible from the main **Configure** tab and then the **Data** subtab. The file browser provides a great deal of additional capability beyond the default Hadoop file browser. Not only can you traverse the directory structure and view file contents as you would expect, but you can also modify the directory structure and file contents.

**NOTE:** The modification of files through the browser is only supported for regular files. Sequence files are not supported because they require some additional processing. Furthermore, sequence files may not appear correctly when being viewed. Some of the content may appear garbled or absent in the display.

The browser interface consists of two panels (Figure 16):

- file browser panel on the left
- file properties/contents panel on the right





**Figure 16:** HDFS browser panels

With the left panel, you can double-click your way around the directory structure. This is what you would expect from a file browser. However, notice five small buttons at the bottom of the file browser panel (Figure 17). These are the operations that you can perform on the file system, assuming you have the appropriate ownership and permissions. You can create a new directory or file in the current working directory. Alternatively, you can delete either a directory or file, by first selecting it and then clicking the delete button. The upload operation allows you to copy a file from your local files system into HDFS. Finally, you are able to refresh the display when the content gets stale.

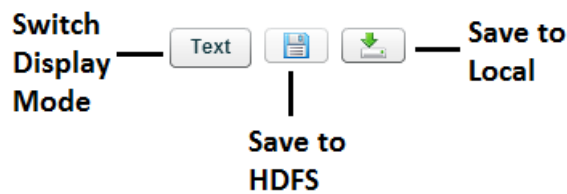


**Figure 17:** File browser operations buttons

As you traverse through the directory structure, you will eventually come to a directory that contains one or more files, and this is where the right panel becomes valuable. Notice that the panel is made up two collapsible accordion subpanels, one named **Properties** and other named **Data Preview**. You can switch between the two panels by clicking on the header.

The **Properties** subpanel provides you with information about that particular file, such the full path, the size, the owner, and so on. When you select the **Data Preview** subpanel, you will see the contents of your file in the display. You will be able to directly edit the contents of the file on HDFS right here. All you need to do is start typing. Naturally, you would only want to do this with relatively small files, due to the usual IO considerations.





**Figure 18:** Buttons for saving content and for switching display mode

At some point, you will be ready to save your changes. This can be done with the three small buttons on the bottom (Figure 18). The center button with a floppy disk icon will save the contents back to HDFS. The button to the right will save the contents to your local file system. This is especially useful when you want to capture a results file to your local hard drive. The button on the left enables you to change the display of the text between HTML and plain Text mode.

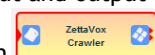
## General strategy and important concepts

There are a number of important concepts that can aid you in working with Statistica Big Data Analytics. This section can help you understand these concepts.

Although Statistica Big Data Analytics can be used to create some very large and complex analytics workflows, you should not start with large and complex workflows. The best strategy is move slowly and meticulously through each step of the workflow. First and foremost, start with a small subset of the overall data. It makes no sense to begin the authoring cycle and throw a petabyte of data at it. Next, create an initial workflow with only the first one or two PEs, output the results into a temporary location, and verify the results are what you expect. Then, add on the next PE to the workflow. Follow this by verifying the results. Gradually work through the workflow step-by-step, checking each step along the way. Once the workflow is processing the data correctly, then you can throw that petabyte at it.

- Every Processing Element (PE) needs a source (either a Data Source or the output of another element) and a sink (either a Data Sink or the input of another Processing Element).
- The Hadoop Distributed File System gets confused by special characters in file names. Stick to simple file names.
- Processing Elements operate on different types. PEs that have Multidocument in their name work internally on what are known as Sequence Files. Certain other elements also consume or create Sequence Files, such as the Statistica Analytics Crawler and XML Biological Extractor. Sequence Files contain multiple files and their header information. You can determine what sort of input and output

elements accept based on their icons. For example, the Statistica Analytics Crawler icon shows two triangles as its output and one triangle as its input. The two triangles indicate that it outputs Multidocument Collections (or Sequence Files) while consuming text files.



- elements accept based on their icons. For example, the Statistica Analytics Crawler icon shows two triangles as its output and one triangle as its input. The two triangles indicate that it outputs Multidocument Collections (or Sequence Files) while consuming text files.
- Take advantage of the file browser's ability to modify the data dynamically, especially during the early authoring stages when you are working with small data sets. This gives you the ability to quickly and easily address different use cases by simply changing the data in the browser and then re-running the analytics workflow.
- To forward documents for indexing to an external Solr search engine, the search engine needs to have a schema that accommodates the facets that are being created by your process. For the system with no special configuration after installation, the provided Solr schema only supports biological named entities supplied by the XML Biological Extractor and XML English Extractor. Modifying the schema involves changing configuration files for the Solr search engine. See the Advanced Analytics and Search chapter for more information on search.
- Never attempt to run a cluster in a virtualized environment. Although it may seem convenient to use an existing hypervisor and to create a set of virtual machines for your cluster, this will result in more problems than it is worth. Hadoop is built upon the premise of many independent hard drives. In a virtualized environment, all of the virtual machines will be using shared disk storage, and the result will be significant IO contention.

- Statistica Big Data Analytics operates as a client to the Hadoop cluster, and thus it depends upon the Hadoop configuration files to be up to date and correctly configured. However, Cloudera does not by default update these files. You must perform a manual operation in the Cloudera interface to update the configuration files, termed Deploy Client Configuration. It is extremely important to always remember to take this action when the cluster configuration changes.

## Installation and administration

This chapter is a guide for installing, running, and administering Statistica Big Data Analytics. SBDA runs on groups of Linux-based computers. The required architecture is based on one or more application servers that host a servlet. The servlet, in turn, communicates with a Hadoop cluster for executing jobs and interacting with the distributed file system, and also supports the user interface. Statistica Big Data Analytics ships with the Jetty application server and therefore does not need additional server support.

## Requirements and planning

Table A shows the required software components and version numbers needed to run Statistica Analytics. Also shown is the source for acquiring the components.

Component	Version	Source
Oracle Java JRE/JDK	7+	<a href="http://www.oracle.com/technetwork/java/index.html">http://www.oracle.com/technetwork/java/index.html</a>
Macromedia Flash Player for Browser	11+	<a href="http://get.adobe.com/flashplayer/">http://get.adobe.com/flashplayer/</a>
Cloudera CDH	5.3+	<a href="https://www.cloudera.com/content/support/en/downloads.html">https://www.cloudera.com/content/support/en/downloads.html</a>
Apache SOLR Search Engine	1.4.1	Included

**Table A:** Required components for Statistica Big Data Analytics installation

You will need to install Cloudera CDH, the Oracle Java JRE/JDK, and Macromedia Flash Player prior to the installation and setup of Statistica Big Data Analytics. For the Cloudera CDH installation, it is recommended that you use the Cloudera Manager installation option.

## Disk space requirements

Disk space requirements vary based on the goals for the deployment of the Cloudera Hadoop cluster. The Hadoop cluster uses a distributed file system called HDFS (Hadoop Distributed File System) to allocate data across multiple nodes in the cluster, thus promoting reliability in the case of node failure.

A general rule of thumb is that data will be replicated at least once. That is, there will be two copies of each file in HDFS. Also, block size in HDFS tends to be very large compared with standard file systems. The default block size is 128mb in HDFS in order to reduce seek times and is also designed to best manage larger file sizes, which are common in Hadoop processing environments. Thus, you can expect data to be at least twice and often three times the size of the original data when uploaded into HDFS.

The size of Statistica Big Data Analytics services, however, is only around 3Gb, including the user interface components. Additionally, models and log files generated by Statistica Big Data Analytics will consume additional space, though this is generally only on the order of megabytes.

## Networking and cluster configuration

A common scenario for installing Statistica Big Data Analytics is to install it initially as a single node containing all services, and then expanded it to a larger, distributed compute cluster. For either of these scenarios, operating the Hadoop cluster requires passwordless SSH to be set up. Hadoop starts and stops services using SSH and needs to contact each node (including localhost) without a password prompt. Note that when Cloudera CDH5 installs using the automated approach (which is recommended), it will require passwordless root or passwordless sudo SSH to work correctly over the cluster nodes.

To test whether passwordless SSH works for a given user, log in as that user and then try the following:

```
[user#] ssh localhost
```

If there is no request for a password in order to log in, then passwordless access is enabled. The same command can also be used to test passwordless SSH on a remote server by including the `-l root` option:

```
[user#] ssh -l root remotehost
```

To establish passwordless SSH on Linux, use `ssh-keygen` as follows and append the resultant key to the available authorized keys.

```
[user#] ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa
```

```
[user#] cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

In addition, the key needs to be distributed to all of the nodes in the cluster (along with the Hadoop codebase itself).

For a distributed Hadoop cluster, additional work needs to be done to configure the individual nodes of the cluster. The standard topology of a Hadoop configuration is to have a single Namenode that coordinates access to HDFS, recording the locations and sizes of blocks across the nodes, as well as node copies, in accord with the cluster replication policy. A secondary Namenode may also be configured to provide backup if the primary Namenode is unreachable or fails. There is also a JobTracker that monitors job execution and each node, and a TaskTracker that executes processes on the local node.

Also, configure the `JAVA_HOME` environment variable to point to the installed version of the Oracle JDK. The Statistica Big Data Analytics installer will use this variable as part of the Hadoop setup process.

## Cloudera CDH5

The Cloudera CDH5 system needs to have some additional parameters in order to work with Statistica Big Data Analytics. The parameters are applied using Cloudera Manager. To access Cloudera Manager, type the URL for the Cloudera Manager instance into a browser. If you do not know the URL, consult the Cloudera Manager installer to obtain the URL and any password information you may need. Table B shows the parameters that need to be added to the CDH5 installation.

Cloudera Manager navigation steps	Parameter
hdfs > configuration > NameNode > Advanced > Java Configuration Options for NameNode	-XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.port=25002
hdfs > configuration > DataNode > Advanced > Java Configuration Options for DataNode	-XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.port=25001

**Table B:** Cloudera CDH5 configuration changes

**NOTE:** Remember to restart the cluster after making these changes.

## Firewall configuration

In addition to the SSH access, Statistica Big Data Analytics monitors all of the Hadoop nodes in the cluster using JMX (Java Management Extensions). Note that during the Cloudera CDH5 install process, firewalls in the cluster will generally be taken offline and SELinux protections will be turned off. The following information is provided to aid in configuring the system in the advent of nonstandard Cloudera installs. All of the nodes in the Cloudera CDH5 Hadoop cluster communicate over specific ports, requesting file system access information from a Namenode and receiving job run requests through a Yarn Daemon on each node. Also, the Jetty server interacts with browsers through a specific port. To enable access to the various components, the following ports (Table C) need to be opened on firewalls within the local cluster:

Service	Role	Port
SSH	Secure shell used by Hadoop cluster to start and stop services on different nodes.	22
JMX	Statistica Big Data Analytics monitoring of Hadoop services.	25001-25002 <sup>1</sup>
Hadoop Namenode	HDFS metadata services	8020
Hadoop Datanode	HDFS data services	50010, 50020
Hadoop Yarn	Task management	8030-8033
Jetty Server	Service interaction	9100
Apache Solr™	Solr Search Engine	9100

**Table C:** Ports to be opened on firewalls within the local cluster

Additional planning around these port numbers may be required based on installation security requirements.

## Installation

Installation of Statistica Big Data Analytics is automated on Linux installations. The installation package also includes Solr and they automatically start as part of the overall system startup.

### Required prerequisite configuration on Linux

In preparation for a Statistica Big Data Analytics installation, there are a few additional steps. First, a new user named “statistica” or something similar should be created and this user added to the Cloudera hdfs and hadoop groups. The installation should then be done from the statistica user identity.

- Become the root user to execute these commands
- Create a statistica user account

```
[root]# useradd -U -G hdfs,hadoop -m -u 500 statistica
```

**NOTE:** It is highly desirable to use 500 for the user id, if at all possible. When the tarball is extracted later, this will prove helpful. However, it would be possible to overcome this with additional commands.

- Add hdfs and yarn to the statistica group

```
[root]# usermod -G statistica -a hdfs
```

```
[root]# usermod -G statistica -a yarn
```

- Allow group access to statistica’s home directory

```
[root]# chmod 770 /home/statistica
```

- Configure Cloudera to use the supergroup

By default Cloudera does not enable the hadoop supergroup, even though it does create the supergroup. You must manually enable the supergroup with the Cloudera admin console.

- Navigate to hdfs->Configuration->Service Wide->Security
- Find the Superuser Group property (dfs.permissions.supergroup, dfs.permissions.superusergroup)
- Change the value to **hadoop**

### Installing Statistica Big Data Analytics on Linux

1. Unpack the provided package and run the startup script.

**NOTE:** A common installation location is /home/statistica:

<sup>1</sup> These ports can be modified. See the section, “Installation,” for the procedure.

```
[Statistica]# tar xfvz statistica-big-data.13.1.tgz
```

```
[Statistica]# cd statistica-big-data-13.1
```

2. Install your license.

```
[Statistica]# cp mylicense.lic sbda/license.lic
```

3. Configure the runtime properties.

```
[Statistica]# cd ~/statistica-big-data-13.1/bin
```

```
[Statistica]# vi sbda.config
```

You must set properties correctly:

- `geo.db.path` - sets the fully qualified path to the GeoNames directory in your Statistica install
- `kas.max.heap.size` - sets the size of the Java memory heap
- `kas.jmx.port` - sets the JMX port of the Statistica server, the default is 24999 which is highly recommended
- `kas.server.dnsname.or.ip` - sets the host name or IP address of the server where Statistica is installed (i.e.: value returned by the `hostname` shell command)
- `hadoop.cluster.jmx.ports` - used to monitor the health of the nodes in your cluster, defaults are:

25001 - monitors HDFS on the data nodes

4. Start Statistica.

```
[Statistica]# bin/statistica-big-data.sh start
```

```
[Statistica]# bin/configure.sh
```

During the initial `Statistica.sh start` command, the system will initially start Statistica Analytics Services.

The configure script waits on HDFS startup for a minute, then copies cached libraries critical to Statistica from the install package into HDFS.

5. Navigate with a browser to log on to the system (substitute your hostname):

[http://\[hostname\]:9100/SBDA/SBDA.html](http://[hostname]:9100/SBDA/SBDA.html)

6. Enter the **User** name and the **Password**. The system is initially configured with an administration user account only:

**User:** admin

**Password:** statistica

7. Modify the `web.xml` file located in the web application subdirectory of the installation to change the admin password (note that this requires unpacking the `ZettaVoxServices.war` file):

```
$ cd app/webapps
```

```
$ mkdir ZettaVoxServices
```

```
$ mv ZettaVoxServices.war ZettaVoxServices
```

```
$ cd ZettaVoxServices
```

```
$ unzip ZettaVoxServices.war
```

```
Edit: WEB-INF/web.xml
```

## Starting and stopping services

Statistica Big Data Analytics services include the Statistica Analytics and Solr servers hosted by Jetty.

There are several reasons for stopping services, including:

- Changing configuration files for Statistica Analytics. When configuration files have been modified, the server must be restarted in order for the Statistica Analytics Service to acknowledge the changes.
- Updating the system.

#### Starting and stopping the system:

1. `bin/statistica-big-data.sh start.`
2. `bin/statistica-big-data.sh stop.`

## Amazon S3

Statistica Big Data Analytics supports retrieving information from Amazon S3 buckets and writing results to other buckets. The necessary components are Sources and Sinks available from the tools pallets within the Author User Interface. However, making use of these components requires that Amazon S3 access has been established by creating an account and downloading the necessary keys. Contact Amazon for more information on how to create and manage Amazon S3 storage:

<http://aws.amazon.com/>

Table D lists the required parameters for accessing S3 buckets.

Property	Description
aws.accessKey	Public access key
aws.secretAccessKey	Private access key
aws.bucket	Storage bucket name
aws.path	Path to bucket

**Table D:** Required parameters for accessing S3 buckets.

## Machine learning and predictive analysis

Statistica Big Data Analytics supports machine learning classification and clustering based on Weka and Mahout.

The Weka offering requires that you create a Weka model and ARFF schema file, retrieve the Weka JAR, and create an analytics workflow. The Weka model needs to have been trained with the Weka tool and then copied to your local disk to be available. The same situation will apply to the ARFF schema file. In addition, you must download Weka 3.6 and add the jar file `weka.jar` to `/zettavox-cache`. Statistica is not able to include the Weka JAR due to licensing considerations. Finally, you will need to create a workflow with the Weka Model MapReduce PE. A sample Weka model, Weka schema and input data (found in the directory `sample/diabetes`) is provided for experimentation.

The Mahout offering can be divided into three major categories: 1) Classifiers, 2) Clustering, and 3) Collaborative Filtering. A significant advantage of the Mahout offering is that it allows you to both train and perform predictive analysis. With the Classifiers, Statistica supports Naïve Bayes, Decision Forest and Breiman. For Clustering, Statistica supports K-Means, Fuzzy K-Means, Drichlet, and Min-Hash. Finally, the Collaborative Filtering supports ALS-WR.

## Managing licenses

To add a license, copy the license file to `sbda/license.lic` in the Statistica Big Data Analytics distribution. If you are replacing a license, you may need to delete the file `license-db` or move it to a safe location. Next, restart the Statistica Big Data Analytics system. If you encounter problems, you can check the `app/logs` subdirectory for messages related to licensing issues.

# Deploying custom processing Components

Sometimes you may want to do some processing actions that Statistica Big Data Analytics does not currently support. Statistica Big Data Analytics is flexible enough to allow for extensions with custom Hadoop MapReduce jobs. These would be Hadoop jobs that have been developed internally and that you want to include the processing as a custom PE in the Analytics Workflow.

## Deploying custom processing elements

1. Unpack the provided package and run the startup script.
2. Create a JAR file for your classes.
3. Modify the master copy of the site-configuration.xml.
4. Copy the site-configuration.xml to the working locations.
5. Place your custom JAR on the classpath of SBDA.
6. Restart SBDA to activate your customization.
7. Log into SBDA and create a new workflow.

For step one, you will need to have a technical resource that is very familiar with the Hadoop Suite and with programming in that environment. This topic is beyond the scope of this document and is not discussed further.

For step two, the final result will be a JAR file that contains the compiled classes in their appropriate package locations. How the JAR is generated does not matter. You have flexibility to use whichever tool desired.

For step three, you must to modify a configuration file used by SBDA. The site-configuration.xml file is used to manage the definitions of all the PEs. This is an XML file in that it uses a simple structure, thus making it reasonably easy to read and edit. However, great caution needs to be exercised when editing this file because an improper edit could result in unexpected and undefined behavior. Before making any edits, create a backup copy of the file, SBDA has a master copy of the site-configuration file in \$STATISTICA\_HOME/bin. This is the baseline version of the site-configuration.xml.

For step four, the file is copied to \$STATISTICA\_HOME/Statistica and \$STATISTICA\_HOME/Statistica/Statistica-data where the copies are used at runtime. You must make the edits to the baseline file in \$STATISTICA\_HOME/bin and then copy the edited file to the other two locations.

It is necessary to create a new entry in the site-configuration.xml for your custom PE. First, you need to create a **Resource** element. This is the element that defines each PE that will appear in the user interface. The resource needs to have the **category** attribute specified. A good choice would be to use either “**Extractor**” or “**Analysis**” for the category. It is necessary to give your PE a name with the child **name** element.

Following the name element will be a series of **NamedParameter** elements. The **NamedParameter** elements are where the specific properties of the PE are defined. They all have the same basic structure. It is necessary to define the **category** attribute. This will be either “**User**” or “**System**”, which directly corresponds to which subpanel that properties will appear in the user interface. Following that, each **NamedParameter** will have three child elements: 1) **version**, 2) **name** and 3) **value**. The version should be set to 0. The name will be used to identify that particular property, and the value will be the value of the property.

There is quite a bit of flexibility in the possibilities for what can be used in the names and values. There is a set of common parameters that are used for Hadoop MapReduce jobs. However, you are also free to use additional named parameters as are needed. The common parameters are:

class - the string representation of the fully qualified class name for the primary Hadoop job entry point, needs to always be “com.Statistica.zettavox.runner.hadoop.HadoopJobRunner”

jar - the string representation of the relative path to your custom JAR, e.g., ./lib/myCustom.jar

mapper - the string representation of the fully qualified class name for the mapper

reducer - the string representation of the fully qualified class name for the reducer

combiner - the string representation of the fully qualified class name for the combiner

outputKey - the string representation of the fully qualified class name for the output key

outputValue - the string representation of the fully qualified class name for the output value

outputFormat - the string representation of the fully qualified class name for the output format, intended for non-typical output such as sequence files

inputFormat - the string representation of the fully qualified class name for the input format, intended for non-typical input formatter such as sequence files or some other custom format

reduceTasks - the number of reducers to use for the job

outputKeyComparator - the string representation of the fully qualified class name for the output comparator to be used in the reduce phase when a custom sort is needed

dependencies - additional dependency JAR files that will be required at runtime

dependenciesCache - the SBDA established directory for known dependencies, set to /zettavox-cache, intended to be used when a Distributed Cache is needed

When the value of a parameter is a fully qualified class name, special consideration may be needed because often times the mapper and reducer classes will be embedded inside of another class as a static top-level class. In these cases, it will be necessary to use the compiled class reference. An example would be the WordCount class found in the Hadoop examples JAR. Here it would be necessary to use the fully qualified class name of "org.apache.hadoop.examples.WordCount\$TokenizerMapper". Notice, the \$ separating the embedded class from the enclosing class.

A more complete example follows:

```
<Resource category="Analysis">
  <name>Word Counter</name>
  <NamedParameter category="User">
    <version>0</version>
    <name>description</name>
    <value>
```

Count all the words in a single text document by splitting on white space and trimming off extra commas, colons, and so forth, then summing over the words.

```
    </value>
  </NamedParameter>
</NamedParameter>
<NamedParameter category="System">
  <version>0</version>
  <name>class</name>
  <value>com.Statistica.zettavox.runner.hadoop.HadoopJobRunner
    </value>
</NamedParameter>
<NamedParameter category="System">
  <version>0</version>
  <name>jar</name>
  <value>../lib/hadoop-examples-2.0.0-mr1-CDH5.3.0.jar
    </value>
</NamedParameter>
<NamedParameter category="System">
  <version>0</version>
  <name>mapper</name>
  <value>org.apache.hadoop.examples.WordCount$TokenizerMapper
```



```

        </value>
    </NamedParameter>
    <NamedParameter category="System">
        <version>0</version>
        <name>reducer</name>
        <value>org.apache.hadoop.examples.WordCount$IntSumReducer
        </value>
    </NamedParameter>
    <NamedParameter category="System">
        <version>0</version>
        <name>combiner</name>
        <value>org.apache.hadoop.examples.WordCount$IntSumReducer
        </value>
    </NamedParameter>
    <NamedParameter category="System">
        <version>0</version>
        <name>outputKey</name>
        <value>org.apache.hadoop.io.Text</value>
    </NamedParameter>
    <NamedParameter category="System">
        <version>0</version>
        <name>outputValue</name>
        <value>org.apache.hadoop.io.IntWritable
        </value>
    </NamedParameter>
</Resource>

```

Once the site-configuration.xml has been edited and copied to the working locations, you will be able to move to step five in the process. Here you will need to copy your custom JAR file into the \$STATISTICA\_HOME/lib directory, so that it will be available at runtime. Notice that the location of this JAR file and the jar parameter in the site-configuration.xml are very closely coupled together.

At this point, you will be ready to use your custom PE. All you need to do is restart SBDA, log on to the user interface, and use your new PE.

## Supporting Quantum 4D (Q4D) visualization

Statistica Analytics contains support for exporting visualizations to the Quantum 4D ([www.quantum4d.com](http://www.quantum4d.com)) 3D visualization engine. Q4D's visualization client must be installed on the local computer (though not necessarily the server computer), and interacts with Statistica Analytics Big Data resources through an intermediary database. See the Q4D chapter for more information on configuring and using Q4D with Statistica Big Data Analytics.

## Geospatial names

Statistica Big Data Analytics contains built-in support for geospatial name normalization. The Map analysis component in the Statistica Analytics user interface uses this name normalization capability to transform location names into latitudes and longitudes for mapping.

The geospatial name resolution capability is supplied as a web-based service at:

[http://\[hostname\]:9100/GeoNameResolver/gn](http://[hostname]:9100/GeoNameResolver/gn)

Table E lists the parameters available:

Parameter	Meaning	Example
q	Query	<a href="http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe">http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe</a>
best	Use only the highest population match	<a href="http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;best">http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;best</a>
minpop	Filter results to ensure that only those results that exceed this parameter	<a href="http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;minpop=10000">http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;minpop=10000</a>
countries	Filter results based on list of country codes	<a href="http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;countries=US,MX">http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;countries=US,MX</a>
Data	Add this value to the results	<a href="http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;data=myvalue">http://[hostname]:9100/GeoNameResolver/gn?q=Santa+Fe&amp;data=myvalue</a>

**Table E:** Parameters available for the Map analysis component

To configure the database for geospatial name resolution, specify the database name in web.xml (variable *path*) in

`app/webapps/GeoNameResolver/WEB-INF/web.xml`

To create your own database, use the GeoNameDB utility to convert lists from the GeoNames format (<http://www.geonames.org>) into a database:

```
% java -Xmx2048m com.Statistica.zettavox.geonames.GeoNameDB sourcefile outputdb  
countrycodes
```

**countrycodes** specifies the GeoNames category codes for filtering the input. Run the utility with no arguments for more direction on the operation of the utility.

## Interacting with Apache Solr

Included with Statistica Analytics is Apache Solr 1.4.1. Apache Solr is a search engine that supports “facetted search” where important features of the text elements can be used to create a map for interacting with the information. Retail websites generally use faceting to support drilling down on search results. For instance, after searching for “dress,” a facet or options panel appears on the right of the search results, offering brand names, price ranges, colors, fabrics, and so forth—all of which are facets of the search results.

With no special configuration, Statistica Analytics supports sending text documents with facet markup on to an external Solr search engine.

Solr is available at the following URL:

[http://\[hostname\]:9100/solr/admin](http://[hostname]:9100/solr/admin)

Solr can also be accessed via the **Search** tab of the Statistica Big Data Analytics user interface.

To create more sophisticated user experiences based on indexed content and facets, see the included Statistica Big Data Analytics Search chapter.

# Advanced analytics and search

Statistica Big Data Analytics (SBDA) Search is designed to reduce or eliminate the complexity and costs associated with creating rich and compelling search experiences. SBDA Search consists of a collection of functional widgets that bind to capabilities of the open source search engine, SOLR. With SBDA Search, creating a beautiful and functional search page does not require extensive experience with web toolkits or familiarity with the details of search engines. We have instead replaced programming with configuration as much as possible.

With no special configuration, SBDA Search supports:

- Autosuggest search boxes that can be configured to interact with special document properties
- Getting images and other information related to the search index from external servers
- Charting, graphing, and visualizing search results
- Drill-down for complex document facets
- Sorting and ordering search results
- Timelines and sliders for numerical and date information related to documents or other indexed resources

SBDA Search is not primarily a programming environment but an authoring one designed for knowledge professionals who are trying to get a job done. Still, the required skill set for composing and “skinning” the search experience is one that requires a fair knowledge of HTML, CSS, and the structure or “schema” of the underlying search engine.

In this section, you will learn the essential features of the SBDA Search system, including how the system interacts with the underlying search engine. You will also understand how to configure your own search experience and where you can find more information to maximize your productivity.

## Terminology

Table F lists the terminology used in this guide for different aspects of the search experience and different ways of referring to search engines. You may find it useful to familiarize yourself with these terms.

Term or Phrase	Meaning
application server	A server engine that provides application services for dynamic web experiences.
autosuggest	A feature of search engines that recommend search terms and phrases to you as you type.
CSS	Cascading Style Sheets, a specialized configuration language for assigning skinning web pages.
custom tag library	Specialized XML-based language for functions in JSP.
facetted search	A search interface that supports drilling down on search results based on metadata displayed as facets. Also called “guided search.”
Hadoop	Open source distributed computing platform. For more information: <a href="http://hadoop.apache.org">http://hadoop.apache.org</a> . Hadoop is used to process large amounts of information in parallel.
index	The compiled database of relationships between terms and documents. As a verb, it refers to the processing of creating that database.
Jetty	Stand-alone and embeddable application server.
JSP	Java Server Pages, a programming language for creating dynamic web experiences, generally hosted by an application server like Tomcat.
Lucene	An open source search engine toolkit. For more information, see: <a href="http://lucene.apache.org">http://lucene.apache.org</a> .
metadata	Data, including text that is about a document and therefore distinguished from the document itself. As an example, the date a news article was written is metadata about the document.
named entities	Phrases and words that reference entities, like people, places, and organizations.

named entity extractor	A software system that automatically finds named entities in text.
query	The terms and search operators that constitute a search request.
search engine	The running software that is used to index documents and respond to queries.
skinning	Assigning colors, borders, highlighting and other “skin” features to the web page.
Solr	An open source search engine that uses Lucene as its core. For more information, see: <a href="http://lucene.apache.org/solr/">http://lucene.apache.org/solr/</a> .
spelling correction	A feature of search engines that recommend spelling corrections to queries.
Tomcat	An open source application server. For more information, see: <a href="http://tomcat.apache.org">http://tomcat.apache.org</a> .
Statistica Big Data Analytics Search (SBDA Search)	Statistica’s search design solution for creating advanced search experiences.
Statistica Big Data Analytics Search Designer	Statistica’s forthcoming drag-and-drop interface designer.
Statistica Big Data Analytics	Statistica’s Hadoop-enabled content mining and search indexing solution. Statistica Big Data Analytics ships with SBDA Search, also.

**Table F:** Important terminology related to search.

## Compatibility

SBDA Search works on Linux, Mac OS X, and Windows XP/Vista/7/8. Other platforms such as Solaris are possible, but untested.

The SBDA Search package is a self-contained demo system that contains the Jetty application server, Solr, and an index of 12,000 sample documents.

Also included is a second demo that can perform sentiment analysis on Facebook and Twitter data and index the content and metadata.

## Requirements

The only software required to use Statistica Analytics Search is Java 1.7 (Version 7) or later. This can be either just the JRE or the JDK. The Java system can be installed from:

<http://www.java.com>

If you are installing Java for the first time on Linux systems, you may need to configure paths to the bin directory of the Java installation. Consult the Java installation guidelines for more information.

For the demonstration system, the host computer should have a minimum of 4GB of RAM and several hundred GB of disk available. The system can run comfortably on even a modest modern laptop for demonstration purposes. For larger collections and server installations more planning is needed.

## Installation

SBDA Search comes preinstalled as part of Statistica Big Data Analytics installations. Also, several sample systems are provided.

### Start the demo services

1. % `cd $SBDA_HOME/search/kas-search/bin.`
2. % `kitenga.sh start.`

The services are automatically started at port 9200 on the local host and include both the SBDA Search environment and the Solr search engine. Note that if your firewall is aggressively blocking access to ports on the host computer, you may need to contact your system administrator to allow access to port 9200.

If port 9200 is unavailable due to a conflict with other services, you can modify the SBDA Search port number by changing the port number in `app/etc/jetty.xml`. You will also need to change the port number that SBDA Search looks for Solr in `app/webapps/ZettaSearch/WEB-INF/web.xml` to the alternative port number.

You can access the SBDA Search interface, here:

[http://\[hostname\]:9200/ZettaSearch](http://[hostname]:9200/ZettaSearch)

*Stopping the Statistica Analytics Search system:*

- `% bin/kitenga.sh stop.`

## A quick tour of a Statistica Big Data Analytics Search user experience

SBDA Search is a designer for search experiences that can be combined to achieve remarkable and unexpected insights into information resources. You are no longer forced to make sense of complex information by reading through search results lists or guessing about their relevance based on tiny snippets. Instead, all of the search results are automatically analyzed to create graphics, charts, timelines, and other features that describe the entirety of the documents that fulfill the query. These analytical tools provide several advantages over results lists. First, the tools provide an overview of hundreds or thousands of documents at once, giving a broad overview of the collection of documents that fulfill the query. This supports serendipitous discovery. Second, the tools can be used to drill down into the results set to find the best documents that answer the searcher's query.

Over time, the range of tools is expanding to include new capabilities based on social network analysis, visualization, and related capabilities that serve unique business and organizational needs. The tools can also, in many cases, tie to information resources that are not indexed but are kept in other repositories such as file systems or relational databases.

Figure 19 shows a demo implementation of SBDA Search that is supplied with no special configuration. Some of the capabilities in the interface include:

- Query automatic suggestion using a dynamic drop-down search box. This capability can be tied to different parts of the indexed data, including specific metadata fields such as people, places, and organizations.
- Spelling correction based on indexed content. This differs from spelling correction based on external dictionaries. Instead, possible matches are drawn directly from the search index, which guarantees that the spelling correction option will produce hits if chosen. An external dictionary does not provide that guarantee.
- Geospatial mapping of location named entities within the search results based on Google maps. All of the searchable locations are projected onto a Google Map and can be clicked on to drill down on the search results.
- A date timeline that gives a view of the range of dates extracted from the documents and can be dragged to filter documents based on the dates. In this case, the dates extracted from the text are used, but individual document dates related to date of publication can also be used.
- A pie chart that is tied to the people mentioned in the articles and supplied with a legend. The color scheme of the chart is specified to "purple" and the individual pie slices can be selected to drill down on the results data set.
- Facetted metadata is also rendered in the individual results, per document. In this case, the top four locations mentioned in the document are shown to the right of the highlighted search result snippets.



space, and so forth. SBDA Search functional widgets all follow a similar layout plan that is built around an outer block container, a title block, an inner container, and possibly additional features.

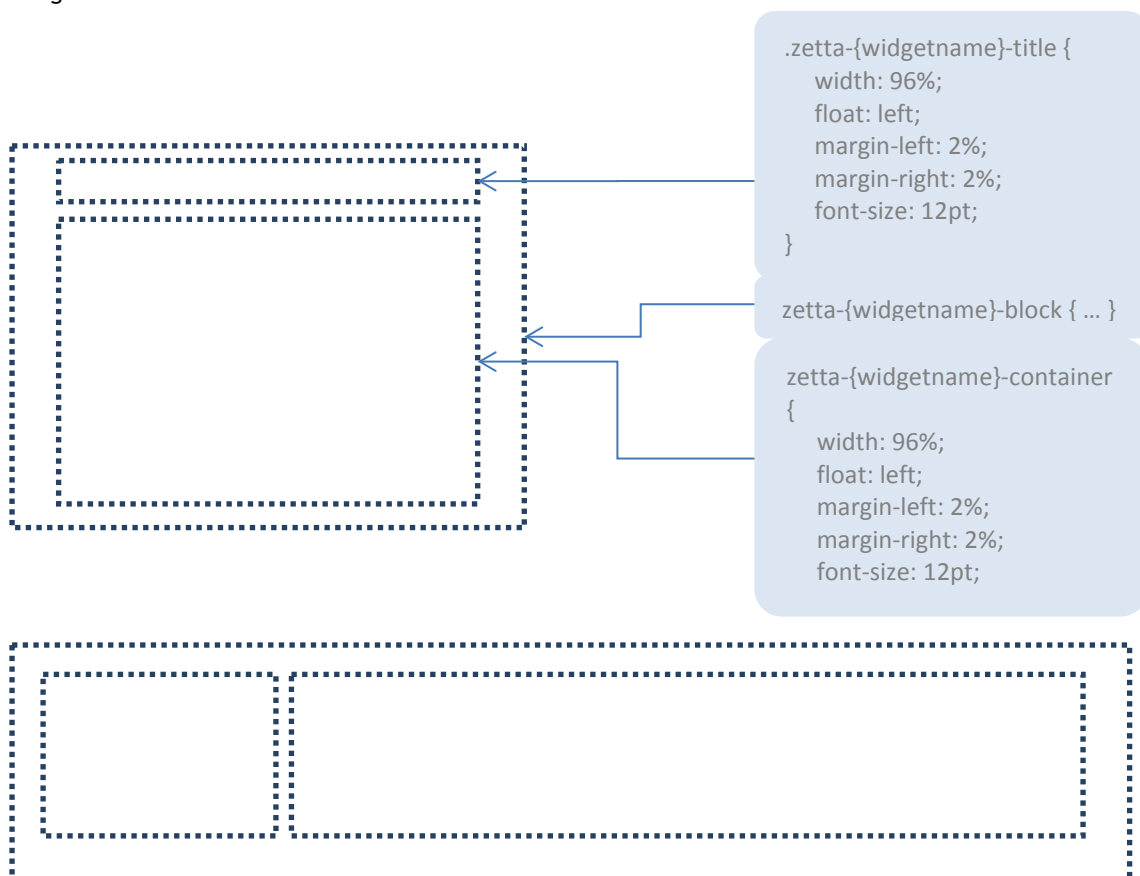
Each of these components can be configured either as a group or individually. It is easy to specify, for instance, that all charts follow a certain layout configuration, but it is also possible to specify that just a single chart in that set deviates a little from the rest of them. In each case, the configuration is accomplished using a CSS file: `app/webapps/ZettaSearch/css/zettasearch.css`. Within this CSS file, all of the charts are configured using sections such as this:

```
.zetta-facetchart-block {
    float: left;
    width: 100%;
    background-color: #ffffff;
    margin-bottom: 10px;
}
```

This CSS block instructs the browser to create a block that fills the entire width of the space allotted to it, to set the background to white, and to space it from the next item below it by 10 pixels. Note the naming convention for the CSS class: `zetta-{widgetname}-{widgetcomponent}`. For individual skinning, the identifier for the widget can be used (“PersonPie” for this example) like this:

```
#PersonPie.zetta-facetchart-block { ... }
```

The skinning of widgets in SBDA Search follows additional general guidelines, though some widgets require specialized layout and configuration properties. In general, however, there is an outer block that contains a title field and a container component. Figure 20 shows examples of how layout works for most functional widgets in SBDA Search.



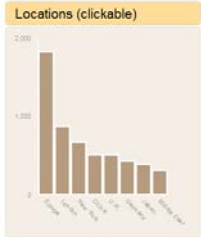
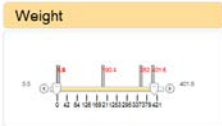


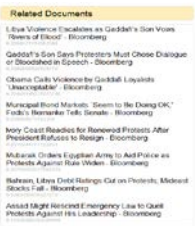
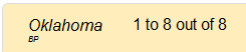
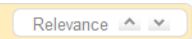
**Figure 20:** General layout and container structure for skinning objects in SBDA Search. The outer container is the “block.” Inside the block are titles and an additional container that holds the chart of rangelslider or other widget device. In the bottom block, the title has been floated “left” but assigned a width of 20% while the container takes up 80% of the space. Additional variations can move the title to the right of the container.

# Taxonomy of Statistica Big Data Analytics Search widgets

Widget	Visual Example	Features	Tag Example	web.xml	CSS Classes/Ids (css/zettsearch.css)
Search Box		<p>Autosuggest</p> <p>Control source index fields and number of results.</p> <p>Frequency filtering.</p> <p>Intelligent case folding</p>	<code>&lt;zetta:searchbox autosuggest="true"/&gt;</code>	<code>autosuggest</code> <code>-lower-frequency</code> <code>autosuggest-fields</code> <code>autosuggest-limit</code>	<code>zetta-searchbox-bar</code> <code>zetta-searchbox-container</code> <code>zetta-searchbox-input</code> <code>zetta-searchbox-button</code> <code>zetta-searchbox-ac-input_popup</code>
Spellcheck		<p>Collection-based</p> <p>Case insensitive</p>	<code>&lt;zetta:spellcheck/&gt;</code>		<code>zetta-spellcheck-block</code> <code>zetta-spellcheck-label</code> <code>zetta-spellcheck-item</code>
Timeline		<p>Attach to metadata dates from document publication or extracted from text.</p>	<code>&lt;zetta:facet timeline title="Date" granularity="100" maxheight="40" ticks="true" id="TheDate" direction="horizontal" group="date"/&gt;</code>		<code>zetta-facet timeline-block</code> <code>zetta-facet timeline-title</code> <code>zetta-facet timeline-resultblock</code> <code>zetta-facet timeline-label</code> <code>zetta-facet timeline-container</code> <code>zetta-facet timeline-slider</code> <code>zetta-facet timeline-horizontal-block</code> <code>zetta-facet timeline-horizontal-title</code> <code>zetta-facet timeline-horizontal-resultblock</code> <code>zetta-facet timeline-horizontal-label</code> <code>zetta-facet timeline-horizontal-container</code> <code>zetta-facet timeline-horizontal-slider</code> <code>zetta-facet timeline-horizontal-bar</code> <code>zetta-facet timeline-horizontal-barhover</code> <code>zetta-facet timeline-horizontal-rule</code> <code>zetta-facet timeline-horizontal-barrule</code> <code>zetta-facet timeline-horizontal-tick</code>
Suggested Queries		<p>Configure the source phrase fields in index</p> <p>Lexical and statistical matches across</p>	<code>&lt;zetta:suggestions id="Suggestions" title="Suggested Queries" limit="5"/&gt;</code>		<code>zetta-suggestions-block</code> <code>zetta-suggestions-title</code> <code>zetta-suggestions-container</code> <code>zetta-suggestions-result</code>



Widget	Visual Example	Features	Tag Example	web.xml	CSS Classes/Ids (css/zettsearch.css)
		collection.	>		
Breadcrumb Tracking		<p>Query tracking</p> <p>Optional counts</p> <p>Subquery tracking</p> <p>Only show queries different from the last</p>	<pre>&lt;zetta:breadcrumbs label="Previous Searches"/&gt;</pre>	<pre>max-breadcrumbs</pre>	<pre>zetta-breadcrumbs-bar zetta-breadcrumbs-label zetta-breadcrumbs-linkbar zetta-breadcrumbs-link zetta-breadcrumbs-queryblock zetta-breadcrumbs-query zetta-breadcrumbs-facetquery zetta-breadcrumbs-count</pre>
Google Map		<p>Location resolution via Google</p> <p>Click-through to drill-down</p>	<pre>&lt;zetta:googlemap draggable="true" id="MyMap" zoom="0" title="Locations" group="location"/&gt;</pre>		<pre>zetta-googlemap-map zetta-googlemap-block zetta-googlemap-title</pre>
Charts		<p>Bar, column, pie, and line charts.</p> <p>Wire to different metadata fields.</p> <p>Configure colors based on themes (orange, red, purple, gradients, gray, green, dark, mix, bluegreen)</p>	<pre>&lt;zetta:facetchart clickable="true" id="LocationBar" limit="8" theme="orange" type="column" title="Locations (clickable)" group="location"/&gt;</pre>		<pre>zetta-facetchart-block zetta-facetchart-title zetta-facetchart-legend zetta-facetchart-container zetta-facetchart-chart</pre>
Range Slider		<p>Tied to numerical metadata.</p> <p>Translates integer and floating point data.</p> <p>Filter results when thumbs released.</p>	<pre>&lt;zetta:facetrangeslider title="Weight" granularity="100" maxheight="40" ticks="true" id="Weight1" direction="horizontal" group="weight"/&gt;</pre>		<pre>zetta-facetrangeslider-horizontal-block zetta-facetrangeslider-horizontal-title zetta-facetrangeslider-horizontal-resultblock zetta-facetrangeslider-horizontal-label zetta-facetrangeslider-horizontal-container zetta-facetrangeslider-horizontal-slider zetta-facetrangeslider-horizontal-bar zetta-facetrangeslider-horizontal-rule zetta-facetrangeslider-</pre>

Widget	Visual Example	Features	Tag Example	web.xml	CSS Classes/Ids (css/zettsearch.css)
					horizontal-barrule zetta-facetrangeslider-horizontal-tick zetta-facetrangeslider-vertical-block zetta-facetrangeslider-vertical-title zetta-facetrangeslider-vertical-resultblock zetta-facetrangeslider-vertical-label zetta-facetrangeslider-vertical-container zetta-facetrangeslider-vertical-slider zetta-facetrangeslider-vertical-bar zetta-facetrangeslider-vertical-rule zetta-facetrangeslider-vertical-barrule zetta-facetrangeslider-vertical-tick
More Like This		Collection-based  Ranked results and scoring based on document content.  Configurable fields for comparisons.	<zetta:morelikethis count="8" title="Related Documents" field="title,text".>		.zetta-morelikethis-block .zetta-morelikethis-title .zetta-morelikethis-result .zetta-morelikethis-resultsblock .zetta-morelikethis-resulttitle .zetta-morelikethis-resultscore
Counts		Configurable.  Shows subqueries.	<zetta:counts/>		zetta-counts-bar zetta-counts-queryblock zetta-counts-query zetta-counts-outof zetta-counts-facetquery zetta-counts zetta-counts-advance zetta-counts-button zetta-counts-next zetta-counts-prev
Sort		Sort by score ascending or descending.	<zetta:sort label="Sort"> <zetta:searchresultsorter label="Relevance" field="score"/> </zetta:so		zetta-sort-bar zetta-sort-label zetta-searchresultssorter zetta-searchresultssorter-up zetta-searchresultssorter-down zetta-searchresultssorter-label zetta-searchresultssorter-group

Widget	Visual Example	Features	Tag Example	web.xml	CSS Classes/Ids (css/zettsearch.css)
			rt>		
Snippet	the <b>Hong Kong</b> developer	Highlight matching terms in context	<zetta:snippet />	highlight- pretag highlight- posttag	zetta-snippets-block zetta-snippets-snippet zetta-snippets-highlight
Document image		2004 Embedded image from external source with resize features	<zetta:documentidimage path="/Zett talmageSer ver/image s" var="q" w="500" h="300"/>		zetta-documentidimage-block
Search Results Facets		Per-document results facets embedded in the search results	<zetta:searchresultfa cets title="Loca tions" limit="4" group="loc ation"/>		zetta-searchresultfacets-block zetta-searchresultfacets-title zetta-searchresultfacets-item
Search Results Title	Barry Bonds's Hat Size Increased, I	Connect search results with display page.	<zetta:searchresultti tle redirect="d isplay.jsp? docno=""/>		zetta-searchresulttitle-block zetta-searchresulttitle-link
Search Results Facet Image		Display images related to facets extracted from the search result.	<zetta:searchresultfa cetimage limit="1" group="per son" path="/Zet talmageSer ver/image s" var="q" w="70" h="80"/>		zetta-searchresultfacetimage- block zetta-documentidimage-block
Search Results Link Image		Connect specific result to external resource via URL.  Match URL prefixes to specific images.	<zetta:searchresultur llinkimage prefix="q4 d" image="im ages/q4dli nksmall.pn g"/>		zetta-searchresulturllinkimage- block zetta-searchresulturllinkimage
Document Display		Display indexed document using custom	<zetta:displaydoc />		zetta-display-document (css/docsheet.css)
Display Document Identifier	id=alero-restarts-cat-cracker-ah-its-andmore-ohlahoma-refinery.html	Shows the unique identifier for the document	<zetta:displaydoc />		zetta-display-id

**Table G:** The complete range of Statistica Analytics Search capabilities.

# Hello Search! with Statistica Analytics Search

As a starting point for understanding how to create your own search experience, let us walk through the steps to creating a search experience. This section assumes that you have installed the Statistica Analytics Search system and have verified that it is up and running.

## Creating a New Search Page

Linux and Mac OS X (**NOTE:** substitute an editor of your choice)

```
% emacs -nw  
SBDASEARCH_HOME/app/webapps/ZettaSearch/mysearch.jsp
```

Enter the following into mysearch.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
  
<%@ taglib tagdir="/WEB-INF/tags/zetta" prefix="zetta" %>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
  
<html>  
<head>  
    <title>Hello Search!</title>  
    <zetta:init root="mysearch.jsp"/>  
</head>  
<body>  
    <zetta:search>  
        <zetta:resolvesearch></zetta:resolvesearch>  
  
        <zetta:searchbox/>  
        <zetta:searchresultsblock>  
            <zetta:counts></zetta:counts>  
            <zetta:searchresult>  
                <zetta:snippet/>  
            </zetta:searchresult>  
        </zetta:searchresultsblock>  
    </zetta:search>  
</body>
```

And with that, you have a basic search engine that displays a ranked list of results and counts in response to a query. So let us take a look at what we did. First, there is some boilerplate related to the JSP system:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
  
<%@ taglib tagdir="/WEB-INF/tags/zetta" prefix="zetta" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

This code informs the application server how to render the resulting HTML page. There is also a reference to the Statistica Analytics Search system and where to find the custom tag library that defines the user interface for the system.

Next, we initialize the SBDA Search system within the HTML head tag and tell it to use mysearch.jsp as its “root” page for resolving searches. In the body of the page, we define a search system, asking SBDA Search to resolve any searches, create a simple search box, and follow it with a search results block. Within the search results block, we ask for the counts of our search results to be rendered, and then a snippet for each result to be rendered as well.

All of the styles for mysearch.jsp are provided for by the standard zettasearch.css in `SBDASEARCH_HOME/app/webapps/ZettaSearch/css`. Overriding those styles can easily be accomplished by creating a new style sheet:

```
% emacs -nw SBDASEARCH_HOME/app/webapps/ZettaSearch /css/mysearch.css
```

and inserting a reference to that style sheet after the `<zetta:init.../>` tag in the `<head>...</head>` block of mysearch.jsp:

```
<link href="css/mysearch.css" rel="stylesheet" type="text/css">
```

Alternatively, the `<zetta:init .../>` tag itself takes an additional attribute, `css` that can be used to specify a complete override of the default `zettasearch.css` style sheet.

## Using Statistica Big Data Analytics Search

In this section, we will review the steps to creating an SBDA Search user experience. In order to use this tutorial, you need to have an operating SBDA Search installation. Install SBDA Search or contact your information technology representative prior to using this section of the guide.

The first step to using SBDA Search is to create a search index. This can be done in several different ways:

- Use Solr tools to index sample documents into the included version of Solr.
- Use Statistica Analytic Suite to crawl, extract, and index large document collections using the distributed Hadoop open-source engine as its core.

## Creating Solr indexes

Solr search indexes are created by indexing documents that are expressed in a special XML language. The XML tags enclose different aspects of the documents to be indexed, supporting the indexing of rich and complex metadata associated with documents and document collections. Each field that is specified for a document must, in turn, be referenced by the schema of the Solr installation. For instance, if you intend to index a field that is numeric such as the weight of a product, you must ensure that the Solr schema specifies that the field is of a numeric type.

Here is an example of an XML formatted product document that is provided with the Solr system:

```
<add>
  <doc>
    <field name="id">3007WFP</field>
    <field name="name">Dell Widescreen UltraSharp
3007WFP</field>
    <field name="manu">Dell, Inc.</field>
    <field name="cat">electronics</field>
    <field name="cat">monitor</field>
    <field name="features">30" TFT active matrix LCD,
2560 x 1600, .25mm dot pitch, 700:1 contrast</field>
```

```

        <field name="includes">USB cable</field>
        <field name="weight">401.6</field>
        <field name="price">2199</field>
        <field name="popularity">6</field>
        <field name="inStock">true</field>
    </doc>
</add>

```

This sample shows the basic layout for adding a document to Solr. The encasing tags `<add>...</add>` instruct the Solr engine to add this document to the index. The field with attribute “ID” must be unique for the standard Solr schema installation and serves to identify the document for update and deletion purposes. Each of the remaining fields is metadata or data about this document and corresponds to an entry in the Solr schema specification at

**SBDASEARCH\_HOME/app/solr/conf/schema.xml**

and the corresponding version on Windows.

For instance, the “price” field has the following entry in the Solr schema:

```
<field name="price" type="float" indexed="true" stored="true"/>
```

This specifies that the price field will be treated as a floating point number and indexed/stored in the index.

Submitting documents to Solr for indexing can be accomplished using the provided tools in

**SBDASEARCH\_HOME/bin**

Two tools are provided, `post.sh` for Linux and Mac OS X, and `post.jar` that can be used on any platform.

To use `post.jar`, issue the following command from the bin directory:

```
% java -Ddata-files -Durl=http://[hostname]:9100/solr/update -jar post.jar file1.xml ... fileN.xml
```

Note that this will update existing files as well.

## Using Statistica Big Data Analytics

The Statistica Big Data Analytics (SBDA) platform provides a drag-and-drop authoring environment for creating rich analytics and search solutions. SBDA Search is also packaged with SBDA. SBDA also contains scalable named entity extractors that can be used to extract people, places, organizations, biological information, and other features from text documents.

To use SBDA to create indexes, use the existing tools such as XML English Extractor and XML Biological Extractor to create a workflow that produces Solr-ready documents for indexing. The Solr Search Engine component can then submit those documents to a Solr instance for indexing.

Note that the Solr instance should have relevant schema definitions that can support the rich metadata created by the indexing process. The following schema modifications are required to use XML Biological Extractor:

```

<field name="PROTEIN" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="DNA" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="RNA" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="CELL_TYPE" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="CELL_LINE" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="doctext" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="plaintext" type="text" indexed="true" stored="false" multiValued="true"/>

<copyField source="doctext" dest="text"/>
<copyField source="PROTEIN" dest="text"/>

```

```

<copyField source="DNA" dest="text"/>
<copyField source="CELL_TYPE" dest="text"/>
<copyField source="CELL_LINE" dest="text"/>
<copyField source="RNA" dest="text"/>

```

Similarly, in order to use the English extractor and document indexing capability in SBDA, the following schema changes are mandatory:

```

<field name="title" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true"/>
<field name="subject" type="textgen" indexed="true" stored="true"/>
<field name="location" type="string" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" termVectors="true"/>
<field name="fileid" type="textgen" indexed="true" stored="true"/>
<field name="head" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" compressed="true" termVectors="true"/>
<field name="first" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" compressed="true" termVectors="true"/>
<field name="note" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" termVectors="true"/>
<field name="doc" type="textgen" indexed="true" termPositions="true" termOffsets="true"
stored="true" termVectors="true" compressed="true" multiValued="true"/>
<field name="second" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" termVectors="true"/>
<field name="dateline" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" termVectors="true"/>
<field name="byline" type="textgen" indexed="true" stored="true" multiValued="true"
termPositions="true" termOffsets="true" termVectors="true"/>
<field name="description" type="textgen" indexed="true" compressed="true" stored="true"/>
<field name="comments" type="textgen" indexed="true" compressed="true" stored="true"/>
<field name="author" type="textgen" indexed="true" stored="true"/>

<field name="type" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="site" type="text" indexed="true" stored="true" multiValued="true"/>
<field name="keywords" type="textgen" indexed="true" stored="true"/>
<field name="category" type="textgen" indexed="true" stored="true"/>
<field name="content_type" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="last_modified" type="date" indexed="true" stored="true"/>
<field name="links" type="string" indexed="true" stored="true" multiValued="true"/>

<field name="organization" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="person" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="jobtitle" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="date" type="date" indexed="true" stored="true" multiValued="true"/>
<field name="money" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="url" type="string" indexed="true" stored="true" multiValued="true"/>

```

```
<field name="overview" type="textgen" indexed="true" stored="true" multiValued="true"/>
```

```
<field name="autosuggest" type="textgen" indexed="true" stored="true" multiValued="true"/>
```

An example of the supporting schema is available in the Statistica Big Data Analytics installation package.

## Third party software

SBDA Search contains third-party software from the Apache Foundation and other sources. The following licenses apply to the included Solr, Dojo, and Jetty components:

### Solr

Copyright 2011 Statistica, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

### Dojo

Copyright (c) 2005-2011, The Dojo Foundation

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of the Dojo Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Jetty

Copyright 2011 Statistica, Inc.



Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

# Working with Q4D Builder

Statistica Big Data Analytics integrates with Q4D's Builder product to support exporting Big Data and interacting with that data in a comprehensive three-dimensional information visualization and modeling tool.

Initial support for Q4D is based on exporting quantitative relationship information derived from Big Data resources to Q4D using an intermediary MySQL database. Future support will expand the interaction styles to increase the types of data and the supported visualizations.

In this guide, we lead you through the steps required for setting-up Q4D support in Statistica as well as example use cases that combine harvested data with Q4D "Metabase" resources.

## Setup and configuration

In order to use Q4D Builder as described in this document, you need to install the system on a local desktop. For this demo, you must also be connected to the Q4D Metabase. You also need to install MySQL so that both the Q4D application and the Statistica application server can communicate with MySQL. Typically, this is over port 3306.

In addition, a new database schema needs to be imported into the MySQL instance. Typically, the schema defines a new database called "q4d", but that can be changed if needed. The schema creation script is included in this document in section 0Q4D Schema Script.

Once the database is created, the final step is to configure Q4D Builder to connect to the database. To do this, select **Application -> Add Connection** and enter the relevant information. After confirming that you can then open a connection to the database (it will appear in the right list of connection types in the Builder window), exit the Builder application.

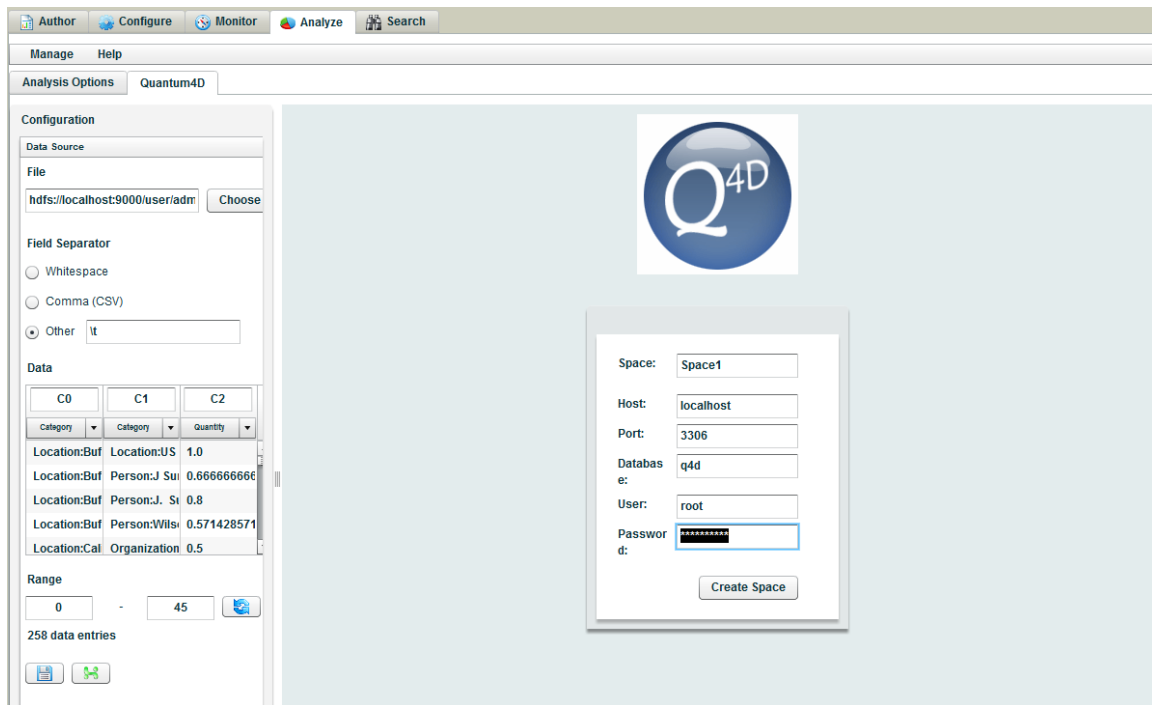
## Creating your Q4D space

Creating a Space in Q4D (Q4D) Builder using Statistica is straightforward. The same techniques that guide other visualizations in Statistica apply to Q4D as well. These techniques involve binding to a Hadoop Distributed File System (HDFS) file, selecting the column-divider character or characters, and deciding what represents the data items and the linking magnitudes.

Figure 21 shows an example of creating a new Q4D Space. The steps required include:

1. Choose an HDFS data source.
2. Specify the separators within the data.
3. Configure at least three columns in the data to represent two categories and one magnitude of the relationships between the categories.
4. Specify the parameters of the MySQL database that will bind the data.
5. Click **Create Space** to export the HDFS data to MySQL and invoke the local Q4D client to bind to the new visualization.

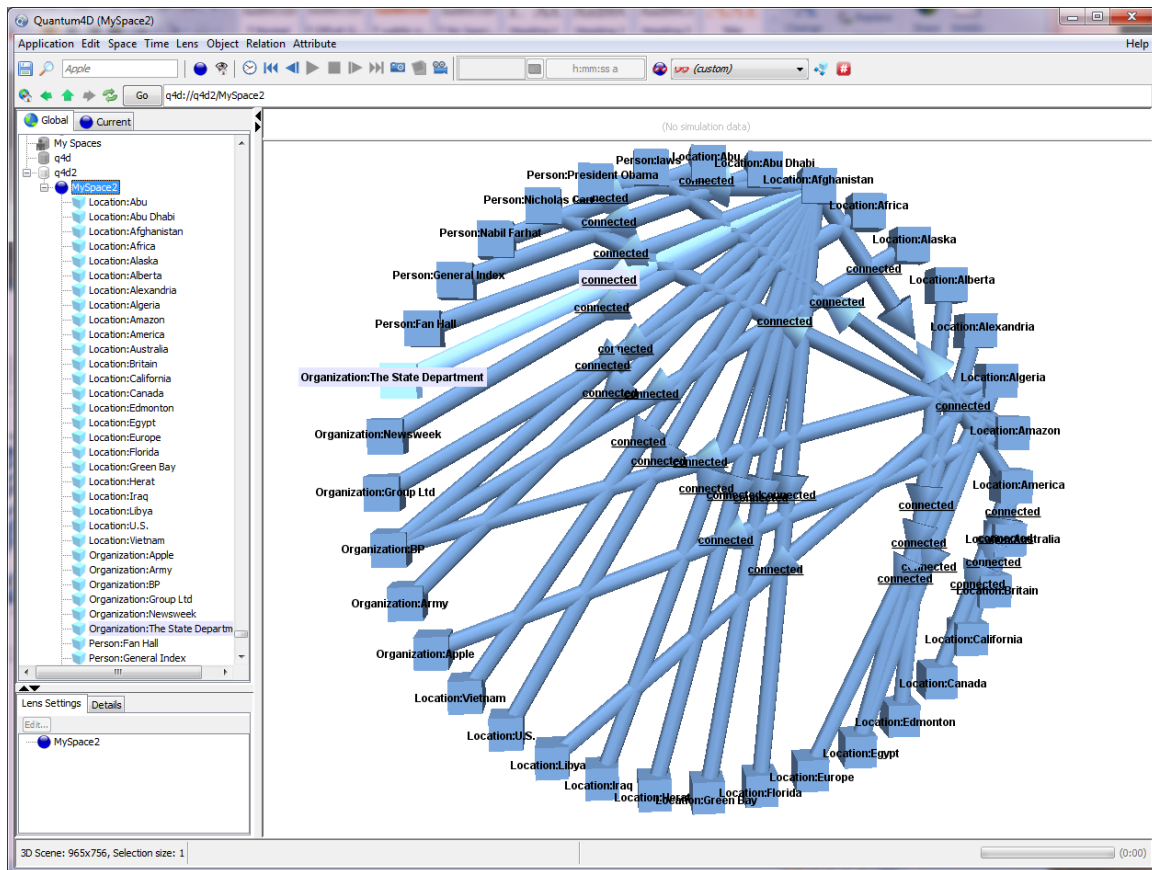
**NOTE:** Q4D Builder must not be currently running during this process in order to ensure efficient binding of Q4D to the data source.



**Figure 21:** Creating a Q4D Space involves binding Q4D to filtered data from Hadoop Distributed File System (HDFS) sources

## Linking spaces in Q4D

In this example, we will use the generated MySpace data to connect to other resources from the Q4D Metabase. To get started, ensure that you have generated a sample space and have it loaded into Q4D (Figure 22). For this task, we will identify important information within the relationship graph and connect that information to other resources. In other words, we will “mash up” the space with existing data.



**Figure 22:** Space generated from Social Network Analysis showing forty or so people, places, and organizations.

To get started, look over the different entities and choose an entity such as “Organization: Apple.” Open the Metabase view (if it is not already open) and search for “Apple” using the search block at the top. Figure 23Figure 22 shows the search results for “Apple” across the entire Metabase. These results include the computer company, a hospitality management company, and a variety of relationships and objects that include “Apple” in their description or other related data. Our goal is to embed the Metabase Space for AAPL inside the Organization: Apple node of our graph to support clicking through to the embedded space and zooming in on the ticker performance of the company. To do this, open the space for Apple (AAPL) Market Focus as shown in Figure 22Figure 24, right-click, and select **Copy** for the space. Next, select the Organization: Apple node either in the graph view or in the right column of the graph in Figure 21. Choose **Paste Special -> Clone** to add the Q4D Space as a child of the Organization: Apple node. There is no need to add child spaces during the add process (dialog box).

You have just linked the two spaces. You can now double-click on the Organization: Apple, select the embedded space and fly into the market data for the space. Click the back arrow at the top to return to your generated space.

As an additional exercise, try linking the Organization: BP to some of the BP-related searchable objects in the Metabase, including the spot oil prices, crude reserves, and other aspects of the BP oil industry. Objects and relationships can also be added, but require additional considerations regarding how they connect to other objects within your visualization.



Figure 23: Results from searching on the keyword “Apple”

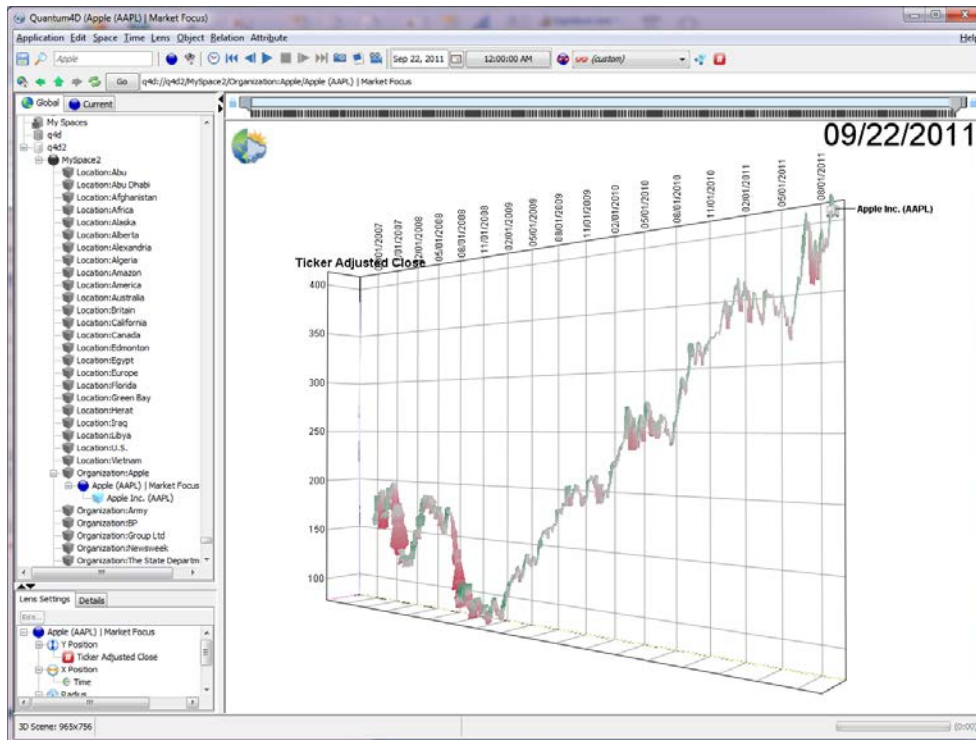


Figure 24: Fly-through from the Organization: Apple object into the Apple AAPL Market data

You can also create new Spaces, Objects, and Relationships within your existing Space, and within Objects as well, populating the new Spaces with other Objects and Relationships. For instance, select Location: Egypt and add a new Space named Government (Figure 25). When the Space is created, your view will fly into the Space. You can then create new Objects and Relationships within that Space that can be discovered by others who enter the Space.

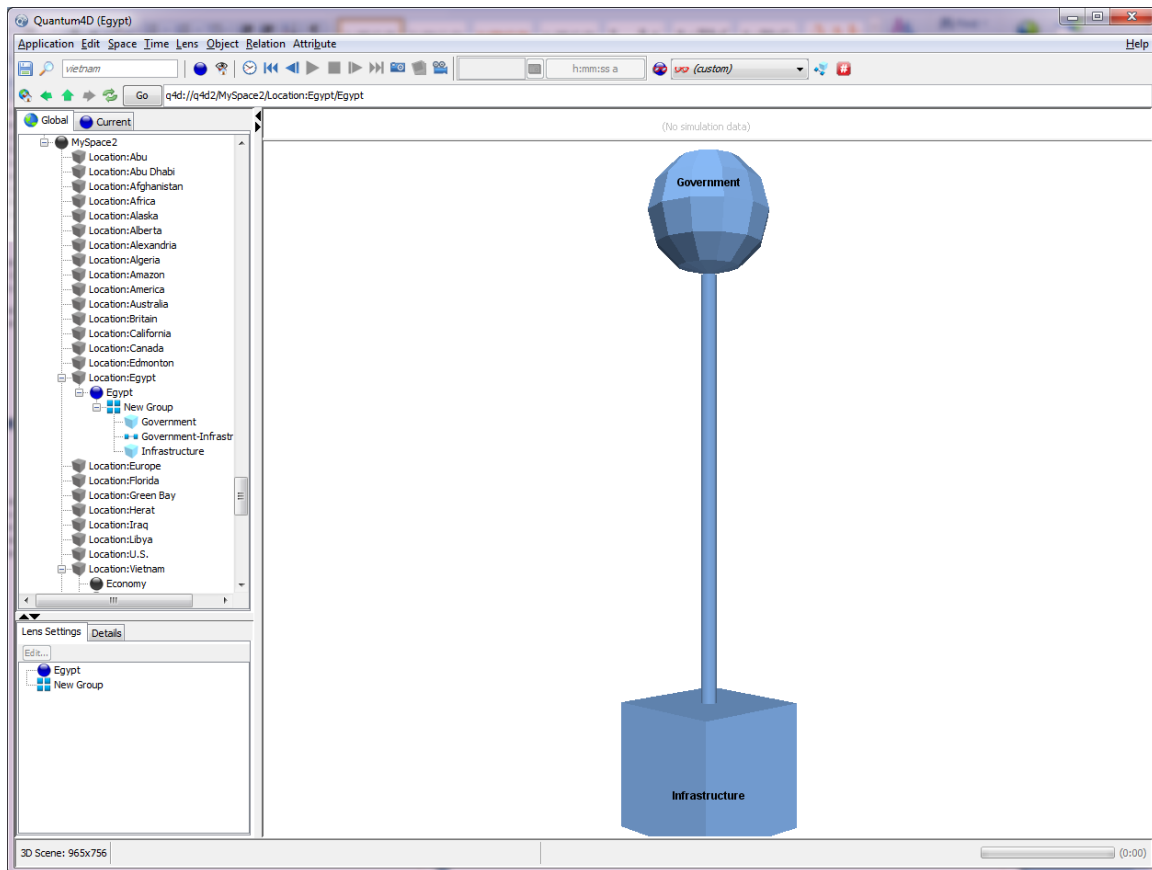


Figure 25: A manually created Space that is embedded in the Location: Egypt node of the visualization

## Q4D schema script

```
CREATE DATABASE q4d;
```

```
USE q4d;
```

```
CREATE TABLE attributedefinition
```

```
(
  ATTRDEF_ID varchar(255) NOT NULL,
  ISSYSTEMATTRIBUTE SMALLINT NOT NULL,
  DEFAULT_VALUE TEXT,
  VALUE_TYPE varchar(255),
  METADATA TEXT,
  DISPLAYNAME varchar(255)
);
```

```
CREATE TABLE entity
```

```
(
  ENTITY_ID varchar(255) NOT NULL,
```

```

METADATA TEXT,
EPO varchar(255),
EP1 varchar(255),
DIRECTION varchar(31),
ETYPE varchar(255) NOT NULL
);

```

```

CREATE TABLE entitytype
(
ETYPE_ID varchar(255) NOT NULL,
ETYPE varchar(255) NOT NULL,
METADATA TEXT
);

```

```

CREATE TABLE entity_attributeinstance
(
ENTITY_ID varchar(255) NOT NULL,
ATTRDEF_ID varchar(255) NOT NULL,
AI_VALUE TEXT,
USE_AD_DEFAULTVALUE SMALLINT NOT NULL,
METADATA TEXT,
AI_VALUE_STR varchar(1023)
);

```

```

CREATE TABLE ontology
(
ONTOLOGY_ID varchar(255) NOT NULL,
METADATA TEXT
);

```

```

CREATE TABLE repository
(
REPOSITORY_ID varchar(255) NOT NULL,
METADATA TEXT
);

```

```

CREATE TABLE qschema
(
SCHEMA_ID varchar(255) NOT NULL,
METADATA TEXT
);

```

```

INSERT INTO `q4d`.`attributedefinition`
(`ATTRDEF_ID`,
`ISSYSTEMATTRIBUTE`,
`DEFAULT_VALUE`,
`VALUE_TYPE`,
`METADATA`,
`DISPLAYNAME`)
VALUES
('gridManualColor', '1', '0.6,0.6,0.6', 'COLOR', NULL, NULL),
('minPositionX2', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('allowNonSpringDisplacement', '1', 'false', 'BOOLEAN', NULL, NULL),
('showRibbon', '1', 'false', 'BOOLEAN', NULL, NULL),
('shapeAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
('normalAxisPosition', '1', '0.0', 'FLOAT', NULL, NULL),
('_resourceModelName', '1', NULL, 'STRING', NULL, NULL),
('textureAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
('inputFormula', '1', NULL, 'FORMULA', NULL, NULL),
('primaryAxisLabel', '1', NULL, 'STRING', NULL, NULL),
('presetInputFormula', '1', 'false', 'BOOLEAN', NULL, NULL),
('layoutGridLimitColumns', '1', 'false', 'BOOLEAN', NULL, NULL),
('normalizationType', '1', 'AUTOMATIC', 'STRING', NULL, NULL),
('flagLabelShowIcon', '1', 'true', 'BOOLEAN', NULL, NULL),
('edgeColorIndex', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('elasticity', '1', '3', 'INTEGER', NULL, NULL),
('visibilityInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
('notForEntry', '1', 'false', 'BOOLEAN', NULL, NULL),
('demarcateTimeUsingKeyframes', '1', 'false', 'BOOLEAN', NULL, NULL),
('texture', '1', NULL, 'STRING', NULL, NULL),
('spinRate', '1', 'default,0.0', 'FLOAT_TIME_SERIES', NULL, NULL),
('log', '1', 'false', 'BOOLEAN', NULL, NULL),
('hexagonDisplayMode', '1', 'GROUP', 'STRING', NULL, NULL),
('secondaryGridValueXZ', '1', '0.0', 'FLOAT', NULL, NULL),
('secondaryGridValueXY', '1', '0.0', 'FLOAT', NULL, NULL),
('labelFont', '1', NULL, 'FONT', NULL, NULL),
('minPositionZ2', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('showYZGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
('_resourceData', '1', NULL, 'BYTE_ARRAY', NULL, NULL),
('subspaceZScale', '1', '1.0', 'FLOAT', NULL, NULL),
('mainLabelBalloonStringLength', '1', '0.0', 'FLOAT', NULL, NULL),
('minimumDistance', '1', '3', 'INTEGER', NULL, NULL),

```



```

('relationStrength', '1', '5', 'INTEGER', NULL, NULL),
('_lifetimeEnd', '1', NULL, 'DATE', NULL, NULL),
('useOverrideColor', '1', 'false', 'BOOLEAN', NULL, NULL),
('showColumn', '1', 'false', 'BOOLEAN', NULL, NULL),
('_description', '1', NULL, 'STRING', NULL, NULL),
('manualColor', '1', '1.0,1.0,1.0', 'COLOR', NULL, NULL),
('lensSourceId', '1', NULL, 'XREF', NULL, NULL),
('relatedRepulsion', '1', '5', 'INTEGER', NULL, NULL),
('viewOrbiterSettings', '1', NULL, 'STRING', NULL, NULL),
('eventURL', '1', "", 'STRING_TIME_SERIES', NULL, NULL),
('springMaxIterations', '1', '100', 'INTEGER', NULL, NULL),
('minPositionY2', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('fadeBackgroundWithTime', '1', 'false', 'BOOLEAN', NULL, NULL),
('maxColor', '1', '0.016,0.502,0.253', 'COLOR', NULL, NULL),
('lineWidth', '1', '1.0', 'FLOAT', NULL, NULL),
('_resourceType', '1', NULL, 'STRING', NULL, NULL),
('axesVisible', '1', 'false', 'BOOLEAN', NULL, NULL),
('primaryLabelOverride', '1', 'false', 'BOOLEAN', NULL, NULL),
('durationInSecondsPerKeyframe', '1', '0.25', 'FLOAT', NULL, NULL),
('showLabel', '1', 'true', 'BOOLEAN', NULL, NULL),
('enableCameraOrbit', '1', 'true', 'BOOLEAN', NULL, NULL),
('secondaryElementLabel', '1', 'default,""', 'STRING_TIME_SERIES', NULL, NULL),
('flagLabelDisplacementIsScaled', '1', 'false', 'BOOLEAN', NULL, NULL),
('axesAndGridsInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
('visible', '1', 'default,true', 'BOOLEAN_TIME_SERIES', NULL, NULL),
('backgroundColor', '1', '1.0,1.0,1.0', 'COLOR', NULL, NULL),
('_resourceImageName', '1', NULL, 'STRING', NULL, NULL),
('axisVisible', '1', 'false', 'BOOLEAN', NULL, NULL),
('showXZSecondaryGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
('defaultColor', '1', '0.459,0.624,0.831', 'COLOR', NULL, NULL),
('useManualInputMin', '1', 'false', 'BOOLEAN', NULL, NULL),
('showXY', '1', 'false', 'BOOLEAN', NULL, NULL),
('lensSourceIsSpace', '1', 'true', 'BOOLEAN', NULL, NULL),
('showXZGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
('showXYGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
('enabled', '1', 'true', 'BOOLEAN', NULL, NULL),
('_resourceThumbnailData', '1', NULL, 'BYTE_ARRAY', NULL, NULL),
('restLength', '1', '3', 'INTEGER', NULL, NULL),
('normalizeToAnimationRange', '1', 'false', 'BOOLEAN', NULL, NULL),
('relationDamping', '1', '5', 'INTEGER', NULL, NULL),
('edgeColor', '1', 'default,"1.0,1.0,1.0"', 'COLOR_TIME_SERIES', NULL, NULL),

```

```

('currentLens', '1', NULL, 'STRING', NULL, NULL),
('edgeLabelFeature', '1', NULL, 'STRING', NULL, NULL),
('outputFormatString', '1', NULL, 'STRING', NULL, NULL),
('showRings', '1', 'false', 'BOOLEAN', NULL, NULL),
('model', '1', 'N:Cube', 'STRING', NULL, NULL),
('showYZSecondaryGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
('nodePlaneYZ', '1', 'false', 'BOOLEAN', NULL, NULL),
('labelFeatureIcn', '1', 'false', 'BOOLEAN', NULL, NULL),
('positionY', '1', 'default,0.0', 'FLOAT_TIME_SERIES', NULL, NULL),
('labelAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
('positionZ', '1', 'default,0.0', 'FLOAT_TIME_SERIES', NULL, NULL),
('overrideColor', '1', '1.0,1.0,1.0', 'COLOR', NULL, NULL),
('showZX', '1', 'false', 'BOOLEAN', NULL, NULL),
('maxPositionY', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('maxPositionZ', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('positionX', '1', 'default,0.0', 'FLOAT_TIME_SERIES', NULL, NULL),
('maxPositionX', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('childRadiusScale', '1', '1.0', 'FLOAT', NULL, NULL),
('springMaxLayoutRadius', '1', '50.0', 'FLOAT', NULL, NULL),
('showYZ', '1', 'false', 'BOOLEAN', NULL, NULL),
('showLabelBackground', '1', 'false', 'BOOLEAN', NULL, NULL),
('_url', '1', NULL, 'STRING', NULL, NULL),
('loopAnimation', '1', 'false', 'BOOLEAN', NULL, NULL),
('secondaryGridDateYZ', '1', '#0', 'DATE', NULL, NULL),
('enableAntialiasing', '1', 'false', 'BOOLEAN', NULL, NULL),
('nodeColorIndex', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
('viewTracksTime', '1', 'false', 'BOOLEAN', NULL, NULL),
('animationLastFrame', '1', NULL, 'DATE', NULL, NULL),
('labelConstantSize', '1', 'true', 'BOOLEAN', NULL, NULL),
('normalizeToIndividualRanges', '1', 'false', 'BOOLEAN', NULL, NULL),
('hideAll', '1', 'false', 'BOOLEAN', NULL, NULL),
('nodeColor', '1', 'default,"1.0,1.0,1.0"', 'COLOR_TIME_SERIES', NULL, NULL),
('showMarks', '1', 'true', 'BOOLEAN', NULL, NULL),
('_lifetimeStart', '1', NULL, 'DATE', NULL, NULL),
('nodePlaneXZ', '1', 'false', 'BOOLEAN', NULL, NULL),
('nodePlaneXY', '1', 'false', 'BOOLEAN', NULL, NULL),
('useManualPositionZ', '1', 'false', 'BOOLEAN', NULL, NULL),
('animationCurrentFrame', '1', NULL, 'DATE', NULL, NULL),
('flagLabelHorizontalDisplacement', '1', '0.0', 'FLOAT', NULL, NULL),
('reportTemplate', '1', NULL, 'STRING', NULL, NULL),
('useManualPositionY', '1', 'false', 'BOOLEAN', NULL, NULL),

```

('useManualPositionX', '1', 'false', 'BOOLEAN', NULL, NULL),  
 ('timeTrailTrailingSegments', '1', '1000', 'INTEGER', NULL, NULL),  
 ('demarcateAxes', '1', 'true', 'BOOLEAN', NULL, NULL),  
 ('autoContrast', '1', 'true', 'BOOLEAN', NULL, NULL),  
 ('showLine', '1', 'false', 'BOOLEAN', NULL, NULL),  
 ('unrelatedRepulsion', '1', '50', 'INTEGER', NULL, NULL),  
 ('axisDemarcationGroupId', '1', NULL, 'STRING', NULL, NULL),  
 ('centerOffsetX', '1', '0.0', 'FLOAT', NULL, NULL),  
 ('centerOffsetY', '1', '0.0', 'FLOAT', NULL, NULL),  
 ('subspaceYScale', '1', '1.0', 'FLOAT', NULL, NULL),  
 ('centerOffsetZ', '1', '0.0', 'FLOAT', NULL, NULL),  
 ('flagLabelBalloonStringLength', '1', '0.5', 'FLOAT', NULL, NULL),  
 ('lensIncludeAttributeTransforms', '1', 'true', 'BOOLEAN', NULL, NULL),  
 ('backgroundImageScaleMode', '1', 'SCALE\_NONE', 'STRING', NULL, NULL),  
 ('radiusScale', '1', '1.0', 'FLOAT', NULL, NULL),  
 ('lensSourceIsRelation', '1', 'false', 'BOOLEAN', NULL, NULL),  
 ('animationFirstFrame', '1', NULL, 'DATE', NULL, NULL),  
 ('labelMainText', '1', '', 'STRING\_TIME\_SERIES', NULL, NULL),  
 ('maxPositionY2', '1', 'default,NaN', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('springTerminationThreshold', '1', '50', 'INTEGER', NULL, NULL),  
 ('subspaceXScale', '1', '1.0', 'FLOAT', NULL, NULL),  
 ('hideNoData', '1', 'false', 'BOOLEAN', NULL, NULL),  
 ('null', '1', 'NaN', 'FLOAT', NULL, NULL),  
 ('nodeRadius', '1', 'default,1.0', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('noteLinkEntity', '1', NULL, 'XREF', NULL, NULL),  
 ('secondaryGridDateXY', '1', '#0', 'DATE', NULL, NULL),  
 ('useManualInputMax', '1', 'false', 'BOOLEAN', NULL, NULL),  
 ('secondaryGridDateXZ', '1', '#0', 'DATE', NULL, NULL),  
 ('drag', '1', '50', 'INTEGER', NULL, NULL),  
 ('outputAttributeId', '1', NULL, 'XREF', NULL, NULL),  
 ('colorAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),  
 ('\_displayName', '1', NULL, 'STRING', NULL, NULL),  
 ('durationInSeconds', '1', '15.0', 'FLOAT', NULL, NULL),  
 ('maxPositionX2', '1', 'default,NaN', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('minPositionY', '1', 'default,NaN', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('minPositionZ', '1', 'default,NaN', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('textureSize', '1', '100', 'INTEGER', NULL, NULL),  
 ('primaryElementLabel', '1', NULL, 'STRING', NULL, NULL),  
 ('minPositionX', '1', 'default,NaN', 'FLOAT\_TIME\_SERIES', NULL, NULL),  
 ('labelFeature', '1', NULL, 'STRING', NULL, NULL),  
 ('mainLabelHorizontalDisplacement', '1', '0.0', 'FLOAT', NULL, NULL),

```
( 'manualInputMax', '1', '0.0', 'FLOAT', NULL, NULL),
( 'openSubspaceXRef', '1', NULL, 'XREF', NULL, NULL),
( 'lensSourceMatchPattern', '1', NULL, 'STRING', NULL, NULL),
( 'showThread', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'edgeRadius', '1', 'default,1.0', 'FLOAT_TIME_SERIES', NULL, NULL),
( 'layoutOnSinglePlane', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'showBorder', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'useParallelProjection', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'normalizeModelSize', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'timetrailAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
( 'dateFormat', '1', NULL, 'STRING', NULL, NULL),
( 'durationIsPerKeyframe', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'manualInputMin', '1', '0.0', 'FLOAT', NULL, NULL),
( '_time', '1', NULL, 'STRING', NULL, NULL),
( 'eventLabel', '1', '', 'STRING_TIME_SERIES', NULL, NULL),
( 'secondaryAxisLabel', '1', NULL, 'STRING', NULL, NULL),
( 'secondaryGridValueYZ', '1', '0.0', 'FLOAT', NULL, NULL),
( 'hideIncompleteData', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'backgroundImage', '1', NULL, 'STRING', NULL, NULL),
( 'shadowAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
( 'showLayoutCircle', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'layoutNormalAxis', '1', 'Z', 'GEOMETRIC_AXIS', NULL, NULL),
( 'edgeStrength', '1', 'default,0.0', 'FLOAT_TIME_SERIES', NULL, NULL),
( 'showFill', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'maxPositionZ2', '1', 'default,NaN', 'FLOAT_TIME_SERIES', NULL, NULL),
( 'showXYSecondaryGrid', '1', 'false', 'BOOLEAN', NULL, NULL),
( '_longName', '1', NULL, 'STRING', NULL, NULL),
( 'midColor', '1', '0.8,0.8,0.8', 'COLOR', NULL, NULL),
( 'gridPositionYZ', '1', '-1.0', 'FLOAT', NULL, NULL),
( 'showDateLabel', '1', 'true', 'BOOLEAN', NULL, NULL),
( 'labelScale', '1', '1.0', 'FLOAT', NULL, NULL),
( 'layoutGridMaxRanks', '1', '-1', 'INTEGER', NULL, NULL),
( 'layoutAttributesInheritanceIndex', '1', '2', 'INTEGER', NULL, NULL),
( 'showDemarcations', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'gridPositionXZ', '1', '-1.0', 'FLOAT', NULL, NULL),
( 'gridPositionXY', '1', '-1.0', 'FLOAT', NULL, NULL),
( 'layoutSize', '1', '1.0', 'FLOAT', NULL, NULL),
( 'layoutMode', '1', 'CIRCLE', 'LAYOUT_MODE', NULL, NULL),
( 'gridColorMode', '1', 'AUTO_CONTRAST_COLOR', 'GRID_COLOR_MODE', NULL, NULL),
( 'flagLabelShowShortText', '1', 'false', 'BOOLEAN', NULL, NULL),
( 'minColor', '1', '0.863,0.114,0.247', 'COLOR', NULL, NULL);
```



# Appendix A: Processing elements reference

The first section covers the processing introduced in Release 13.1. The second section is a matrix covering the processing elements introduced in Release 2.0 and before.

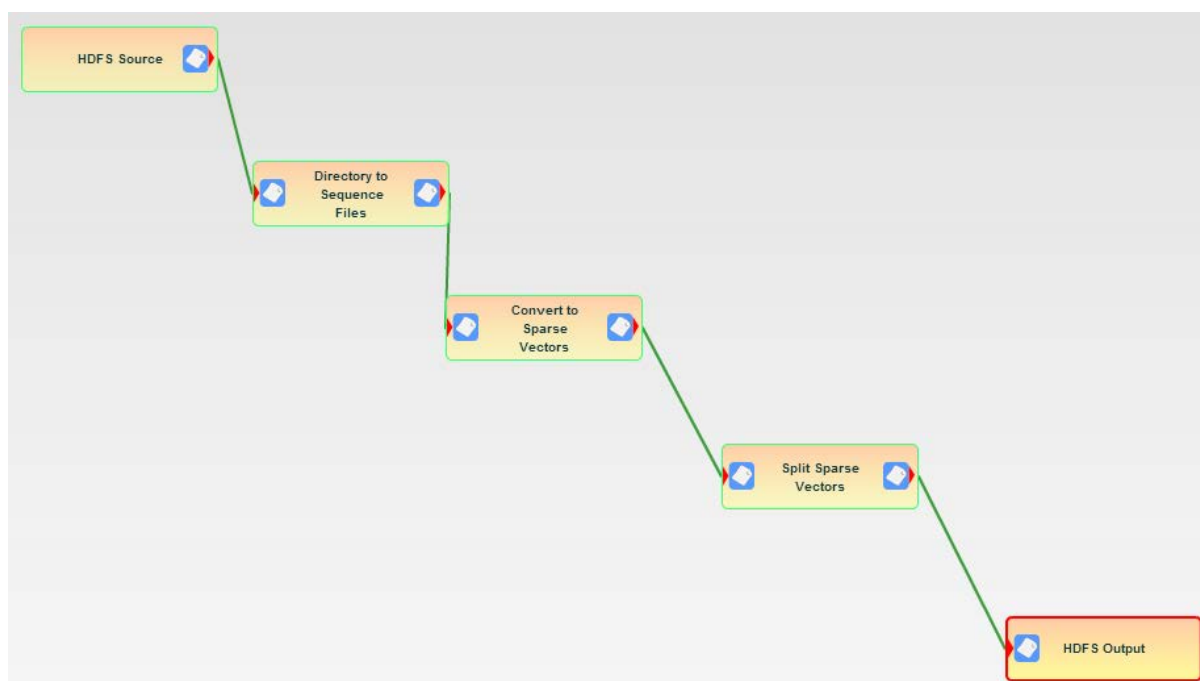
## Release 13.1 processing elements

### Classifiers

#### Convert to Sparse Vectors

Converts sequence files to sparse vectors. The input data needs to be a sequence file.

#### Convert to Sparse Vectors example



#### User Properties

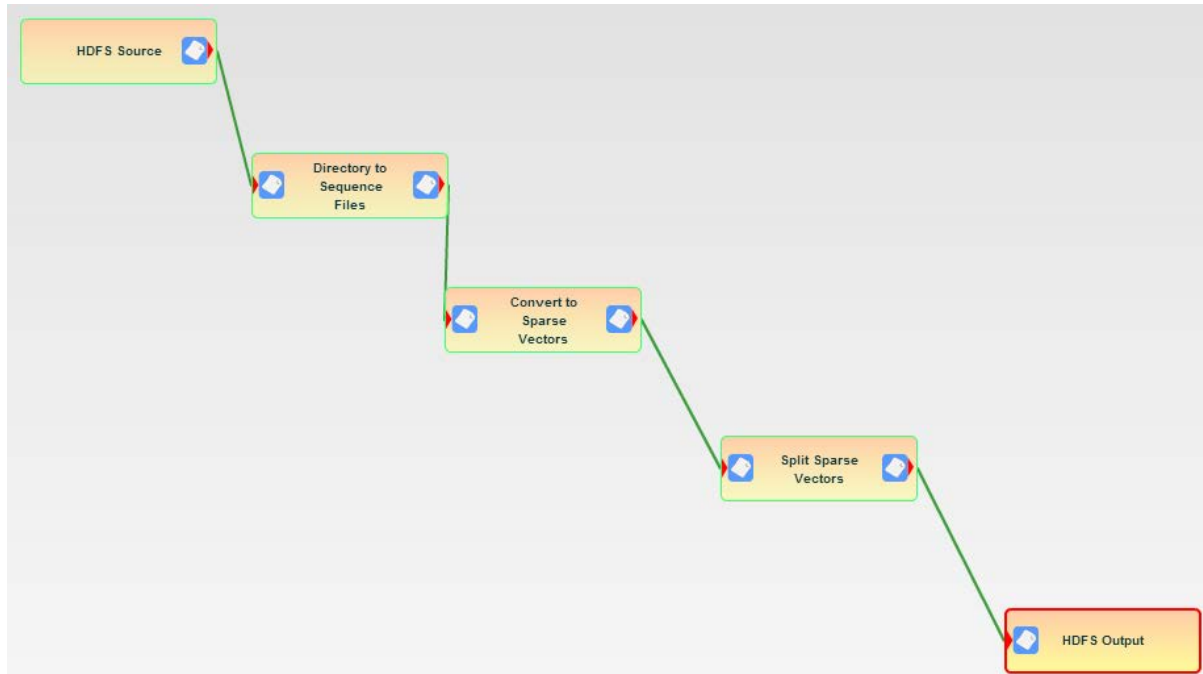
Parameter	Value	Example
minSupport	1	1
analyzerName	Use the class name of the analyzer	
chunkSize	Insert the chunk size in MB. The default is 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TDIDF or TF.	TF

minDF	Insert the minimum document frequency. The default is 1.	1
maxDFPercent	Insert the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	10
maxDFSigma	Insert the portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to be specified is 3.0. In the case where the value is less than 0, no vectors will be filtered. The default is -1.0. This will override the value of maxDFPercent.	3.0
minLLR	Insert with the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Insert the norm to use that is expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE.	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = biGrams, 3 = triGrams, and so on). The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE.	TRUE
namedVector	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	TRUE

## Split Sparse Vectors

Splits the sparse vectors into training and holdout data sets. The input data needs to be a vector file (TF or TFIDF).

### Split Sparse Vectors example



### User Properties

Parameter	Value	Example
testSplitSize	Insert the number of documents held back as test data for each category.	2
testSplitPct	Insert the percent of documents held back as test data for each category.	10
splitLocation	Insert the location for the start of test data expressed as a percentage of the input	50

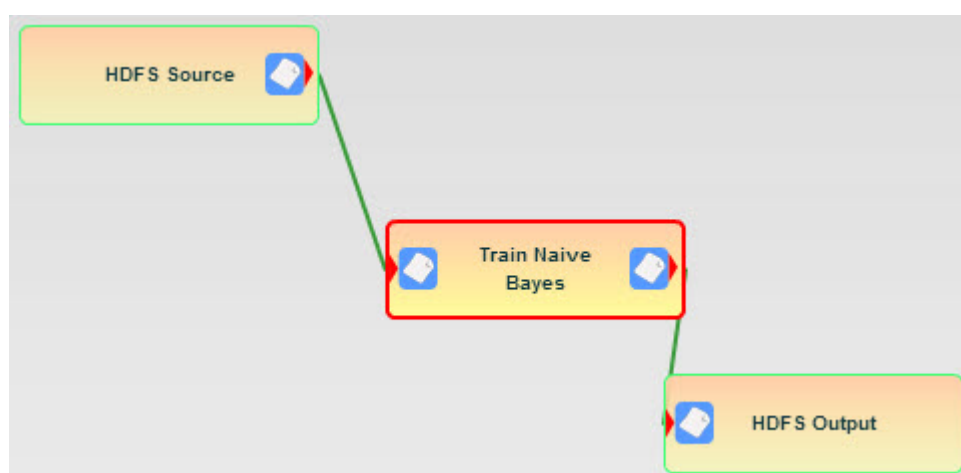
file size (0 = start, 50 = middle, and 100 = end).

randomSelectionSize	Insert the number of items to be randomly selected as test data.	2
randomSelectionPct	10	10
charset	Insert with the name of the character encoding of the input files. This is not needed if using sequence files.	
keepPct	Insert with the percentage of total data to keep in map-reduce mode. The rest will be ignored. The default is 100%.	10
splitMethod	Choose either SEQUENTIAL or MAP-REDUCE.	SEQUENTIAL
mapRedOutputDir	Choose the output directory for map-reduce jobs.	
sequenceFiles	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	TRUE

## Train Naive Bayes

The input data needs to be a vector file (TF or TFIDF) for the Train Naïve Bayes Model.

### Train Naïve Bayes example



### User Properties

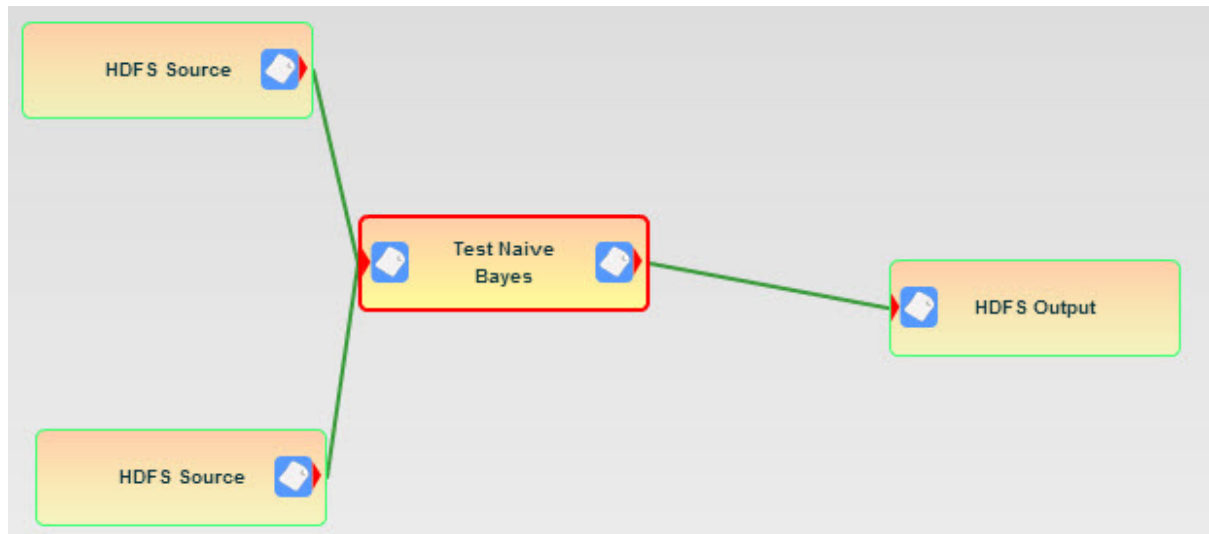
Parameter	Value	Example
labels	Insert a comma-separated list of labels to include in training.	
alpha	Insert a smoothing parameter.	
extractLabels	Choose either TRUE or FALSE.	TRUE
complementary	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	TRUE

## Test Naive Bayes

Runs the Naive Bayes Model on vectors using the existing model. It takes two inputs: the path containing the model and the label index and the path containing the vector file (TF or TFIDF).



## Test Naïve Bayes example



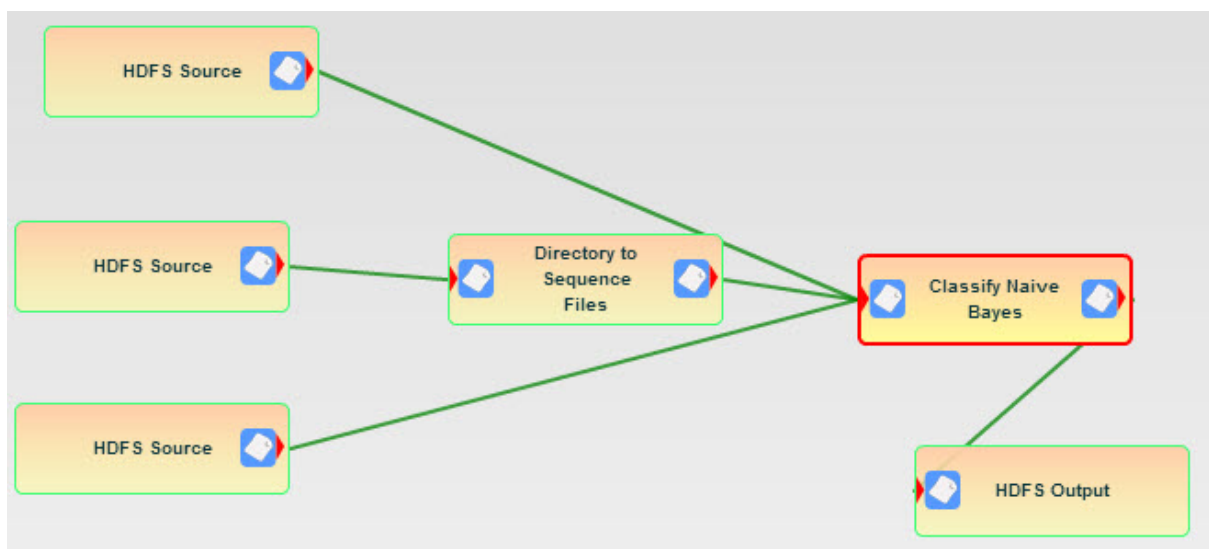
### User Properties

Parameter	Value	Example
complementary	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	TRUE
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Classify Naïve Bayes

Runs the Naïve Bayes Model on input data using the existing model. It takes three input: the path contain the model and the label index, the path containing the dictionary file, and the path that contains the data to classify and that needs to be a sequence file.

### Classify Naïve Bayes example



### User Properties

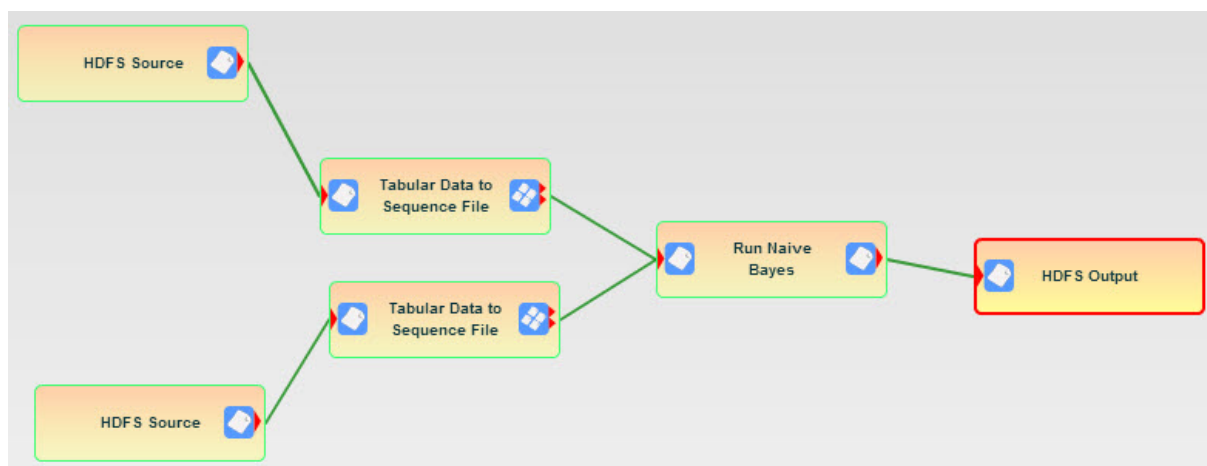
Parameter	Value	Example
minSupport	1	1
analyzerName	Insert the class name of the analyzer.	

chunkSize	Insert the chunk size of MB. The default is 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TRUE or FALSE.	TRUE
minDF	Choose the minimum document frequency. The default is 1.	1
maxDFPercent	Choose the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	10
maxDFSigma	Insert what portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to specify is 3.0. In the case where the value is less than 0, then no vectors will be filtered out. The default is -1.0. This will override the value of maxDFPercent.	3.0
minLLR	Insert the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Replace with the norm to use expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE.	TRUE
maxNGramSize	Insert the maximum size of NGrams to create (2 = BIGrams, 3 = TRIGrams, and so on. The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE.	TRUE
namedVector	Choose either TRUE or FALSE.	TRUE
runSequential	Choose either TRUE or FALSE.	TRUE
complementary	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Run Naive Bayes

Runs the Mahout Naïve Bayes Classifier on training data and input data, both the training data and the data to classify need to be a sequence file.

### Run Naïve Bayes example



### User Properties

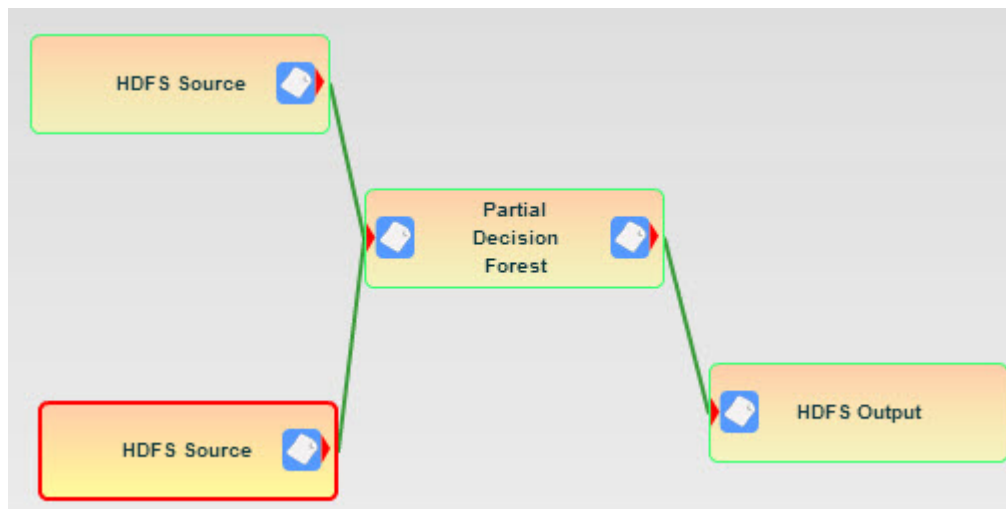
Parameter	Value	Example
minSupport	1	1

analyzerName	Use the class name of the analyzer.	
chunkSize	Choose the chunk size in MB. Defaults to 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TFIDF or TF.	TF
minDF	Insert the minimum document frequency. The default is 1.	1
maxDFPercent	Use the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	99
maxDFSigma	Insert that portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to specify is 3.0. In the case where the value is less than 0, no vectors will be filtered.	3.0
minLLR	Use the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Use the number of reduce tasks. Default value is 1.	1
norm	Insert the norm to be used expressed as either a float or "INF" if you want to use the infinite norm. Must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE.	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = BIGrams, 3 = TRIGrams, and so on. The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE.	TRUE
namedVector	Choose either TRUE or FALSE.	TRUE
runSplit	Choose either TRUE or FALSE.	TRUE
testSplitSize	Insert the number of documents held back as test data for each category.	2
testSplitPct	Use the percentage of documents held back as test data for each category.	10
splitLocation	Specify the location for the start of test data expressed as a percentage of the input file size (0 = start, 50 = middle, 100 = end).	50
randomSelectionSize	Specify the number of items to be randomly selected as test data.	2
randomSelectionPct	10	10
sequenceFiles	Choose either TRUE or FALSE.	TRUE
keepPct	Specify the percentage of total data to keep in Mapreduce mode. The rest will be ignored. The default is 100%.	100
splitMethod	Choose either SEQUENTIAL or MAPREDUCE.	SEQUENTIAL
mapRedOutputDir	Choose the output directory for Map Reduce jobs.	
labels	Insert a comma-separated list of labels to include in training.	
alpha	Choose a smoothing parameter.	
extractLabels	Choose either TRUE or FALSE.	TRUE
complementary	Choose either TRUE or FALSE.	TRUE
overwrite	Choose either TRUE or FALSE.	TRUE
delimiter	Choose the delimiter to delimit the output. For example the tab delimiter is \t. The default is a space character.	/t

## Partial Decision Forest

Runs the Mahout Partial Decision Forest Algorithm

## Partial Decision Forest example



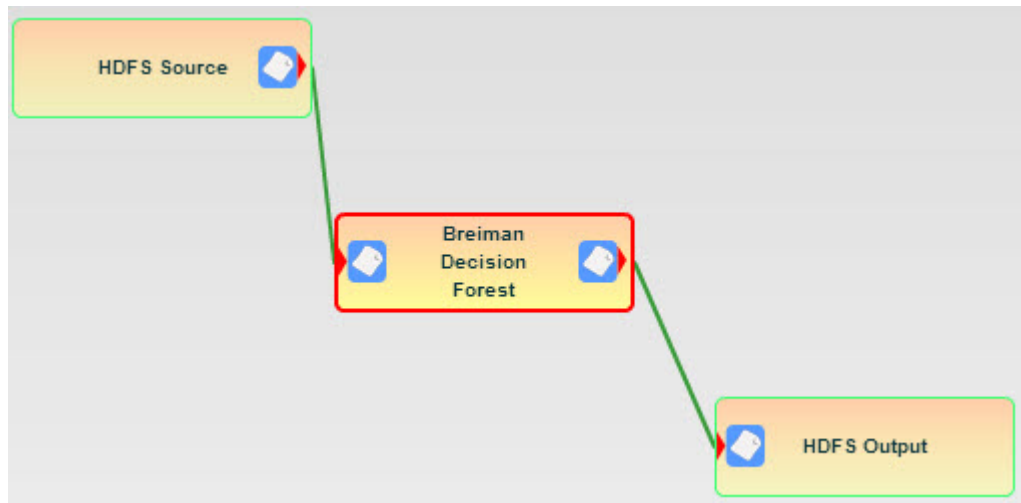
### User Properties

Parameter	Value	Example
descriptor	Specify the data descriptor.	
regression	Choose either TRUE or FALSE.	TRUE
selection	Specify the number of variables to select randomly at each tree-node. For the classification problem, the default is the square root of the number of explanatory variables. For the regression problem, the default is 1/3 of the number of explanatory variables.	5
no-complete	Choose either TRUE or FALSE.	TRUE
minsplit	Replace with a number. If the branching data size is smaller than this value, the tree-node is not divided. The default is 2.	2
minprop	Replace with a number. If the proportion of the variance of branching data is smaller than this value, the tree-node is not divided. In the case of a regression problem, this value is used. The default is 1/1000(0.001).	0.001
seed	Specify the seed value used to initialize the random number generator.	5
partial	Choose either TRUE or FALSE.	TRUE
nbtrees	Choose the number of trees to grow (required).	100
analyze	Choose either TRUE or FALSE.	TRUE
mapreduce	Choose either TRUE or FALSE.	TRUE
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Breiman Decision Forest

Runs the Mahout Breiman Decision Forest Algorithm

## Breiman Decision Forest example



### User Properties

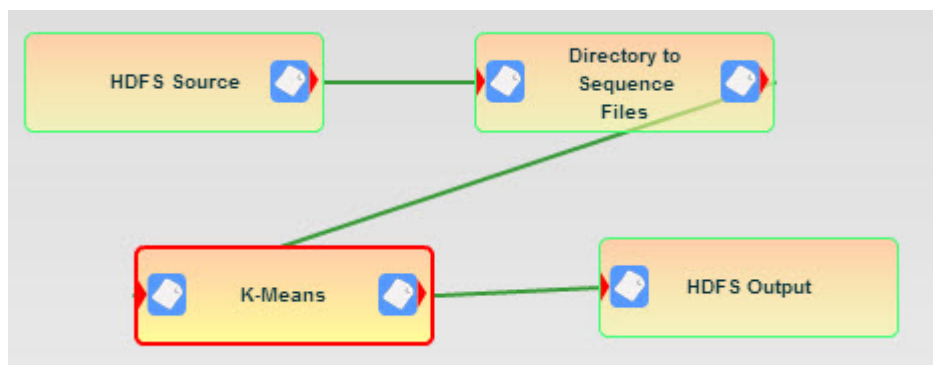
Parameter	Value	Example
descriptor	Insert the data descriptor	
regression	Choose either TRUE or FALSE	TRUE
nbtrees	Insert the number of trees to grow, each iteration (required)	100
iterations	Insert the number of times to repeat the test (required)	2

## Clustering

### K-Means

Runs the Mahout K-Means algorithm. The input data must be a sequence file.

#### K-Means example



### User Properties

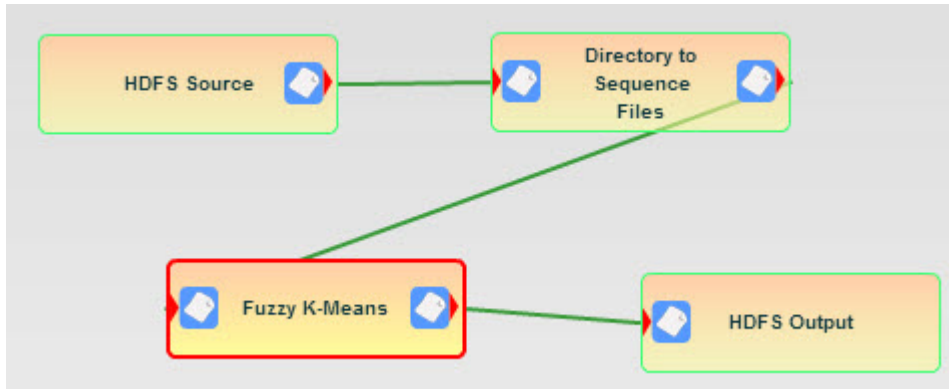
Parameter	Value	Example
minSupport	1	1
analyzerName	Insert the class name of the analyzer	
chunkSize	Choose the chunk size in MG. Defaults to 64MB. The value must be between 100MB and 10000MB.	64
weight	Choose either TDIF or TF	TF
minDF	Choose the minimum document frequency. The default is 1.	1

maxDFPercent	Insert the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	99
maxDFSigma	Insert the portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to be specified is 3.0. In the case where the value is less than 0, no vectors will be filtered. The default is -1.0. This will override the value of maxDFPercent.	1.0
minLLR	Insert with the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Insert the norm to use that is expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = biGrams, 3 = triGrams, and so on). The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE	TRUE
namedVector	Choose either TRUE or FALSE	TRUE
distanceMeasure	Choose ChebyshevDistanceMeasure, CosineDistanceMeasure, EuclideanDistanceMeasure, MahalanobisDistanceMeasure, ManhattanDistanceMeasure, MinkowskiDistanceMeasure, SquaredEuclideanDistanceMeasure, TanimotoDistanceMeasure, WeightedEuclideanDistanceMeasure, or WeightedManhattanDistanceMeasure	ChebyshevDistanceMeasure
numClusters	Choose the number of clusters to create.	2
kmeansClusters	Select the input centroids as vectors. This must be a sequence file of writeable cluster/canopy. If K is also specified, then a random set of vectors will be selected and written out to this path first.	
convergenceDelta	Insert the convergence delta value. The default is 0.5.	0.5
maxIter	Insert the maximum number of iterations (required).	3
clustering	Choose either TRUE or FALSE	TRUE
outlierThreshold	Specify the outlier threshold value.	
overwrite	Choose either TRUE or FALSE	TRUE
clusterDumper.outputFormat	Choose the optional output format for writing out the results. The options are: text, csv, or graph_ml.	csv
clusterDumper.substring	Specify the number of characters of the asformatstring() to print (required).	
clusterDumper.numWords	Specify the number of characters of the top terms to print (required).	
clusterDumper.samplePoints	Specify the maximum number of points to include per cluster. The default is to include all points (required).	
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Fuzzy K-Means

Runs the Mahout Fuzzy K-Means algorithm. The input data needs to be a sequence file.

## Fuzzy K-Means example



### User Properties

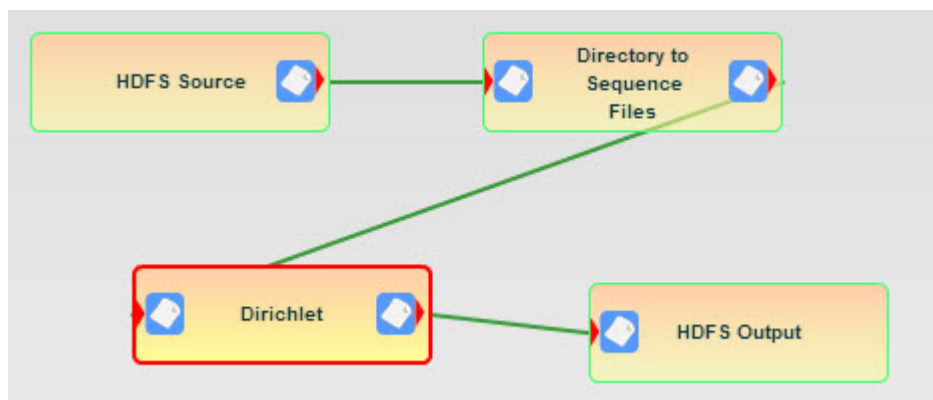
Parameter	Value	Example
minSupport	1	1
analyzerName	Insert the class name of the analyzer	
chunkSize	Choose the chunk size in MG. Defaults to 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TDIF or TF	TF
minDF	Choose the minimum document frequency. The default is 1.	1
maxDFPercent	Insert the maximum percentage of document for the DF. This can be used to remove really high frequency terms. Expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	99
maxDFSigma	Insert the portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to be specified is 3.0. In the case where the value is less than 0, no vectors will be filtered. The default is -1.0. This will override the value of maxDFPercent.	3.0
minLLR	Insert with the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Insert the norm to use that is expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = biGrams, 3 = triGrams, and so on). The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE	TRUE
namedVector	Choose either TRUE or FALSE	TRUE
distanceMeasure	Choose ChebyshevDistanceMeasure, CosineDistanceMeasure, EuclideanDistanceMeasure, MahalanobisDistanceMeasure, ManhattanDistanceMeasure, MinkowskiDistanceMeasure, SquaredEuclideanDistanceMeasure, TanimotoDistanceMeasure, WeightedEuclideanDistanceMeasure, or WeightedManhattanDistanceMeasure	ChebyshevDistanceMeasure
numClusters	Choose the number of clusters to create.	2
kmeansClusters	Select the input centroids as vectors. This must be a sequence file of writeable cluster/canopy. If K is also specified, then a random set of vectors will be selected and written out to this	

	path first.	
convergenceDelta	Insert the convergence delta value. The default is 0.5.	0.5
maxIter	Insert the maximum number of iterations (required).	
clustering	Choose either TRUE or FALSE	TRUE
outlierThreshold	Specify the outlier threshold value.	
m	Specify the coefficient normalization factor. This must be greater than 1 (required).	2
emitMostLikely	Choose either TRUE or FALSE	TRUE
threshold	Specify the PDF threshold used for cluster determination. The default is 0.	0
overwrite	Choose either TRUE or FALSE	TRUE
clusterDumper.outputFormat	Choose the optional output format for writing out the results. The options are: text, csv, or graph_ml.	
clusterDumper.substring	Specify the number of characters of the asformatstring() to print (required).	csv
clusterDumper.numWords	Specify the number of characters of the top terms to print (required).	
clusterDumper.samplePoints	Specify the maximum number of points to include per cluster. The default is to include all points (required).	
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Dirichlet

Runs the Mahout Dirichlet algorithm. The input must be a sequence file.

### Dirichlet example



### User Properties

Parameter	Value	Example
minSupport	1	1
analyzerName	Insert the class name of the analyzer	
chunkSize	Choose the chunk size in MG. Defaults to 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TDIF or TF	TF
minDF	Choose the minimum document frequency. The default is 1.	1
maxDFPercent	Insert the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If	99

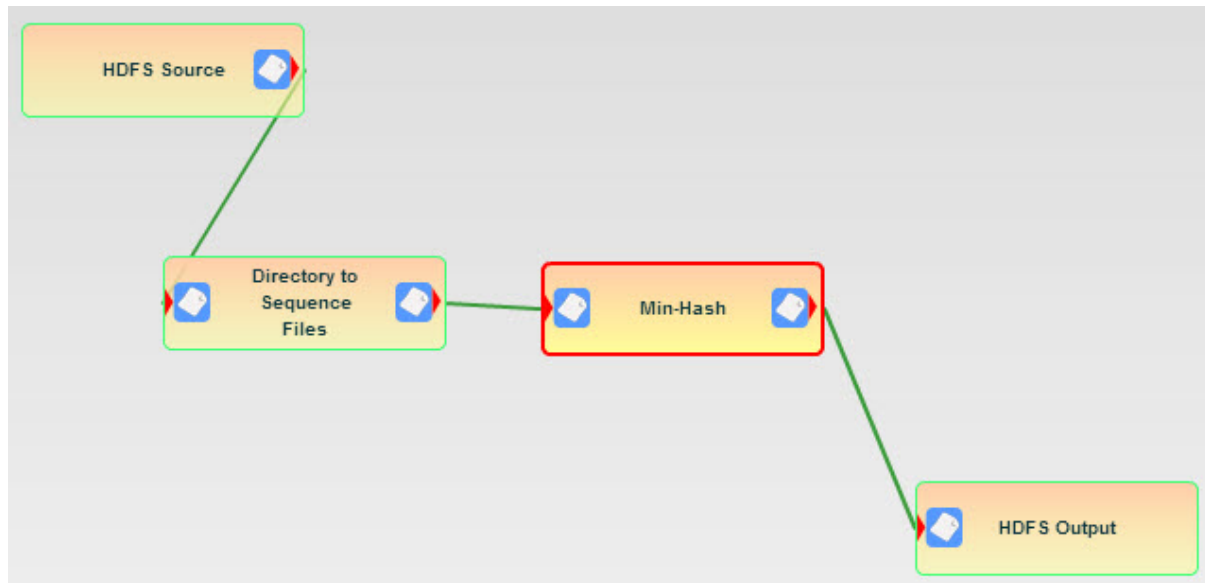


	maxDFSigma is also set, it will override this value.	
maxDFSigma	Insert the portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to be specified is 3.0. In the case where the value is less than 0, no vectors will be filtered. The default is -1.0. This will override the value of maxDFPercent.	3.0
minLLR	Insert with the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Insert the norm to use that is expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = biGrams, 3 = triGrams, and so on). The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE	TRUE
namedVector	Choose either TRUE or FALSE	TRUE
distanceMeasure	Choose ChebyshevDistanceMeasure, CosineDistanceMeasure, EuclideanDistanceMeasure, MahalanobisDistanceMeasure, ManhattanDistanceMeasure, MinkowskiDistanceMeasure, SquaredEuclideanDistanceMeasure, TanimotoDistanceMeasure, WeightedEuclideanDistanceMeasure, or WeightedManhattanDistanceMeasure	ChebyshevDistanceMeasure
modelPrototype	Insert the model distribution prototype vector class name. Defaults to randomaccessparsevector.	
numClusters	Choose the number of clusters to create.	2
convergenceDelta	Insert the convergence delta value. The default is 0.5.	0.5
maxIter	Insert the maximum number of iterations (required).	2
clustering	Choose either TRUE or FALSE	TRUE
emitMostLikely	Choose either TRUE or FALSE	TRUE
threshold	Specify the PDF threshold used for cluster determination. The default is 0.	0
alpha	Specify the alpha0 value for the Dirichlet distribution. The default is 1.0.	1.0
overwrite	Choose either TRUE or FALSE	TRUE
clusterDumper.outputFormat	Choose the optional output format for writing out the results. The options are: text, csv, or graph_ml.	csv
clusterDumper.substring	Specify the number of characters of the asformatstring() to print (required).	
clusterDumper.numWords	Specify the number of characters of the top terms to print (required).	
clusterDumper.samplePoints	Specify the maximum number of points to include per cluster. The default is to include all points (required).	
delimiter	Choose the delimiter to delimit the output. For example, the tab delimiter is \t. The default is a space character.	/t

## Min-Hash

Min-Hash clustering creates clusters based on data similarity using a hashing of the data. Similar items are grouped together using the hash function that serves as an approximation of the Jaccard similarity between different data items.

## Min-Hash example



### User Properties

Parameter	Value	Example
minSupport	1	1
analyzerName	Insert the class name of the analyzer	
chunkSize	Choose the chunk size in MG. Defaults to 64MB. The value must be between 100MB and 10000MB.	100
weight	Choose either TDIF or TF	TF
minDF	Choose the minimum document frequency. The default is 1.	1
maxDFPercent	Insert the maximum percentage of document for the DF. This can be used to remove really high frequency terms expressed as an integer between 0 and 100. The default is 99. If maxDFSigma is also set, it will override this value.	99
maxDFSigma	Insert the portion of the TF (TF-IDF) vectors to be used expressed in times the standard deviation (Sigma) of the document frequencies of these vectors. This can be used to remove really high frequency terms expressed as a double value. A good value to be specified is 3.0. In the case where the value is less than 0, no vectors will be filtered. The default is -1.0. This will override the value of maxDFPercent.	3.0
minLLR	Insert with the minimum log likelihood ratio (float). The default is 1.0.	1.0
numReducers	Insert the number of reduce tasks. The default value is 1.	1
norm	Insert the norm to use that is expressed as either a float or "INF" if you want to use the infinite norm. The value must be greater than or equal to 0. The default is not to normalize.	INF
logNormalize	Choose either TRUE or FALSE	TRUE
maxNGramSize	Use the maximum size of NGrams to create (2 = biGrams, 3 = triGrams, and so on). The default value is 1.	1
sequentialAccessVector	Choose either TRUE or FALSE	TRUE
namedVector	Choose either TRUE or FALSE	TRUE
minClusterSize	Specify the minimum points inside a cluster.	
minVectorSize	Specify the minimum size of vector to be hashed.	
hashType	Choose the type of hash function to use: linear, polynomial, and murmur.	
numHashFunctions	Specify the number of hash functions to be used.	
keyGroups	Specify the number of key groups to be used.	

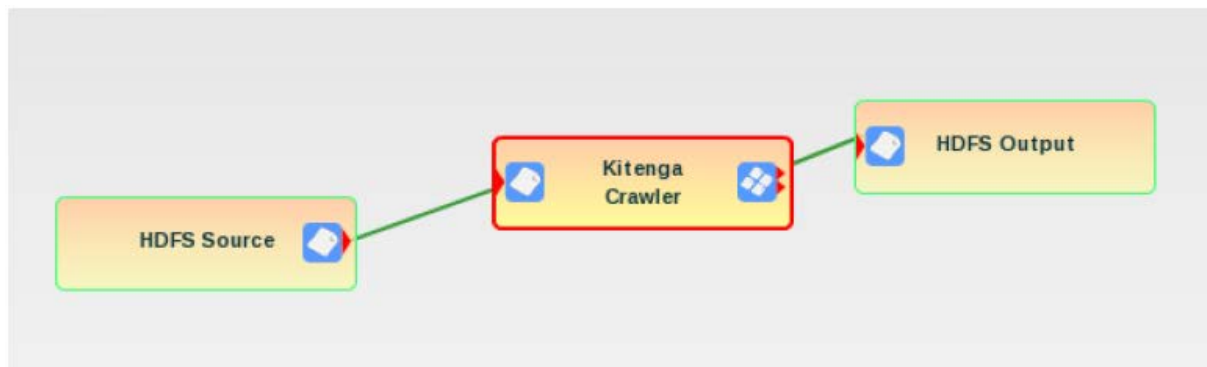
numReducers	Insert the number of reduce tasks. The default value is 2.	2
debugOutput	Choose either TRUE or FALSE	TRUE

## Crawlers

### Statistica crawler

Crawls a set of URLs as specified in the input files and distributes them among the nodes. Only sites specified in the input URLs will be crawled, i.e., no external sites will be crawled. You must specify the User Agent to identify who is doing the crawling.

#### Statistica crawler example



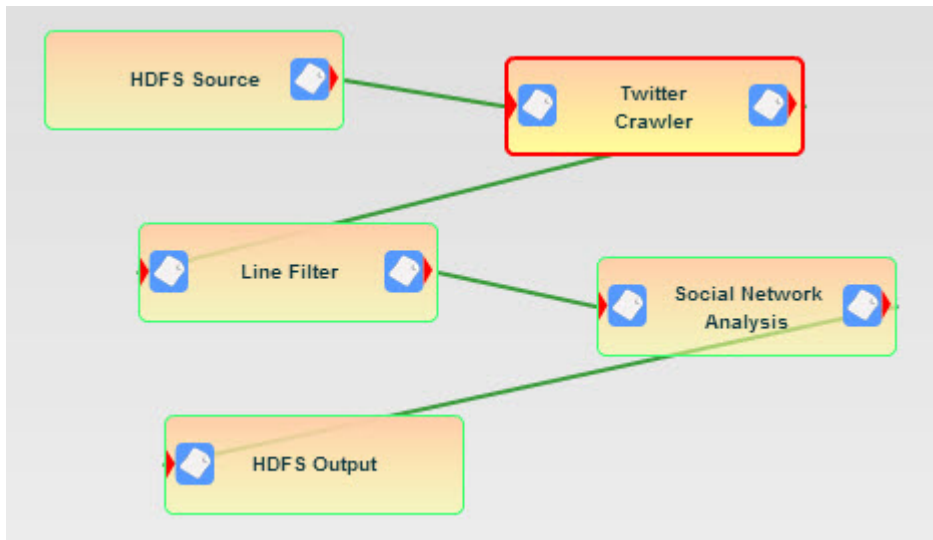
#### User Properties

Parameter	Value	Example
crawler.useragent	Statistica-Crawler-1.0	1
crawler.threaded	Choose either TRUE or FALSE	TRUE
crawler.followredirects	Choose either TRUE or FALSE	TRUE
crawler.exclude.extensions.list	List URL extensions delimited by white space to exclude in the crawling.	
crawler.includefilters	List regular expressions delimited by white space to indicate the URLs to include in the crawling.	
crawler.excludefilters	List regular expressions delimited by white space to indicate the URLs to exclude in the crawling.	
crawler.maxdepth	Specify the maximum crawl depth. The default is 4.	4
crawler.maxiterations	Specify the maximum crawl iterations. The default is 2048.	2048
crawler.user	Specify the username in crawling URLs that require login credentials.	
crawler.password	Specify the password in crawling URLs that require login credentials.	

### Twitter crawler

Fetches the most recent tweets related to a topic keyword using the Twitter Search API. Note that you are limited to 1500 fetches per hour per IP address, or you may be blacklisted by Twitter. You must supply your Twitter authentication.

## Twitter crawler example



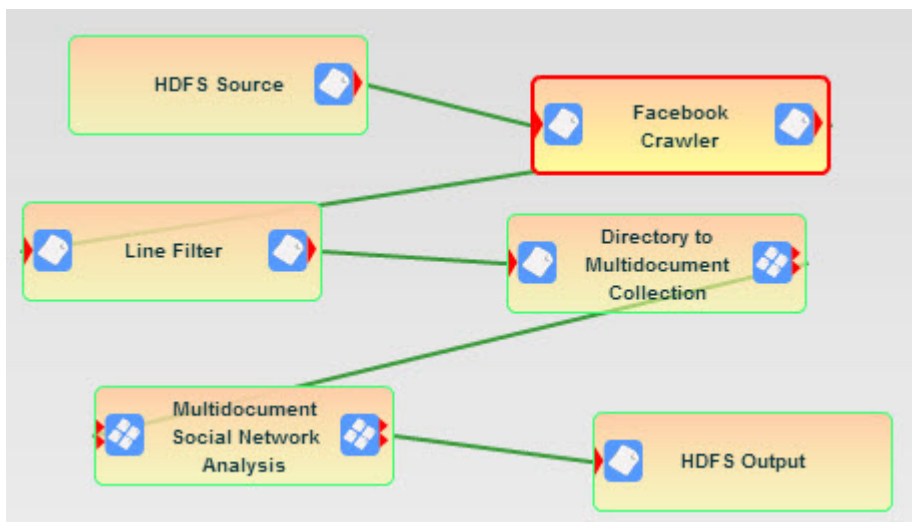
### User Properties

Parameter	Value	Example
twitter.consumer.key		VU6t26M4qsjZwhgvSPsGKA
twitter.consumer.secret		Spa4PzteOoLaMMUYa8BtbOcyzYKnkfzYjOA9iKDTM3Gx0
twitter.access.token		lGmHSif6CrglZ8kFpnxQzq1Mu0FmCrp9VdahhUf
twitter.access.token.secret		HyV5zsMvgKhIOJXft9cbpt4i83ebSTH19h0gMdaQ

## Facebook crawler

Fetches the most recent public Facebook posts related to a topic keyword using the Facebook Search API. Supplies a Facebook application access token, a maximum query size (defaults to 1000) and the keywords in the input file on HDSF. The output is a tab-delimited row of data.

### Facebook crawler example



### User Properties

Parameter	Value	Example
facebook.url		https://graph.facebook.com/search?type=post
facebook.query.limit	1000	

facebook.app.id

---

facebook.app.secret

---

## Filters

### Pig script

Executes a self-contained Pig Script. This processing element can be stand-alone, or it can be the first processing element in a chain of processing elements. It should never be followed directly with a sink processing element.

#### Pig script example



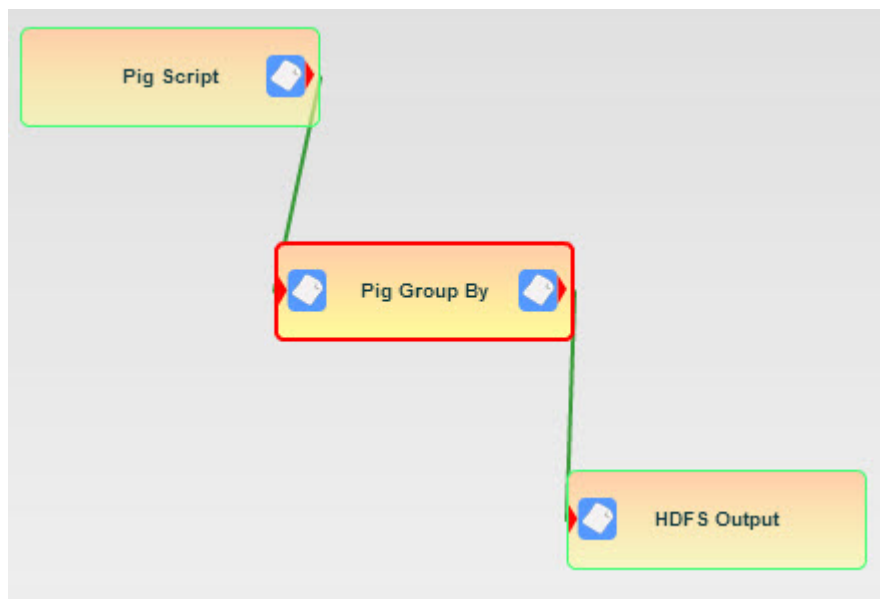
#### User Properties

Parameter	Value	Example
scriptFile	Use a fully qualified file system	hdfs://[Server Name]/user/filter/input/batting_score_lead_element_test.pig
path	Use HDFS path to allow outputs to be passed on to other processing elements	hdfs://[Server Name]/user/filter/testPigScript/out

### Pig Group By

Allows an input to be grouped upon a data column in the input.

#### Pig Group By example



#### User Properties

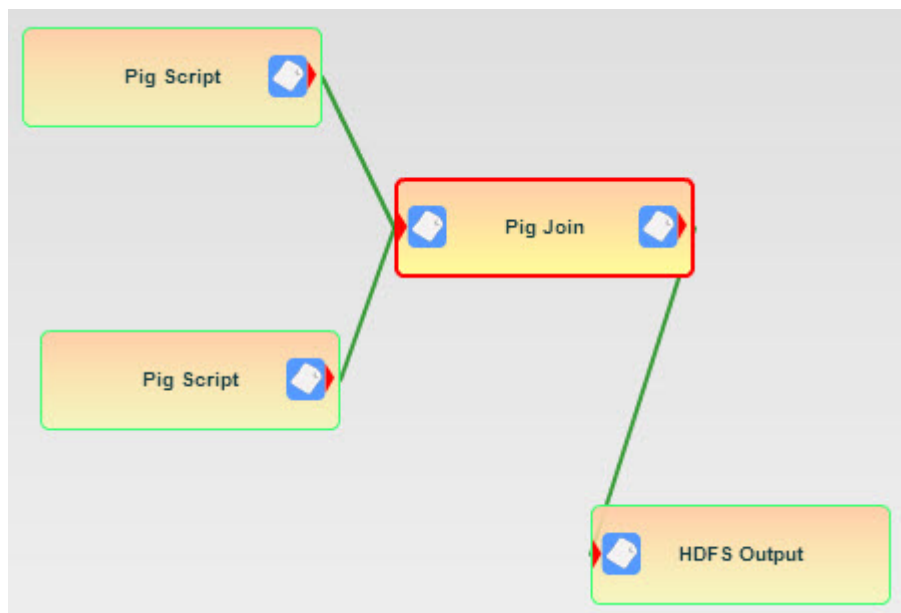
Parameter	Value	Example
-----------	-------	---------

schema	Insert a schema using pig latin syntax	yearID:chararray, teamID:chararray, runs:double
groupBy	Replace with input columns to use for the grouping separated with commas	teamID
inputDelimiter	Choose a delimiter for input. Defaults to comma (,).	/t
outputDelimiter	Choose a column delimiter for output. Defaults to comma (,).	

## Pig join

Allows two inputs to be joined based on a matching data element in both. It is necessary to define a schema, the column delimiter, and the common column to join upon.

### Pig join example



### User Properties

Parameter	Value	Example
inputPath1	Use the HDFS input path	hdfs://[Server Name]/user/filter/testPigJoin/out/pigtuples/batting ScoreData
inputDelimiter 1	Choose the delimiter for input. Defaults to comma (,).	/t
schema1	Insert a schema using pig latin syntax.	year:int, playerID:chararray, runs:double
joinCol1	Insert a column to use for the joining.	playerID
inputPath2	Use the HDFS input path	hdfs://[Server Name]/user/filter/testPigJoin/out/pigtuples/players ListData
inputDelimiter 2	Choose the delimiter for input. Defaults to comma (,).	/t
schema2	Insert a schema using pig latin syntax.	playerID:chararray, fname:chararray, lname:chararray
joinCol2	Insert a column to use for the joining.	playerID
outputDelimiter	Choose a column delimiter for output. Defaults to comma (,).	,

## Release 2.0 and earlier processing elements

### Data sources

Data source elements are inputs to analytics workflows using Processing Elements and Data Sinks.

#### File source

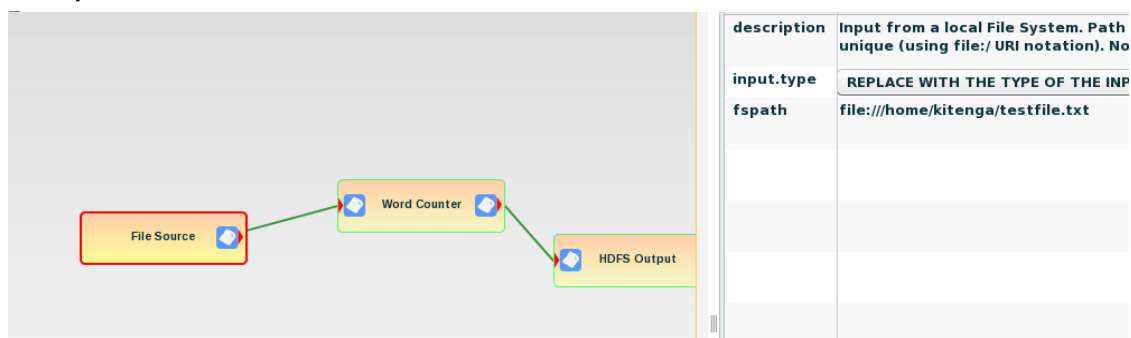
##### Description

The File Source pulls data from files on the computer's file system. The path must be specified using a file system URI. Also, for a cluster environment, DataNodes will try to pull data from the same path on each DataNode. Generally, that means that the file must be mounted using NFS on each computer.

##### Parameters

Name	Type	Description	Example
input.type	Optional	Select if directing special Mahout sources	
fspath	Required	URL of file, may include glob notation	<a href="file:///home/Statistica/testfile.txt">file:///home/Statistica/testfile.txt</a>

##### Example



#### HDFS source

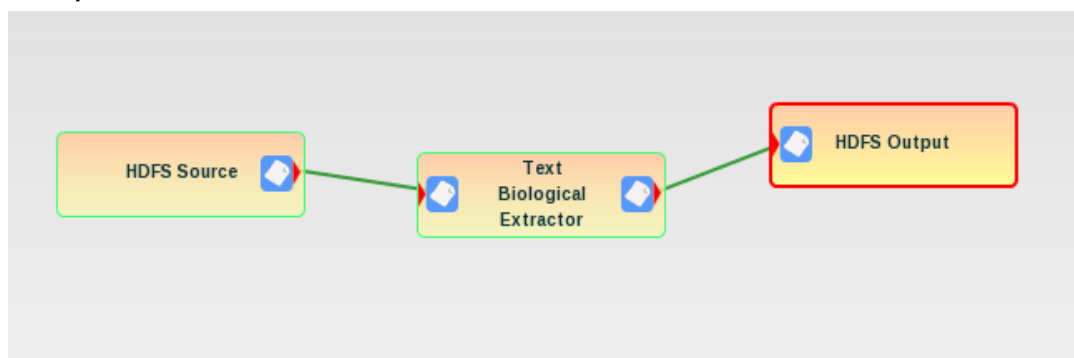
##### Description

The HDFS Source (Hadoop Distributed File System Source) pulls data from records in the Hadoop Distributed File System. The records can be individual files or directories containing multiple files. For the latter, however, default behavior in Hadoop is to not recursively descend through file structures.

##### Parameters

Name	Type	Description	Example
input.type	Optional	Select if directing special Mahout sources	
fspath	Required	URL of file, may include glob notation	<a href="file:///home/Statistica/testfile.txt">file:///home/Statistica/testfile.txt</a>

## Example



## Amazon S3 source

### Description

The Amazon S3 Source pulls data from an Amazon S3 Bucket. An Amazon S3 Bucket is identified by a unique bucket string and the files to be processed within that bucket must be identified. In addition, the user's access key and secret access key are needed to use the data in the bucket.

### Parameters

Name	Type	Description	Example
input.type	Optional	Select if directing special Mahout sources	
aws.accessKey	Required	The user access key for Amazon S3 services	AKIAIOSFODNN7EXAMPLE
aws.secretAccessKey	Required	The user secret access key for S3 services	wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKE
aws.endpoint	Required	The URL for accessing S3 services	<a href="https://s3.amazonaws.com">s3.amazonaws.com</a>
aws.bucket	Required	The S3 bucket identifier	Statistica-bucket-1
aws.path	Required	The path of the files with the bucket	Myfile1.txt

## Example



## Database source

### Description

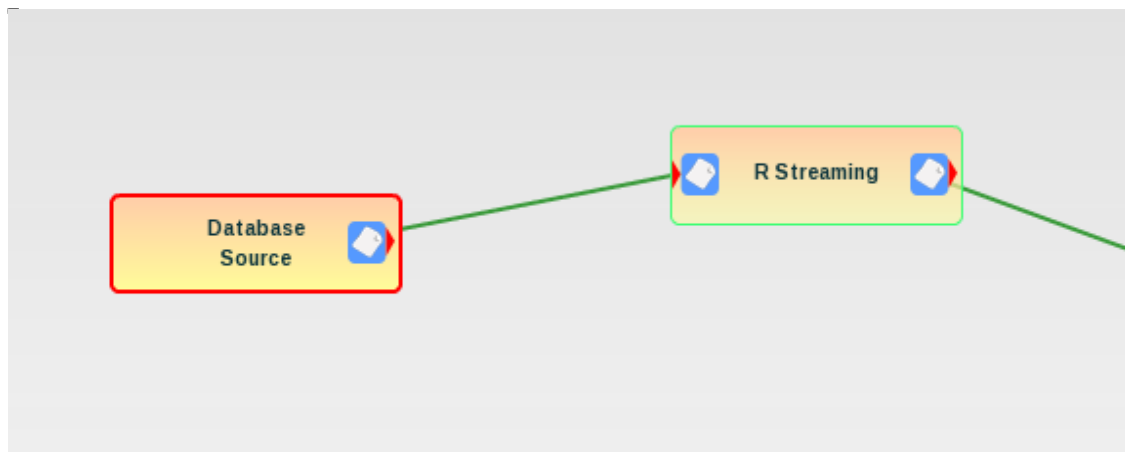
The Database Source pulls data from a database using a JDBC bridge. The JDBC connection can be specified either using parameters or by connecting and selecting the required tables and columns directly from the database itself. To use a Database Source, a range of parameters are needed, reflecting the database connectivity strings, user identity, and the target tables, columns, or queries for database interrogation.



## Parameters

Name	Type	Description	Example
input.type	Required	Type of input for Mahout	MAHOUT_DATA_TRAIN
mapred.jdbc.driver.class	Required	Class for the JDBC driver	com.mysql.jdbc.Driver
jdbc.driver.jar.path	Required	The JDBC driver jar (must be present in all computing nodes)	/tmp/mysql-connector-java-5.0.4-bin.jar
mapred.jdbc.url	Required	JDBC URL	jdbc:mysql://localhost:te st
mapred.jdbc.username	Required	User name for DB access	admin
mapred.jdbc.password	Required	Password for DB access	admin
mapred.jdbc.schema	Required	Schema for the database	my_schema
mapred.jdbc.input.table.name	Required	Table from the database	my_table
mapred.jdbc.input.field.names	Required	Fields to extract from DB	my_column1,mycolumn2
jdbc.id.column.name	Required	ID column in DB	id
mapred.jdbc.input.conditions	Optional	The where clause in a SQL query but without the keyword "where"	gender='male'
mapred.jdbc.input.query	Optional	A SQL query	select * from person
mapred.jdbc.input.count.query	Optional	The SQL count query corresponding to the "mapred.jdbc.input.query"	select count(1) from person
delimiter	Optional	The delimiter to delimit the output from the database query	,
output.sequence.file	Optional	Whether the output from the database query should be sequence file or not	FALSE
run.mahout.naivebayes	Optional	Whether to run Naïve Bayes using Mahout machine learning algorithm	TRUE
mahout.naivebayes.data.label	Optional	The known data label when running Naïve Bayes	gender
columns.to.exclude	Optional	The data columns to exclude from the output of the database query, delimited by comma	id,creation_time

## Example



# Extractors

Extractors are pattern matching and content extraction elements that work on unstructured or semi-structured data sets.

## Text biological extractor

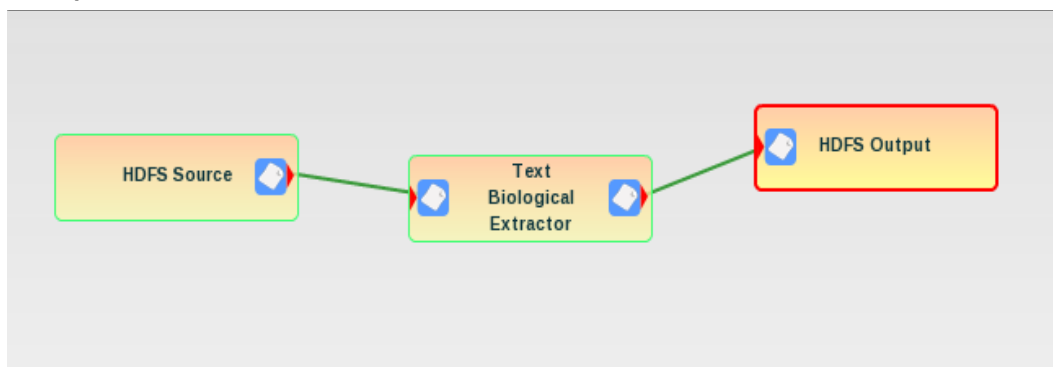
### Description

Extract biological terminology and phrases from text documents. The categories of extracted entities include protein names, DNA/RNA sequence names, cell lines, and related concepts. The document must be in plain text format. A directory of text files may also be submitted. The output is a file of entries, one per line, which contains the category followed by a tab character, the extracted entity, a tab, and then the count of the entity across all of the input documents.

### Parameters

None

### Example



## Multidocument text biological extractor

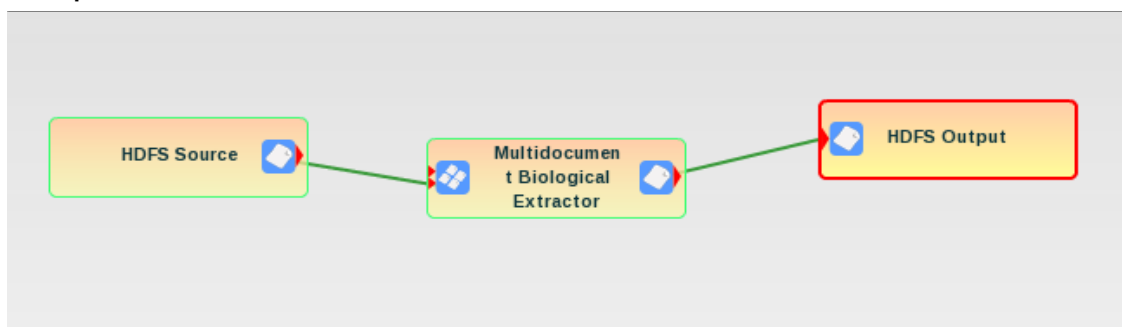
### Description

Extract biological terminology and phrases from text documents in a Multidocument Collection. The categories of extracted entities include protein names, DNA/RNA sequence names, cell lines, and related concepts. The documents in the Multidocument Collection must be in plain text format. The output is a file of entries, one per line, which contains the category followed by a tab character, the extracted entity, a tab, and then the count of the entity across all of the input documents.

### Parameters

None

### Example



## XML biological extractor

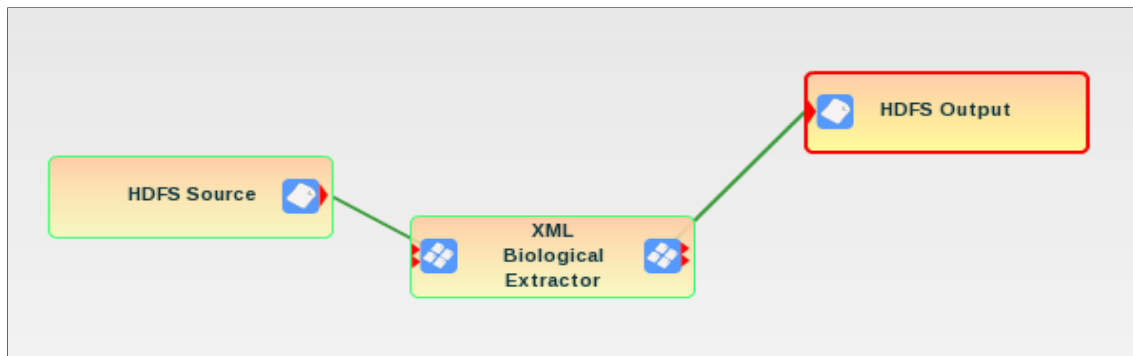
### Description

Extract biological terminology and phrases from text documents in a Multidocument Collection, creating an XML representation of the original document content and the new entities as properties. The categories of extracted entities include protein names, DNA/RNA sequence names, cell lines, and related concepts. The documents in the Multidocument Collection must be in plain text format. The output is a Multidocument Collection containing XML representations of the original documents with the entities added as properties.

### Parameters

None

### Example



## Text English extractor

### Description

Extract named entities from text documents. The available categories of extracted entities include People, Location, Organization, Date, JobTitle, Money, and URL. The default configuration of the Processing Element only defines People, Places, and Organizations, however. The input documents (either single documents or directories of documents) can be in either plain text or HTML format. The output is a file of entries, one per line, which contains the category followed by a tab character, the extracted entity, a tab, and then the count of the entity across all of the input documents.

Customization of entities and categories can be accomplished by expanding the lists of categories and providing lists of words/phrases as well as rules for the mapping of those categories.

### Customization

The English Extractor and Multidocument English Extractor can be customized through a combination of two methods. First, specific terms and phrases can be added to a gazetteer file. Second, rules that combine or modify those phrases can be specified. Usually, it is adequate to supply term lists, however. To do this customization, modify the files in the GATE subdirectory of the Statistica Analytic Suite installation directory. Note that you will need to copy the changes to all DataNodes after completion. In the following example, an extractor category of Hospitals will be created:

1. Create a new file in GATE/plugins/ANNIE/resources/gazetteer/ called hospitals.lst and add the following lines to the file:

**Super Hospital**

**Special Hospital**

**Regional Medical Center**

2. Edit GATE/plugins/ANNIE/resources/gazetteer/lists.def to add the new hospitals.lst to the end of the file, including the category of Hospital:

**hospital.lst:Hospital**

3. Create a new file in GATE/plugins/ANNIE/resources/NE/ called hospitals.jape that contains the following content:

Phase: hospital

Input: Token Lookup

Options: control = brill

Rule: Hospital

Priority: 20

```
( {Lookup.majorType == "Hospital"}
```

```
):hospital -->
```

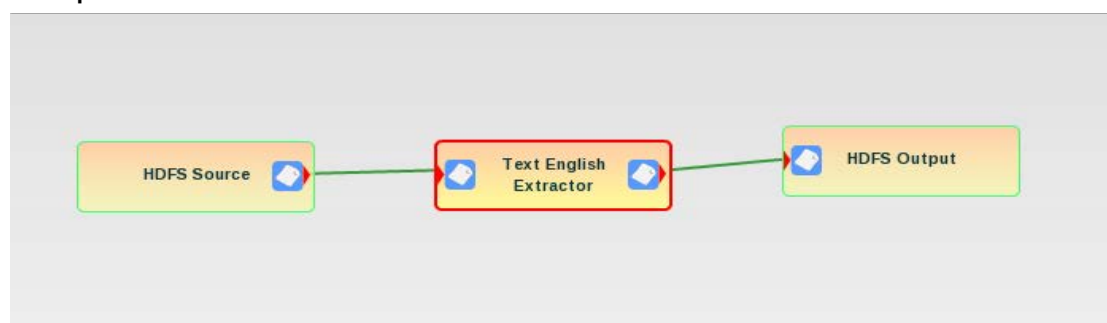
```
:hospital.Hospital = { rule = "Hospital" }
```

4. Edit GATE/plugins/ANNIE/resources/NE/main.jape to include the term **hospitals** at the end of the file (no quotes).
5. You can then add **Hospital** to the categories parameter to extract the hospital name occurrences from text files.

### Parameters

Name	Type	Description	Example
extractor.configfiles	Required	Configuration directory for the extraction system. This directory must exist on all DataNodes and be readable by the mapreduce system.	/home/Statistica/Statistica-13.1 / GATE
extractor.categories	Required	The comma-separated categories of named entities for extraction	Date,Person,Location

### Example



## Multidocument English extractor

### Description

Extract named entities from Multidocument Collections. The available categories of extracted entities include People, Location, Organization, Date, JobTitle, Money, and URL. The default configuration of the Processing Element only defines People, Places, and Organizations, however. The input documents in a Multidocument Collection can be in either plain text or HTML format. The output is a file of entries, one per line, which contains the category followed by a tab character, the extracted entity, a tab, and then the count of the entity across all of the input documents.

Customization of entities and categories can be accomplished by expanding the lists of categories and providing lists of words/phrases as well as rules for the mapping of those categories.

## Customization

The English Extractor and Multidocument English Extractor can be customized through a combination of two methods. First, specific terms and phrases can be added to a gazetteer file. Second, rules that combine or modify those phrases can be specified. Usually, it is adequate to supply term lists, however. For this customization, modify the files in the GATE subdirectory of the Statistica Analytic Suite installation directory. Note that you will need to copy the changes to all DataNodes after completion. In the following example, an extractor category of Hospitals will be created:

1. Create a new file in GATE/plugins/ANNIE/resources/gazetteer/ called hospitals.lst and add the following lines to the file:

```
Super Hospital
Special Hospital
Regional Medical Center
```

2. Edit GATE/plugins/ANNIE/resources/gazetteer/lists.def to add the new hospitals.lst to the end of the file, including the category of Hospital:

```
hospital.lst:Hospital
```

3. Create a new file in GATE/plugins/ANNIE/resources/NE/ called hospitals.jape that contains the following content:

```
Phase: hospital
Input: Token Lookup
Options: control = brill
Rule: Hospital
Priority: 20

( {Lookup.majorType == "Hospital"}
):hospital -->

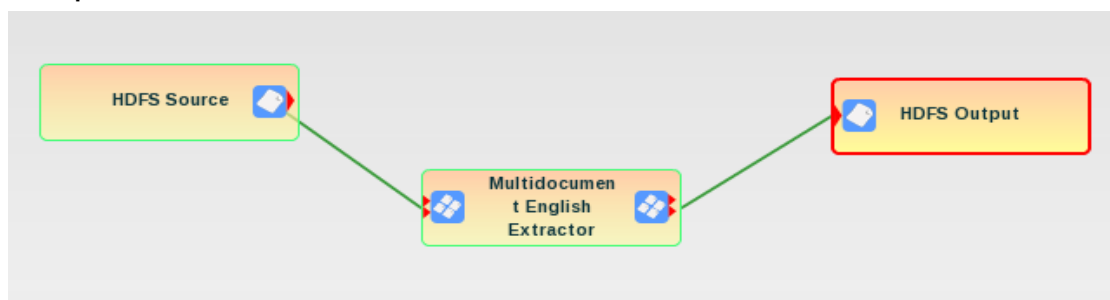
:hospital.Hospital = { rule = "Hospital" }
```

4. Edit GATE/plugins/ANNIE/resources/NE/main.jape to include the term **hospitals** at the end of the file (no quotes).
5. You can then add **Hospital** to the categories parameter to extract the hospital name occurrences from text files.

## Parameters

Name	Type	Description	Example
extractor.configfiles	Required	Configuration directory for the extraction system. This directory must exist on all DataNodes and be readable by the mapreduce system.	/home/Statistica/Statistica-13.1 / GATE
extractor.categories	Required	The comma-separated categories of named entities for extraction	Date,Person,Location

## Example



## Social network analysis

### Description

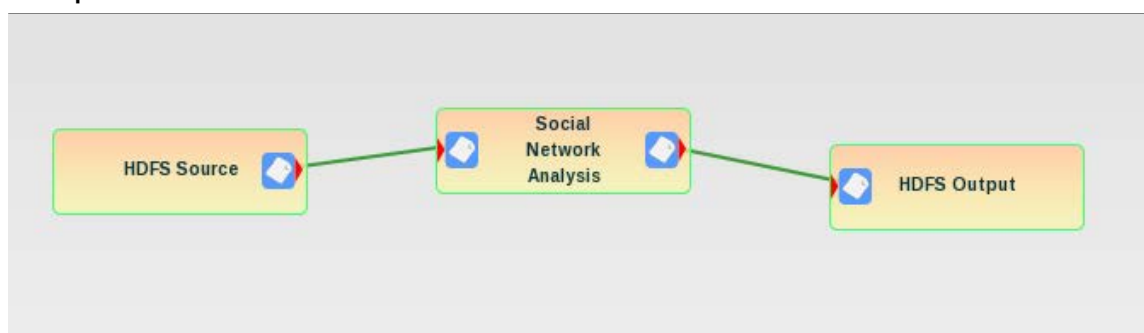
Extract named entities from text documents and calculate the co-occurrence patterns of the entities within the same sentence context. The available categories of extracted entities include People, Location, Organization, Date, JobTitle, Money, and URL. The default configuration of the Processing Element only defines People, Location, and Organizations, however. The input documents can be in either plain text or HTML format. The output is a file of entries, one per line, which contains the pair of co-occurring entities followed by their normalized counts across all of the input documents.

Customization of entities and categories can be accomplished by expanding the lists of categories and providing lists of words/phrases as well as rules for the mapping of those categories. See description of customization for the Text English Extractor for details on how to modify those categories.

### Parameters

Name	Type	Description	Example
extractor.configfiles	Required	Configuration directory for the extraction system. This directory must exist on all DataNodes and be readable by the mapreduce system.	/home/Statistica/Statistica-13.1 / GATE
extractor.categories	Required	The comma-separated categories of named entities for extraction	Date,Person,Location

## Example



## Multidocument social network analysis

### Description

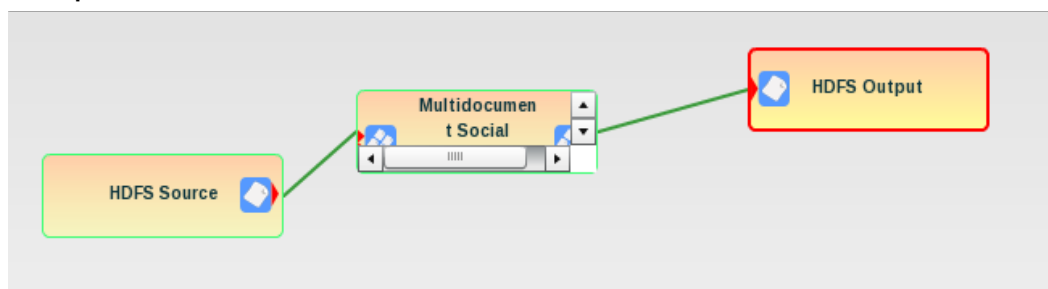
Extract named entities from Multidocument Collections and calculate the co-occurrence patterns of the entities within the same sentence context. The available categories of extracted entities include People, Location, Organization, Date, JobTitle, Money, and URL. The default configuration of the Processing Element only defines People, Location, and Organizations, however. The input documents in the Multidocument Collection can be in either plain text or HTML format. The output is a file of entries, one per line, which contains the pair of co-occurring entities followed by their normalized counts across all of the input documents.

Customization of entities and categories can be accomplished by expanding the lists of categories and providing lists of words/phrases as well as rules for the mapping of those categories. See description of customization for the Text English Extractor for details on how to modify those categories.

#### Parameters

Name	Type	Description	Example
extractor.configfiles	Required	Configuration directory for the extraction system. This directory must exist on all DataNodes and be readable by the mapreduce system.	/home/Statistica/Statistica-13.1 / GATE
extractor.categories	Required	The comma-separated categories of named entities for extraction	Date,Person,Location

#### Example



## XML English extractor

#### Description

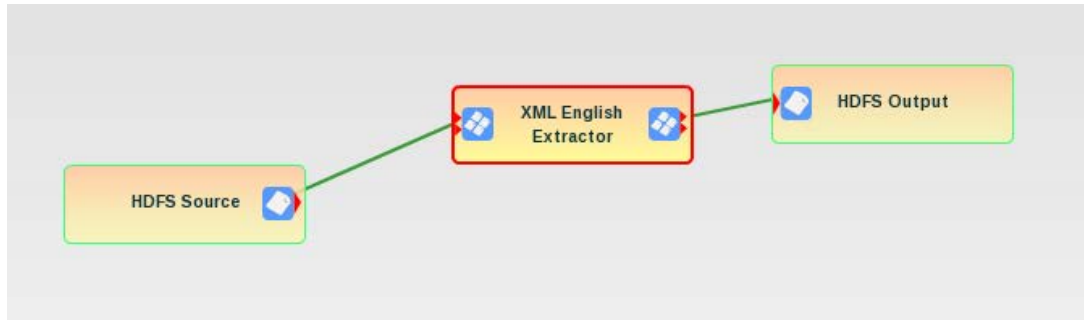
Extract named entities from Multidocument Collections and create an XML representation of the documents and the named entities. The available categories of extracted entities include People, Location, Organization, Date, JobTitle, Money, and URL. The default configuration of the Processing Element only defines People, Location, and Organizations, however. The input documents in the Multidocument Collection can be in either plain text or HTML format. The output is a Multidocument Collection of files that contain XML representations of the original content and the extracted named entities. For HTML input documents, the existing HTML metadata fields will be converted into XML fields as well.

Customization of entities and categories can be accomplished by expanding the lists of categories and providing lists of words/phrases as well as rules for the mapping of those categories. See description of customization for the Text English Extractor for details on how to modify those categories.

#### Parameters

Name	Type	Description	Example
extractor.configfiles	Required	Configuration directory for the extraction system. This directory must exist on all DataNodes and be readable by the mapreduce system.	/home/Statistica/Statistica-13.1 / GATE
extractor.categories	Required	The comma-separated categories of named entities for extraction	Date,Person,Location

## Example



## HTML to XML converter

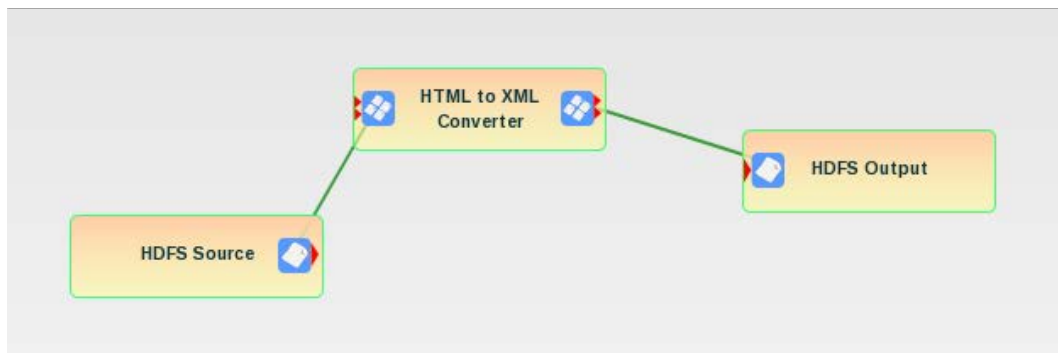
### Description

Convert HTML documents contained in a Multidocument Collection into XML suitable for Solr Search Provider or other uses. For HTML input documents, the existing HTML metadata fields will be converted into XML fields as well.

### Parameters

None

## Example



## HTML text extractor

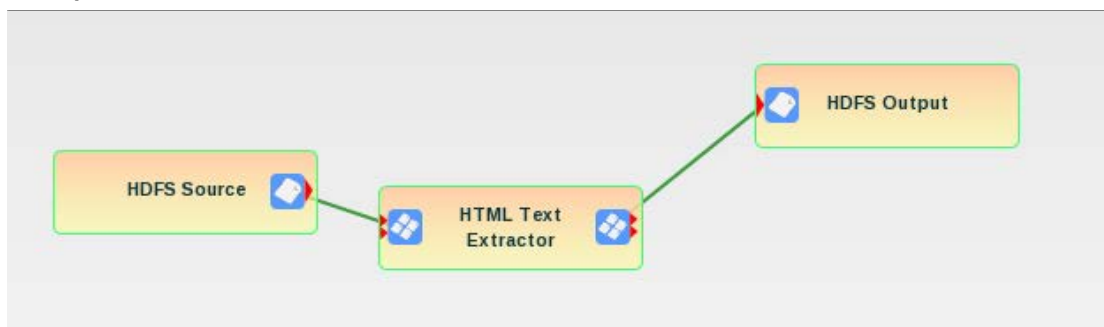
### Description

Extract text from HTML documents in a Multidocument Collection and put the text into the output Collection.

### Parameters

None

## Example





## HTML link extractor

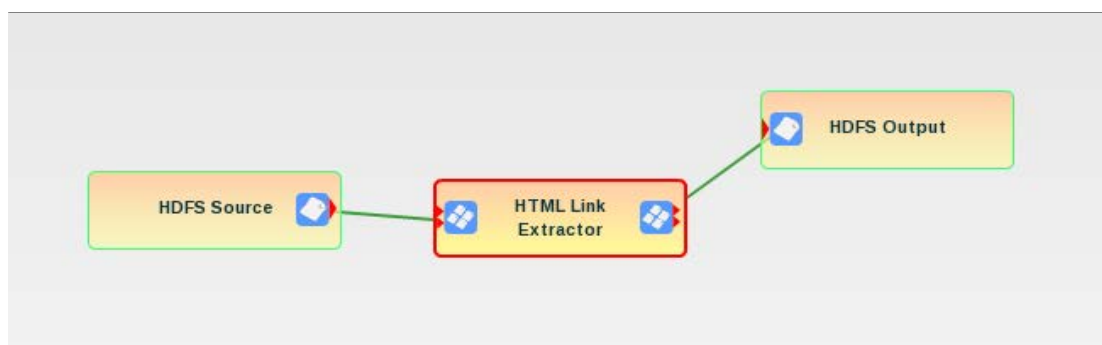
### Description

Extract links from HTML documents in a Multidocument Collection and puts the links into the output Collection. The links are rendered, one per line, in the output documents.

### Parameters

None

### Example



## Classifiers

Classifier Processing Elements are used for machine learning based classification. These components can be part of the classification process (data prep, data conversion, training, classification) or can be completely self-contained for simplicity.

### Weka PMML MapReduce

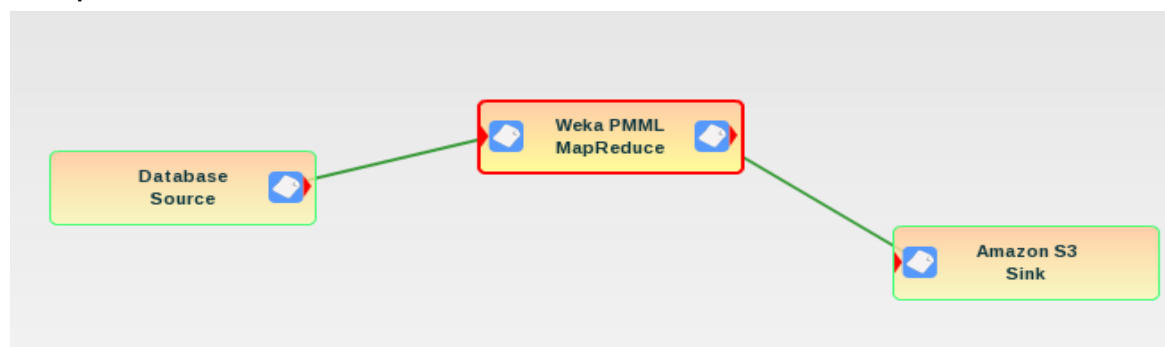
#### Description

For each line of an input file, the Weka PMML MapReduce PE executes a PMML model to attempt to classify the input line and appends the resulting class to the data line in the output.

#### Parameters

Name	Type	Description	Example
weka.pmml	Required	Specifies the PMML for the .	<?xml version="1.0" ?> <PMML version="3.2" xmlns="http://www.dmg.org/PMML-3_2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">...
linesplitter.splitkey	Required	Regular expression for the input split	\t

## Example



## Other Classifier Processing Elements

Name	Description
Convert to Sparse Vectors	Converts an input source consisting of a SequenceFile with possible training labels into a Sparse Vector representation
Split Sparse Vectors	Split Sparse Vectors into training and test data sets
Train Naïve Bayes	Train a Naïve Bayes model based on the training data split (optionally cross-validated)
Test Naïve Bayes	Test a Naïve Bayes model using the test data split
Classify Naïve Bayes	Using a trained Naïve Bayes model, classify new data items in a similar representation
Partial Decision Forest	Use a Partial Decision Forest training and classification model
Breiman Decision Forest	Use a Breiman-style Decision Forest for training and classification

## Clustering

Clustering algorithms group or cluster data items based on feature similarities.

Name	Description
K-Means	Use a K-Means clustering algorithm to cluster data items.
Fuzzy K-Means	Use a variant of a K-Means algorithm that allows for fuzzy cluster membership for clustering
Train Naïve Bayes	Train a Naïve Bayes model based on the training data split (optionally cross-validated)
Dirichlet	Use a Dirichlet clustering algorithm based on a Dirichlet distribution model
Min-Hash	Cluster based on a MinHash or locally sensitive hashing function

## Collaborative filtering

Collaborative Filtering algorithms use the large-scale patterns between data items to suggest unknown items for new data items.

Name	Description
ALS-WR	Use an Alternative Least Squares with Weight Lambda Filtering algorithm for data recommendations

## Crawlers

Crawlers retrieve remote files and entries from social media sources and websites.

## Statistica crawler

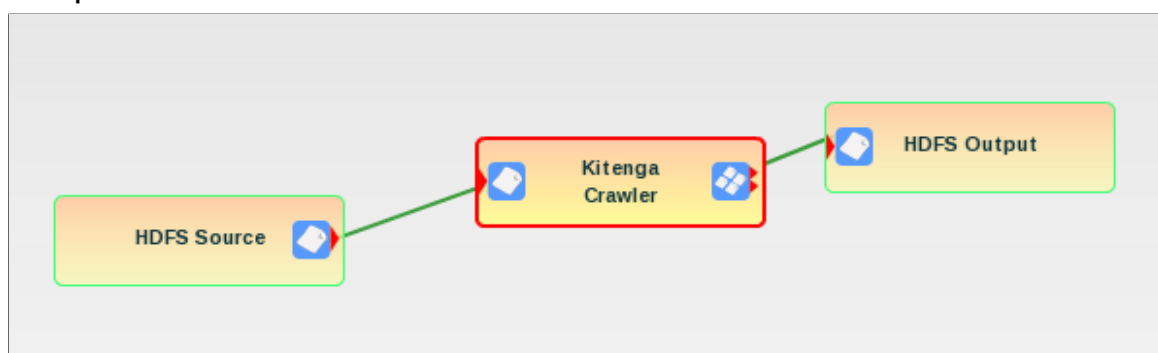
### Description

For each line of an input file that is a URL, the Statistica Crawler will attempt to retrieve the target of the URL as well as traverse the linked documents present in the page. Set various parameters to control the retrieved content, which may be of any form, including binaries, etc. The output is a Multidocument Collection that combines together all of the retrieved results with the URL as the key and the content as the value.

### Parameters

Name	Type	Description	Example
<code>crawler.useragent</code>	Optional	Specifies the User Agent string that is provided by the crawler to the target website during crawling.	<code>My-Agent-1.0</code>
<code>crawler.threaded</code>	Optional	Turns on multithreaded retrieval of the web site content per DataNode. This can improve speed on multi-core DataNodes.	<code>false</code>
<code>crawler.exclude.extensions.list</code>	Optional	A comma-separated list of file type extensions to exclude.	<code>.pdf,.png</code>
<code>crawler.includefilters</code>	Optional	A regular expression for the acceptable files to crawl.	<code>.*myserver.*/stuff/.*</code>
<code>crawler.excludefilters</code>	Optional	A regular expression for the files to exclude from the crawl.	<code>.* /badpath/.*</code>
<code>crawler.maxdepth</code>	Optional	Don't crawl through more links than this when crawling a website	<code>3</code>
<code>crawler.maxiterations</code>	Optional	Stop crawling after this many results.	<code>2000</code>
<code>crawler.user</code>	Optional	Try to use this user identity to access user/password protected pages.	<code>frank</code>
<code>crawler.password</code>	Optional	Try to use this password to access protected pages	<code>mypassw0rd23</code>

### Example



## Twitter crawler

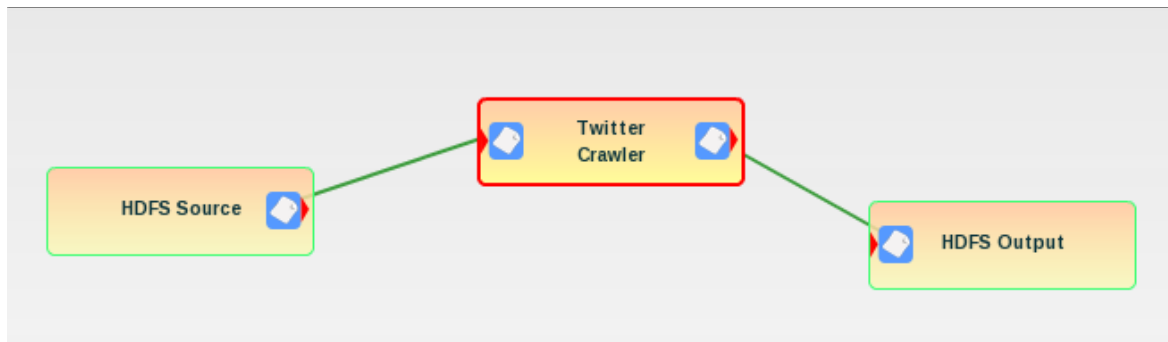
### Description

For each line of an input file, the Twitter Crawler will query Twitter using the input line as the query string. For instance, a line that reads “Dell XPS” will query the Twitter search API to acquire up to 1000 Tweets that mention Dell XPS. Required input parameters include Twitter API access tokens.

### Parameters

Name	Type	Description	Example
twitter.consumer.key	Required	The Twitter Consumer Key	lQCJtZaePvCMUy4heNnCk
twitter.consumer.secret	Required	The Twitter Consumer Secret	erSGv6pQMr0XanqETcts0y...
twitter.access.token	Required	Twitter Access Token	2772882-HuDpGGALafHjsLehOAaG...
twitter.access.token.secret	Required	Twitter Access Token Secret	Bzhk4mbKxyWAsDs2gbA...

### Example



## Facebook crawler

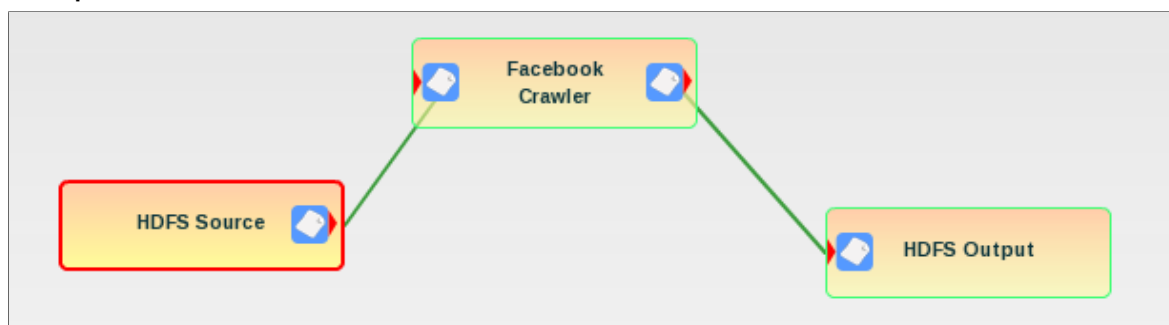
### Description

For each line of an input file, the Facebook Crawler will query Facebook using the input line as the query string. For instance, a line that reads “Dell XPS” will query the Facebook Graph API to acquire up to 1000 public Facebook posts that mention Dell XPS. Required input parameters include Facebook Graph API access tokens.

### Parameters

Name	Type	Description	Example
facebook.url	Required	The URL to the Facebook Graph API search function	<a href="https://graph.facebook.com...">https://graph.facebook.com...</a>
facebook.query.limit	Required	The maximum number of posts to retrieve	1000
facebook.app.id	Required	Facebook application ID	myappid
facebook.app.secret	Required	Facebook Access Token Secret	QAA44mbKxyWAsDs2gbA...

## Example



## Search

The Search Processing Elements interface to search engines for indexing of content and metadata.

### Solr Indexer

#### Description

The Solr Indexer Processing Element is a self-contained combination of the Document Converter PE and the Solr Search Engine PE for ease of use and deployment. The Solr Indexer consumes a document or directory of documents in PDF, .docx, .xls, HTML, etc., formats, and then forwards the Solr XML version to a Solr instance. Note that document conversion can require significant memory on the data nodes for larger documents, so an increase in memory is often required.

#### Parameters

Name	Type	Description	Example
zettavox.solrhost	Required	The URL to the Solr instance update function.	http://localhost:9100/solr/update
mapred.child.java.opts	Recommended	Change MapReduce child VM parameters	-Xmx1024m

## Example



### Multidocument Solr Indexer

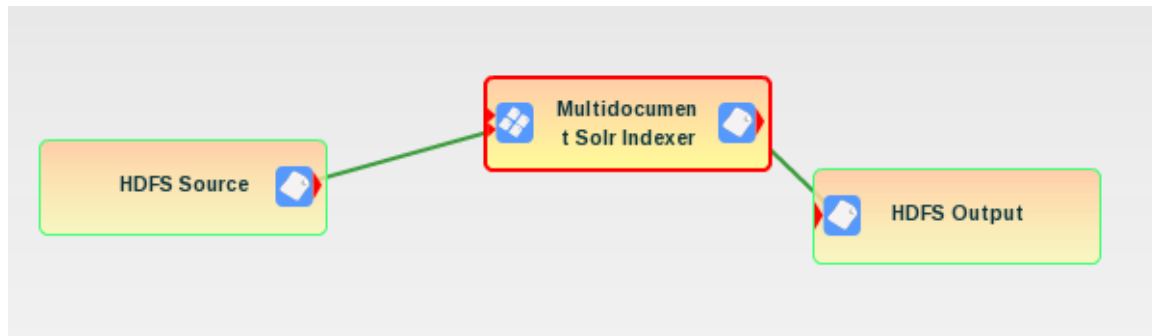
#### Description

The Solr Indexer Processing Element is a self-contained combination of the Document Converter PE and the Solr Search Engine PE for ease of use and deployment. The Solr Indexer consumes a Multidocument Collection containing document content in PDF, .docx, .xls, HTML, etc. formats, and then forwards the Solr XML version to a Solr instance. Note that document conversion can require significant memory on the data nodes for larger documents, so an increase in memory is often required.

## Parameters

Name	Type	Description	Example
zettavox.solrhost	Required	The URL to the Solr instance update function.	<a href="http://localhost:9100/solr/update">http://localhost:9100/solr/update</a>
mapred.child.java.opts	Recommended	Change MapReduce child VM parameters	-Xmx1024m

## Example



## Solr search engine

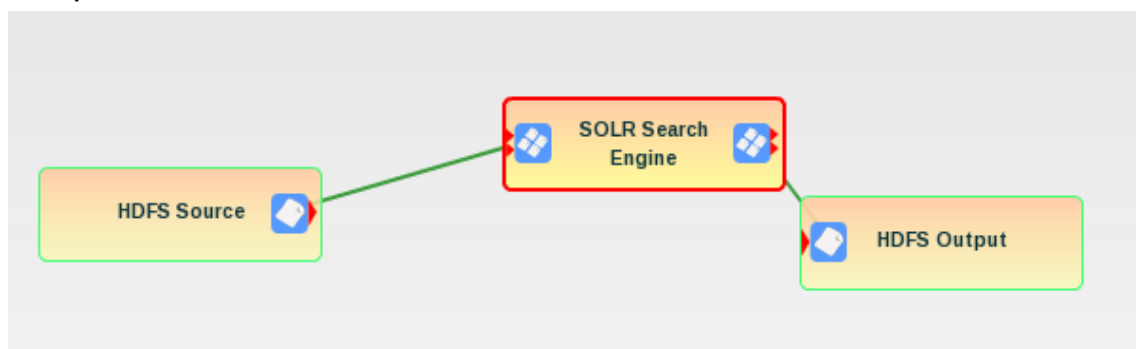
### Description

The Solr Search Engine Processing Element forwards XML documents in Solr XML format to a Solr instance. The PE consumes a Multidocument Collection containing document content in Solr XML format generated by tools such as the XML English Extractor and forwards the content to a Solr instance. The output is the same as the input for additional chaining of jobs.

## Parameters

Name	Type	Description	Example
zettavox.solrhost	Required	The URL to the Solr instance update function.	<a href="http://localhost:9100/solr/update">http://localhost:9100/solr/update</a>

## Example



## Filters

Filters provide the ability to transform data by limiting it or rearranging it in some way.

## Document converter

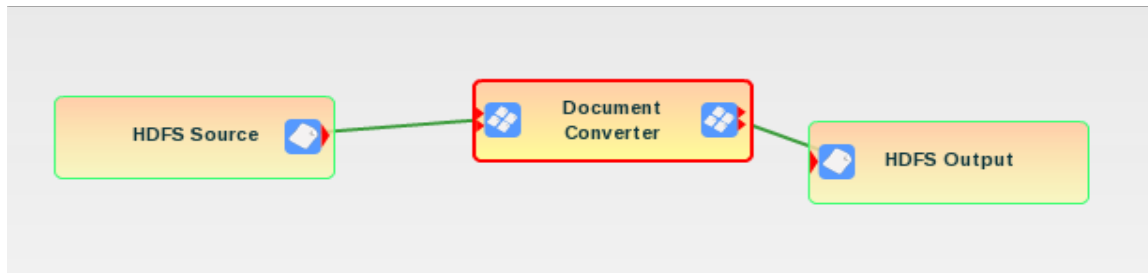
### Description

The Document Converter Processing Element converts documents from their original format into HTML for further processing. The PE consumes a Multidocument Collection containing document content in the original format (PDF, .docx, etc.). The output is the HTML version of the extracted content.

### Parameters

None

### Example



## Line filter

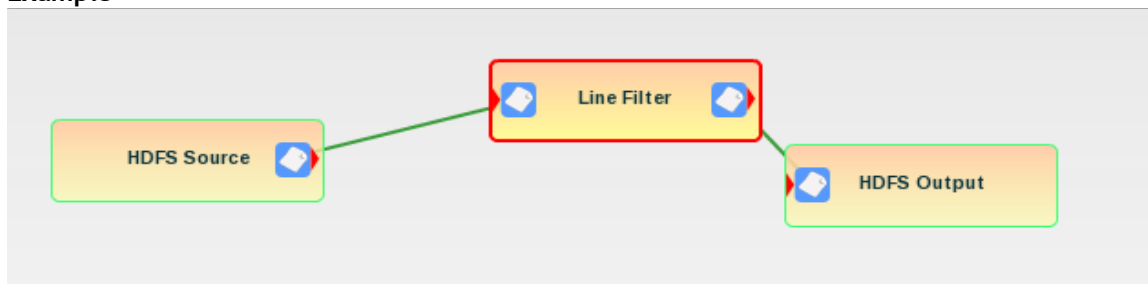
### Description

The Line Filter Processing Element is used to filter columns in tabular data. The Line Filter assumes that the input content is multiple lines in a text file (or directory of text files) and that the lines are separated by whitespace or other consistent patterns. The PE generates a single output file that contains the columns requested from the original content.

### Parameters

Name	Type	Description	Example
linesplitter.splits	Required	The columns requested from the original content, starting with column 0	1,4,7
linesplitter.splitkey	Required	The separator regular expression for the input data	\sK\s
linesplitter.separator	Required	The output separator for the generated data	SEP\t

### Example



## Row to XML

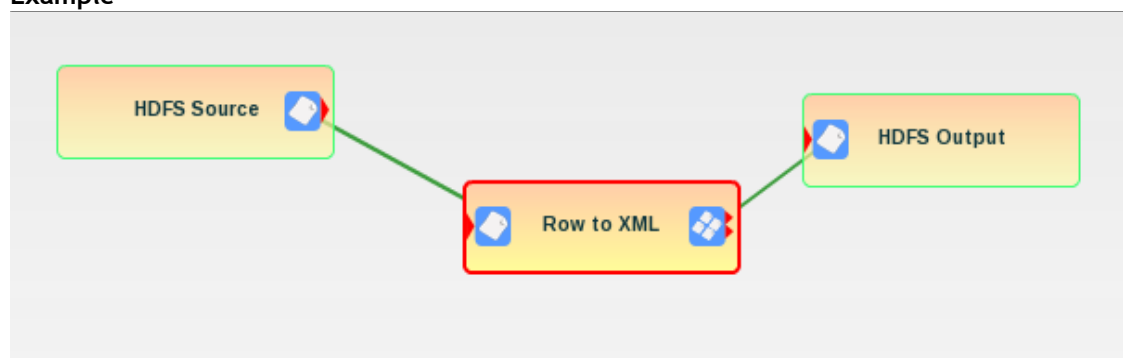
### Description

The Row to XML Processing Element is used to convert rows of tabular data into XML format suitable for consumption by Solr. The Row to XML PE assumes that the input content is multiple lines in a text file (or directory of text files) and that the lines are separated by whitespace or other consistent patterns. The PE then uses a simple schema to convert the columns into an XML output rendering. If there is a mismatch between the number of columns present in the data and the schema element count, the offending line will be skipped.

### Parameters

Name	Type	Description	Example
linesplitter.splitkey	Required	The separator regular expression for the input data	\sK\s
row2solr.schema	Required	Comma-separated list of terms that will be applied to	name,organization,location,salary

### Example



## Regular expression filter

### Description

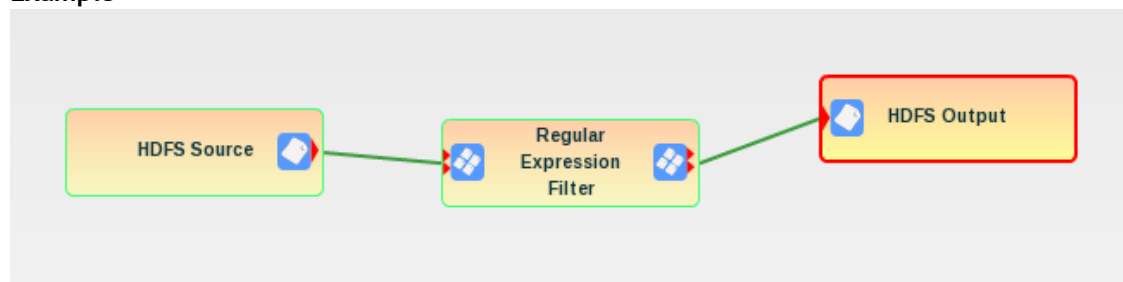
The Regular Expression Processing Element filters content in a Multidocument Collection based on a regular expression. This PE is useful for Multidocument Collections that contain HTML or text content, though it may have applications to other areas. The PE will create a new Multidocument Collection that contains the content that matches the provided regex. For instance, if the regular expression is “.\*2195”, the content of the resulting Collection will contain all of the strings that end in the number 2195. Note that each occurrence within the input content will be listed within the result.

### Parameters

Name	Type	Description	Example
mapred.mapper.regex	Required	The regular expression for extraction	.*2195



## Example



## Directory to Multidocument Collection

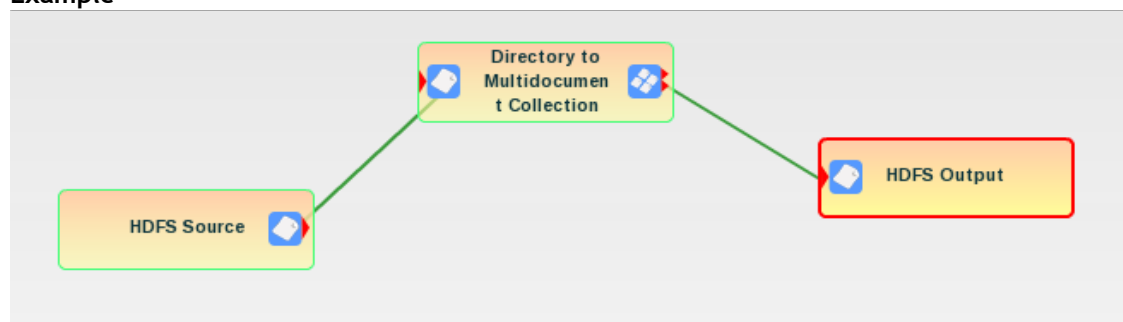
### Description

The Directory to Multidocument Collection Processing Element converts a directory of content into a Multidocument Collection. The keys in the collection will be the URIs for the input files (HDFS or FS) and the content will be the content of the files.

### Parameters

None

## Example



## Tabular data to sequence file

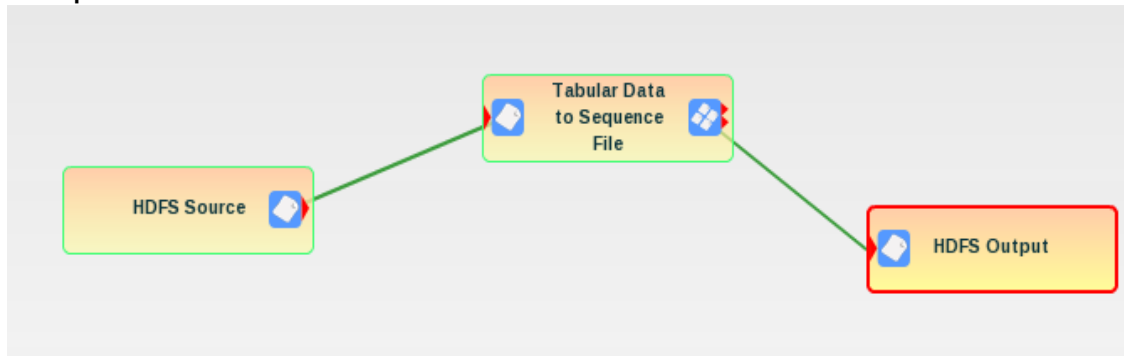
### Description

The Tabular Data to Sequence File Processing Element converts a table of content in a file into a Multidocument Collection or SequenceFile. The keys of the SequenceFile are taken from one of the columns in the tabular data. Also, a data label can be extracted from the columns in the data and it will be appended to the key.

### Parameters

Name	Type	Description	Example
tabular.data.id.position	Optional	The column id position for the key	2
mahout.naivebayes.data.label.position	Required	The column id position for Mahout labels	1
columns.to.exclude	Optional	Comma-delimited list of columns to exclude	3,5,7
delimiter	Required	Delimiter for breaking rows into columns	\\s\\t

## Example



## Pig script

### Description

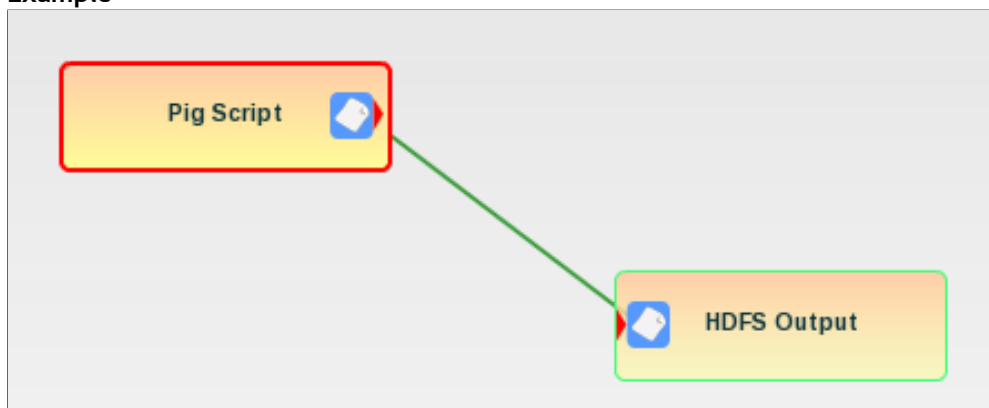
The Pig Script PE allows for the execution of arbitrary Pig scripts written in Pig Latin. The Pig scripts can be run from either HDFS or the local file system on the SBDA server. Since the Pig scripts are relatively small in size, it is recommended to keep them on the local file system. There are some limitations:

- The PE must be the first element of the workflow
- The inputs used in the LOAD command must be available on the file system before execution starts
- When passing output to the next PE in the workflow the fully qualified path needs to be specified in the STORE command
- The PE will not automatically delete the output directory, as with most SBDA PEs
- The PE does not support external UDF libraries

### Parameters

Name	Type	Description	Example
scriptFile	Required	The path on FS or HDFS for a Pig Script file	<a href="#">hdfs://myhost/scripts/s.pig</a>
path	Required	The output path of the Pig store command internally to the script so that the output sink or next PE can receive the output of the script.	<a href="#">hdfs://myhost/myoutput</a>

## Example



**NOTE:** Pig has a bug with how it manages temporary files. When run from within a web application, the temporary files are not cleaned up until the web application shuts down. The expected behavior is that the temporary files are cleaned up after the Pig job completes. A ticket has been entered with the Pig

development team and a fix is pending. In the meantime, the work around is to periodically restart the SBDA web application with the command "bin/Statistica.sh restart".

## Pig GroupBy

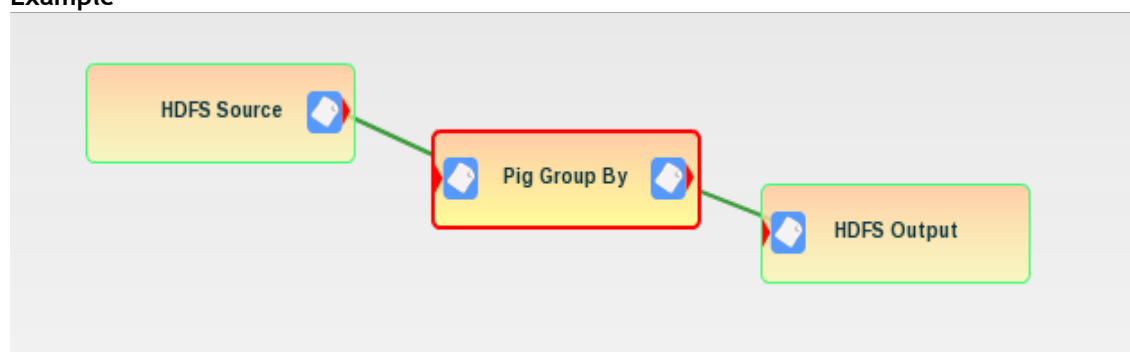
### Description

The Pig Group By is a convenience function that performs a group by operation for aggregation using an embedded Pig script. The output of the Group By PE is a Pig Bag with the group key and a list of items in that grouping. A schema must be supplied that expresses the columns in the input content, and the columns to be used in the grouping must be supplied. Also, the input and output delimiters can be specified, if desired.

### Parameters

Name	Type	Description	Example
schema	Required	A Pig Latin schema description for the input	f1:int,f2:int,f3:int
groupBy	Required	The column(s) to use for the grouping operation. More than one column can be used, but will need to be separated with a comma.	f1,f2
inputDelimiter	Optional	The delimiter character to use when parsing the input. The default is comma.	\t
outputDelimiter	Optional	The delimiter character to use for the resulting output. The default is comma.	\t

### Example



**Note:** Pig has a bug with how it manages temporary files. When run from within a web application, the temporary files are not cleaned up until the web application shuts down. The expected behavior is that the temporary files are cleaned up after the Pig job completes. A ticket has been entered with the Pig development team and a fix is pending. In the meantime, the work around is to periodically restart the SBDA web application with the command "bin/Statistica.sh restart".

## Pig join

### Description

The Pig Join PE is a convenience function that performs a join operation using an embedded Pig script. It will combine two input sources into an output that has been joined upon one or more common columns. Although the PE will be in a workflow with connectors, it will also be necessary to specify the input parameters in the PE properties in order to ensure that the schema is correctly matched with the appropriate input source. In addition, it will be necessary to provide a schema and the join column(s) for each input source. The join column(s) must be common to both input sources. If multiple join columns are used, they will need to be enclosed in parenthesis and separated by commas. Also, the input and output delimiters can be changed from the default of commas if needed. This PE has the following limitations:

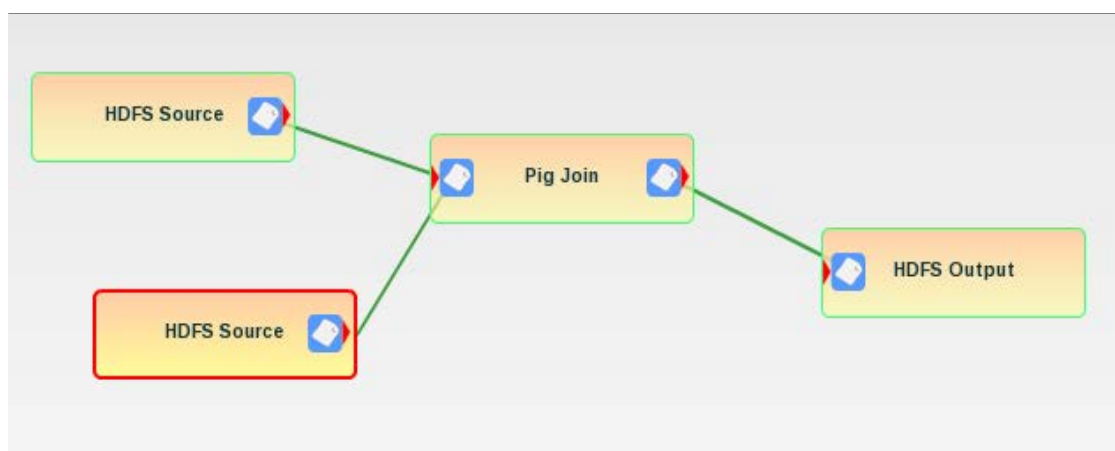
- It will only support two input sources.

- It will only perform an INNER join. However, this could be worked around by using the PE Script PE to use Pig Latin to produce a different kind of join.

#### Parameters

Name	Type	Description	Example
<b>inputPath1</b>	Required	The HDFS input path for the first data set	hdfs://myhost/myinput1
<b>inputDelimiter1</b>	Optional	The delimiter for the first input data set, defaults to comma	\t
<b>schema1</b>	Required	A Pig Latin schema description for the first input	f1:int,f2:int,f3:int
<b>joinCol1</b>	Required	The column to be used for the join in the first input	f1 or (f1,f2)
<b>inputPath2</b>	Required	The HDFS input path for the second data set	hdfs://myhost/myinput2
<b>inputDelimiter2</b>	Optional	The delimiter for the second input data set, defaults to comma	\t
<b>schema2</b>	Required	A Pig Latin schema description for the second input	f1:int,f2:int,f3:int
<b>joinCol2</b>	Required	The column to be used for the join in the second input	f1 or (f1,f2)
<b>outputDelimiter</b>	Optional	Reorganize the joins into a tabular form with this separator, defaults to comma	\t

#### Example



**Note:** Pig has a bug with how it manages temporary files. When run from within a web application, the temporary files are not cleaned up until the web application shuts down. The expected behavior is that the temporary files are cleaned up after the Pig job completes. A ticket has been entered with the Pig development team and a fix is pending. In the meantime, the work around is to periodically restart the SBDA web application with the command "bin/Statistica.sh restart".

## Analysis

Analysis components are used for counting and other analytical tasks.

## Word counter

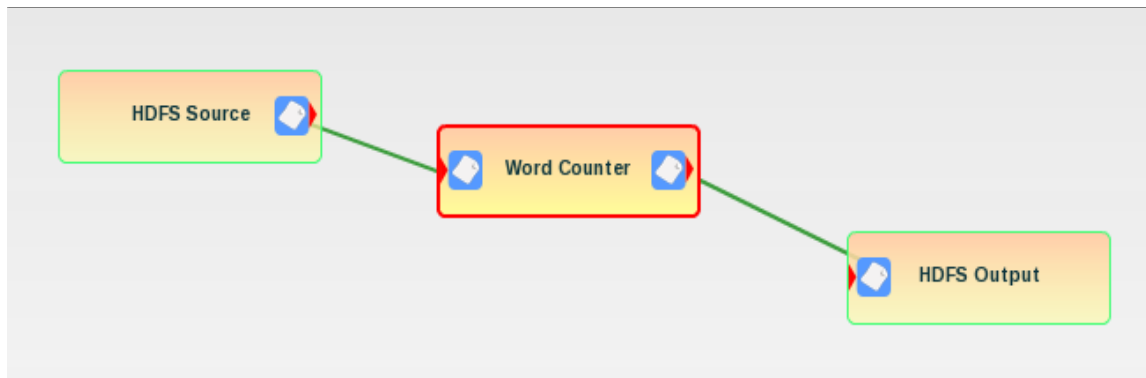
### Description

The Word Counter counts words by breaking up text or other content one line at a time and running a simple tokenizer over the lines. The tokenizer breaks on whitespace and does not understand complex punctuation. The input is a text file or a directory of text files. The output is a single file containing the words and their counts separated by a tab character.

### Parameters

None

### Example



## Multidocument word counter

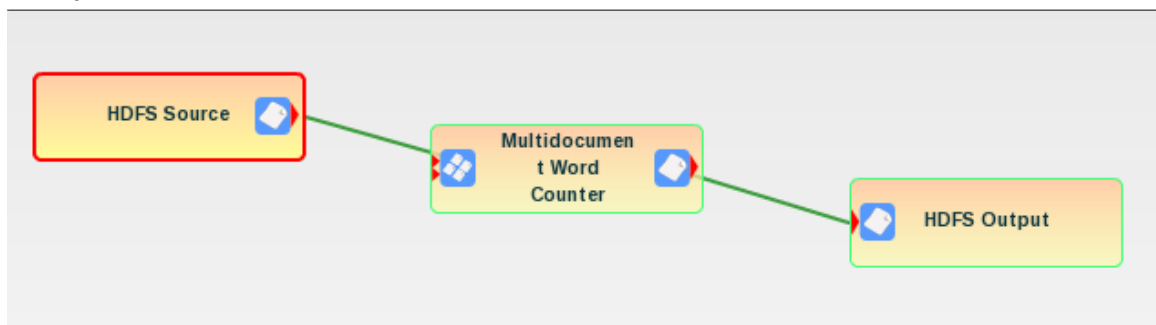
### Description

The Multidocument Word Counter counts words by breaking up text or other content one line at a time and running a simple tokenizer over the lines. The tokenizer breaks on whitespace and does not understand complex punctuation. The input is a Multidocument Collection of text files. The output is a single file containing the words and their counts separated by a tab character.

### Parameters

None

### Example



## Streaming

Streaming components make use of the Hadoop streaming capability to execute MapReduce systems that run out-of-process.

## R Streaming

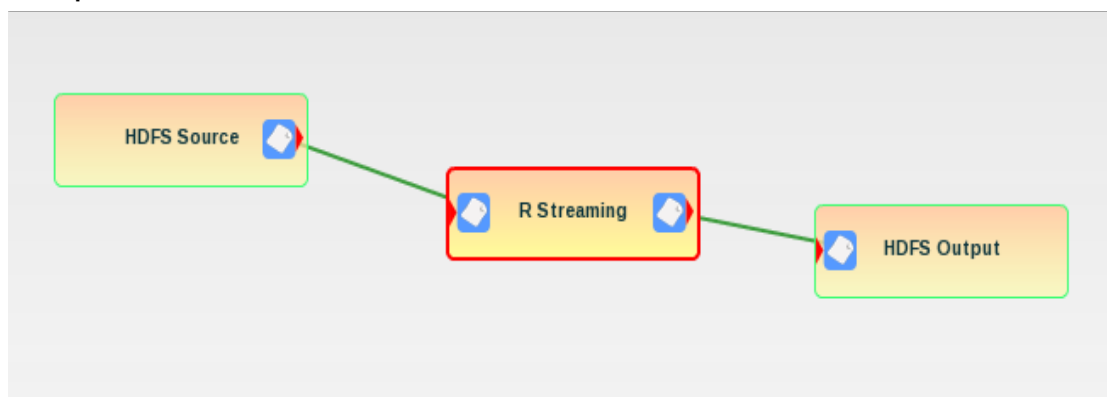
### Description

The R Streaming PE executes an R executable that must be present on each data node.

### Parameters

Name	Type	Description	Example
mapper	Required	The mapper executable	/tmp/R/stock_day_avg.R
combiner	Required	The combiner executable	TBD
reducer	Required	The reducer executable	/tmp/R/stock_cma.R
mapperfile	Required	The location of the mapper executable available locally on the compute nodes	/tmp/R/stock_day_avg.R
combinerfile	Required	The location of the combiner executable available locally on the compute nodes	Leave as blank
reducerfile	Required	The location of the reducer executable available locally on the compute nodes	/tmp/R/stock_cma.R
inputformat	Required	The class which should return key/value pairs of text class	Leave as blank
outputformat	Required	The class which should take key/value pairs of text class	Leave as blank
partitioner	Required	The class that determines which reduce a key is sent to	Leave as blank
inputreader	Required	The input record reader spec	Leave as blank
cmdenv	Required	The environment variable to streaming commands	Leave as blank
verbose	Required	Whether to print verbose output	FALSE
io	Required	The identifier	Leave as blank
numReduceTasks	Required	The number of reducers	Leave as blank
mapdebug	Required	The script to call when map task fails	Leave as blank
reducedebg	Required	The script to call when reduce task fails	Leave as blank
lazyOutput	Required	Whether to create outputs lazily	FALSE

### Example



# Data sinks

Data Sink elements are outputs from analytics workflows using Processing Elements and Data Sources.

## Filesink

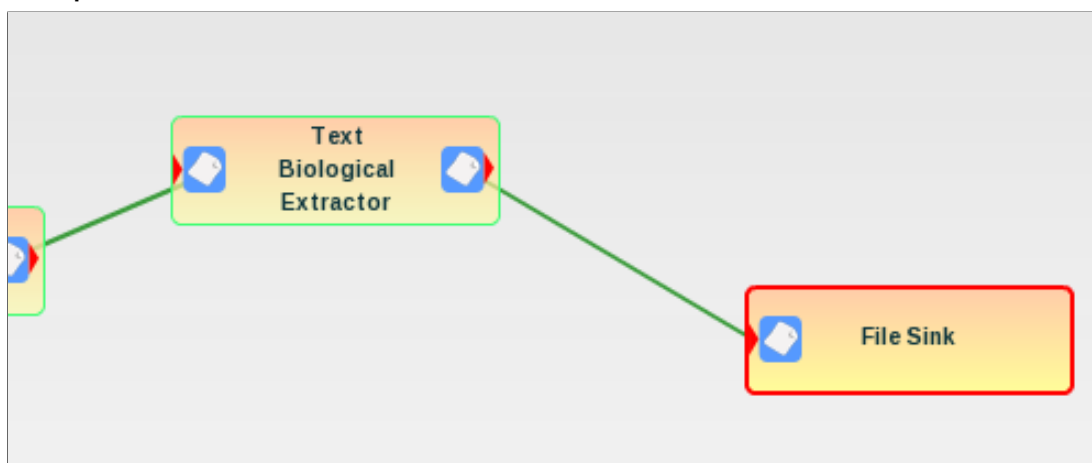
### Description

The File Sink receives data from files on the computer's file system. The path must be specified using a file system URI. Also, for a cluster environment, DataNodes will try to put data to the same path on each DataNode. Generally, that means that the file must be mounted using NSF on each computer.

### Parameters

Name	Type	Description	Example
fspath	Required	URL of file	<a href="file:///home/Statistica/testfile.txt">file:///home/Statistica/testfile.txt</a>

### Example



## HDFS sink

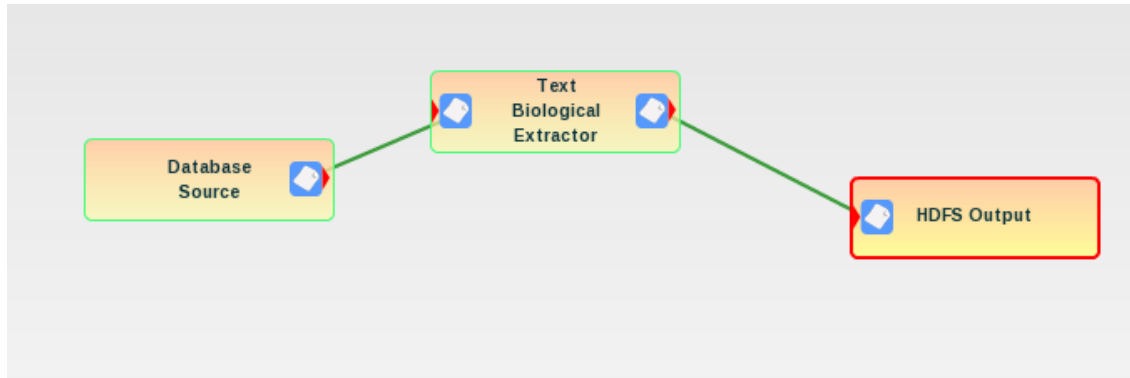
### Description

The HDFS Sink (Hadoop Distributed File System Sink) outputs to directories the Hadoop Distributed File System. For most Hadoop processes, the output files will be part-r-XXXXX files in the directory. If the directory does not exist it will be created.

### Parameters

Name	Type	Description	Example
fspath	Required	URL of output directory	<a href="file:///home/Statistica/test">file:///home/Statistica/test</a>

## Example



## Amazon S3 sink

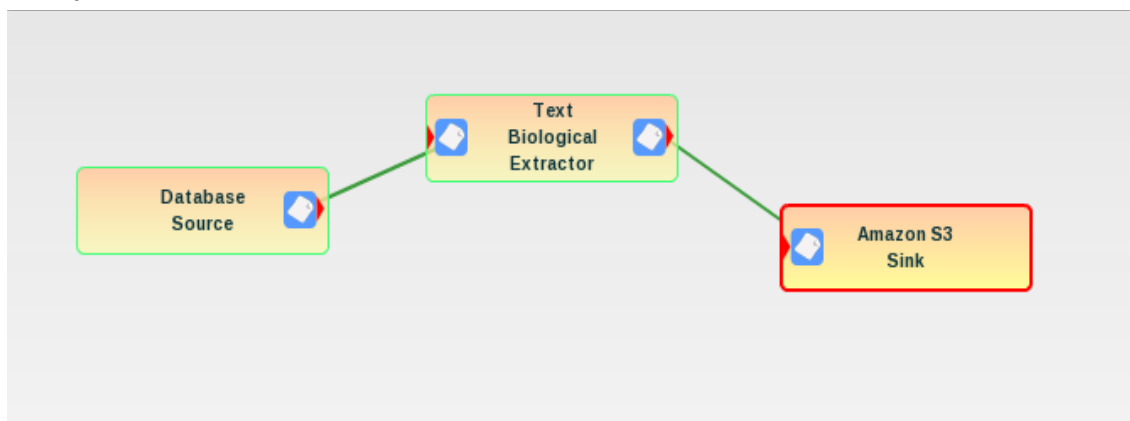
### Description

The Amazon S3 Source pulls data from an Amazon S3 Bucket. An Amazon S3 Bucket is identified by a unique bucket string and the files to be processed within that bucket must be identified. In addition, the user's access key and secret access key are needed to use the data in the bucket.

### Parameters

Name	Type	Description	Example
<b>aws.accessKey</b>	Required	The user access key for Amazon S3 services	AKIAIOSFODNN7EXAMPLE
<b>aws.secretAccessKey</b>	Required	The user secret access key for S3 services	wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKE
<b>aws.endpoint</b>	Required	The URL for accessing S3 services	s3.amazonaws.com
<b>aws.bucket</b>	Required	The S3 bucket identifier	Statistica-bucket-1
<b>aws.path</b>	Required	The path of the files with the bucket	Myfile1.txt

## Example





## Appendix B: Manual configuration of data nodes for GATE

This appendix provides the details of how to manually configure the Data Nodes to allow for the GATE library to be used with the Natural Language Processing related Processing Elements. Essentially, the GATE directory, from the server where the Statistica Big Data Analytics was installed will need to be copied onto each of the Data Nodes, while ensuring that the user, directory structure, and groups permissions are matched on each Data Node. Ordinarily this would have been completed as part of the configuration step as the bin/configure.sh script gets executed in the installation process. However, if for some reason this was not completed correctly, the manual process can be applied.

### Configuring an SBDA server

Complete these steps on the server where the Statistica Big Data Analytics was installed.

1. Create a tar file
  - a. Log into the server where the Statistica Big Data Analytics was installed as the SBDA user (i.e.: Statistica).
  - b. Change to the directory above the GATE directory:
 

```
[Statistica]$ cd /home/Statistica/Statistica-13.1
```
  - c. Create a tar file of the entire GATE directory
 

```
[Statistica]$ tar -cf GATE.tar GATE
```

2. Make the generated tar file available to all of the Data Nodes

There are many possible options, such as using SCP, using rsync, using an NFS mount, etc. In this case, the simplest and most direct approach of using SCP will be used.

From the directory where the tar file was generated, enter the following command with each of the Data Nodes as the destination:

```
[Statistica]$ scp GATE.tar root@<data_node>:/tmp/GATE.tar
```

This places the tar file in the /tmp directory of each of the Data Nodes, but it will be owned by the root user on the remote server.

### Configuring data nodes

Repeat these steps on each Data Node.

1. Create a user for SBDA to run as

It will be necessary to create a user account for the Statistica Big Data Analytics to use. The user name and user identifier should match with the values used on the SBDA server. It will be necessary to be the root user, or execute these commands with a user that has sudo privileges. The suggested command is:

```
[root]# useradd -U -G mapred,hdfs,hadoop -m -u 500 Statistica
```

2. Give your user a password to be able to login:

```
[root]# passwd Statistica
```

By default this will create the home directory of /home/Statistica. **Note:** The home directory should match with what was done when the matching user was created on the SBDA server.

### 3. Ensure the user is in the CDH groups

The SBDA user needs to be a member of the CDH groups (hadoop, hdfs and mapred). If the user had already been created or for some reason the group relationship was not established, it will need to be done here. First verify what the current group assignments are for the SBDA user with the command:

```
[root]# id -Gn Statistica
```

The output of this command should be something like **Statistica hdfs hadoop mapred**. If it is, you can move on to the next step. Otherwise, you will need to establish the group relationships with the command:

```
[root]# usermod -G hadoop,hdfs,mapred -a Statistica
```

Repeat the id command to confirm.

### 4. Add the CDH users to the SBDA user group

The group relationship will also need to allow for the CDH users (hadoop, hdfs and mapred) to be members in the SBDA user group.

```
[root]# usermod -G Statistica -a hadoop
```

```
[root]# usermod -G Statistica -a hdfs
```

```
[root]# usermod -G Statistica -a mapred
```

### 5. Grant group access to the SBDA user home directory

Here you will need to change the permissions of the SBDA user's home directory to allow for the members in the group to be able to access the files in there. This is needed to allow the CDH users to access the GATE directory.

```
[root]# chmod 770 /home/Statistica
```

### 6. Extract the GATE tar file

Finally, you will need to change the ownership of the GATE tar file to allow for the SBDA user to manipulate it, become the SBDA user, and extract the GATE files into the appropriate place.

```
[root]# chown Statistica:Statistica 770 /tmp/GATE.tar
```

```
[root]# su - Statistica
```

```
[Statistica]$ cd Statistica-13.1
```

```
[Statistica]$ tar -xf GATE.tar GATE
```

After completing this, the GATE directory should be a subdirectory below Statistica-13.1 and be owned by the SBDA user and have permissions of rwxrwxr-x.

# About Dell

Dell listens to customers and delivers worldwide innovative technology, business solutions and services they trust and value. For more information, visit [www.software.dell.com](http://www.software.dell.com).

## Contacting Dell

Technical Support:

[Online Support](#)

Product Questions and Sales:

(800) 306 - 9329

Email:

[info@software.dell.com](mailto:info@software.dell.com)

## Contacting Support

Support is available to customers who have a trial version or who have purchased Dell software and have a valid maintenance contract. The Support Portal [www.software.dell.com/support](http://www.software.dell.com/support) is the definitive resource for technical support with self-help capabilities so you can solve problems quickly and independently 24 hours a day, 365 days a year. The portal also provides direct access to our support engineers through an online service request facility. From one central location, you will find everything you need - support offerings, policies and procedures, contact information, as well as:

- Create, update, and manage Service Requests (cases)
- Knowledge Base
- Product notifications
- Software downloads<sup>1</sup>
- How-to videos
- Community discussions
- Chat option

<sup>1</sup> For trial users please use the [Trial Downloads](#) to get the latest generally available version of the software.

Quest Software is now Dell Software.