HDFS Connector

Plugin module	Data provider name	Automatic metadata detection
HADOOP	HDFS	No

The HDFS connector allows external tables to fetch data from a Hadoop filesystem using libhdfs.

Prerequisites

- Kognitio version 7.9 or later
- A working libhdfs (64-bit recommended), and Java Runtime Environment of the same bitness, on all DB nodes.
- A working Hadoop installation (at least the client libraries, particularly the class files) on all DB nodes
- Access to files on your Hadoop filesystem from all DB nodes

Examples

Load the plugin

```
create module hadoop;
alter module hadoop set mode active;
```

Create a connector

```
create connector myhdfscon source hdfs
target 'namenode address:port, user hdfsuser';
```

Create an external table

```
create external table customer2013 (
        id int,
        name varchar(100),
        company varchar(100),
        address varchar(400)
) from myhdfscon
target 'file /user/fred/customers/2013/*.csv';
```

Attributes

Attribute	Type	Default	Description
namenode	string	default:0	The address or hostname of the Hadoop cluster's namenode, with an optional port number. e.g. 172.30.21.1:9000 or hdfs://172.30.21.1:9000.
user	string	root	Username to use when connecting to HDFS.
bitness	integer	64 *(v8.2)* 32 *(v8.1)*	Specify whether the libhdfs library and JRE are 32-bit or 64-bit.
file	string	none	The HDFS file name to load. This can also be a wildcard, e.g. /customers/2013/*.csv.
wholefiles	boolean	false	Do not assign different blocks of a file to separate threads - make one thread read a whole file.
list	string		List files in this directory or which match this pattern. If given, a directory listing is returned rather than the file contents, and any file attribute is ignored.
recursive	boolean	false	When listing, recursively descend into directories.
dironly	boolean	false	When listing, return information about the directory itself, not its contents.
allow_empty_dir	boolean	true	If false, then if a path matches only empty directories, it's an error. If true, it gives 0 rows.
allow_non_match	boolean	false	If true, specifying a path which matches nothing isn't an error, it gives you 0 rows.

Additional attributes can be used to specify how the input files are formatted; see the **Target String Format Attributes** reference sheet.

Module parameters

Example: `alter module hadoop set parameter java_home to '/usr/java/latest';`

Parameter name	Default	Description
libhdfs	libhdfs.so	The name of the libhdfs binary. This can be an absolute path, e.g. /usr/local/lib/libhdfs.so, or a relative path, in which case certain locations are searched (see the man page for dlopen(3)).
libjvm	libjvm.so	The name of the libjvm binary to use, if not libjvm.so.
java_home	/usr/lib/jvm/jre	The location of your JRE.
hadoop_client	/usr/bin/hadoop	The location of the Hadoop client executable. This is used to determine the correct classpath.

Notes

- Large files are divided into blocks and each thread parses a block. Newline characters are taken as record terminators. If the input files contain any newline characters inside fields (quoted CSV files may have this), you need to set the wholefiles attribute.
- It is common for files in HDFS to be compressed with gzip. Files whose names end with .gz are assumed to be gzip files, and are transparently decompressed by the connector. The connector behaves as if wholefiles is true for these files.