**Workflow of Sqoop and its benefits**

Apache Sqoop was created to efficiently transfer bulk data between Hadoop and external structured datastores, such as RDBMS and data warehouses, because databases are not easily accessible by Hadoop.

Sqoop is a bulk data transfer tool that allows easy import/export of data from structured datastores such as relational databases, enterprise data warehouses, and NoSQL systems. Using Sqoop, you can provision the data from an external system into HDFS, as well as populate tables in Hive and HBase. Similarly, Sqoop integrates with the workflow coordinator Apache Oozie (incubating), allowing you to schedule and automate import/export tasks. Sqoop uses a connector-based architecture which supports plugins that provide connectivity to additional external systems.



The sqoop action runs a Sqoop job.

The workflow job will wait until the Sqoop job completes before continuing to the next action.

To run the Sqoop job, you have to configure the sqoop action with the =job-tracker=, name-node and Sqoop command or arg elements as well as configuration.

A sqoop action can be configured to create or delete HDFS directories before starting the Sqoop job.

Sqoop configuration can be specified with a file, using the job-xml element, and inline, using the configuration elements.

Oozie EL expressions can be used in the inline configuration. Property values specified in the configuration element override values specified in the job-xml file.

Note that Hadoop mapred.job.tracker and fs.default.name properties must not be present in the inline configuration.

**Sqoop command**

The Sqoop command can be specified either using the command element or multiple arg elements.

When using the command element, Oozie will split the command on every space into multiple arguments.

When using the arg elements, Oozie will pass each argument value as an argument to Sqoop.

The arg variant should be used when there are spaces within a single argument.

Consult the Sqoop documentation for a complete list of valid Sqoop commands.

All the above elements can be parameterized (templatized) using EL expressions.

**Sqoop Action Counters**

The counters of the map-reduce job run by the Sqoop action are available to be used in the workflow via the [hadoop:counters() EL function](https://oozie.apache.org/docs/4.0.0/WorkflowFunctionalSpec.html#HadoopCountersEL) .

If the Sqoop action run an import all command, the hadoop:counters() EL will return the aggregated counters of all map-reduce jobs run by the Sqoop import all command.

**Sqoop Action Logging**

Sqoop action logs are redirected to the Oozie Launcher map-reduce job task STDOUT/STDERR that runs Sqoop.

From Oozie web-console, from the Sqoop action pop up using the 'Console URL' link, it is possible to navigate to the Oozie Launcher map-reduce job task logs via the Hadoop job-tracker web-console.

The logging level of the Sqoop action can set in the Sqoop action configuration using the property oozie.sqoop.log.level . The default value is INFO.

**Benefits of Sqoop**

Ease of Use - Sqoop lets connectors to be configured in one place, which can be managed by the admin role and run by the operator role. This centralized architecture helps in better deployment of Big Data analytics and solutions.

Ease of Extension - The connectors of Sqoop are not restricted to just the JDBC model. It has the competencies to extend and define its own vocabulary without having the need to mention a table name.

Security - The fact that Sqoop operates as server based application that secures access to external systems and does not allow code generation, makes its security to go by.