Offload Data Warehousing to Hadoop by using DataStage

See also a guided tour „Offload Data Warehousing to Hadoop by using DataStage”
Use IBM® InfoSphere® DataStage® to load Hadoop and use YARN to manage DataStage workloads in a Hadoop cluster (a registered IBM Cloud Id is needed!):
https://www.ibm.com/cloud/garage/dte/producttour/offload-data-warehousing-hadoop-using-datastage
See also a YouTube video which describes the details of this guided tour:
https://www.youtube.com/watch?v=QyWdzCeD6cU

Environment

This environment is a front-end Windows workstation that has the DataStage Designer version 11.5 installed. It will connect to a back-end Linux Server that is running the InformationServer Server components, and a configuration known as BigIntegrate. The DataStage Engine will run inside a remote Hadoop Cluster, and participate in the resource management of the Yarn service within Hadoop.

- Pre-configured, Autostarted software and services
- Allows you to see the configuration necessary to utilize the Hadoop cluster for running ETL processes

Tutorial

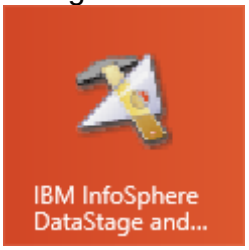
In this demo, you use DataStage to complete extract, transform, and load (ETL) data processing in a traditional enterprise data warehouse. You then offload the data and ETL processing into scalable, high-value Hadoop clusters and data lakes.
In this product tour, you will walk through the following tasks:
In this product tour, you get experience with the following features:

- Learn how to run DataStage traditional ETL jobs
- Configure DataStage to run inside Hadoop Clusters
- Examine execution logs to ensure configuration worked correctly

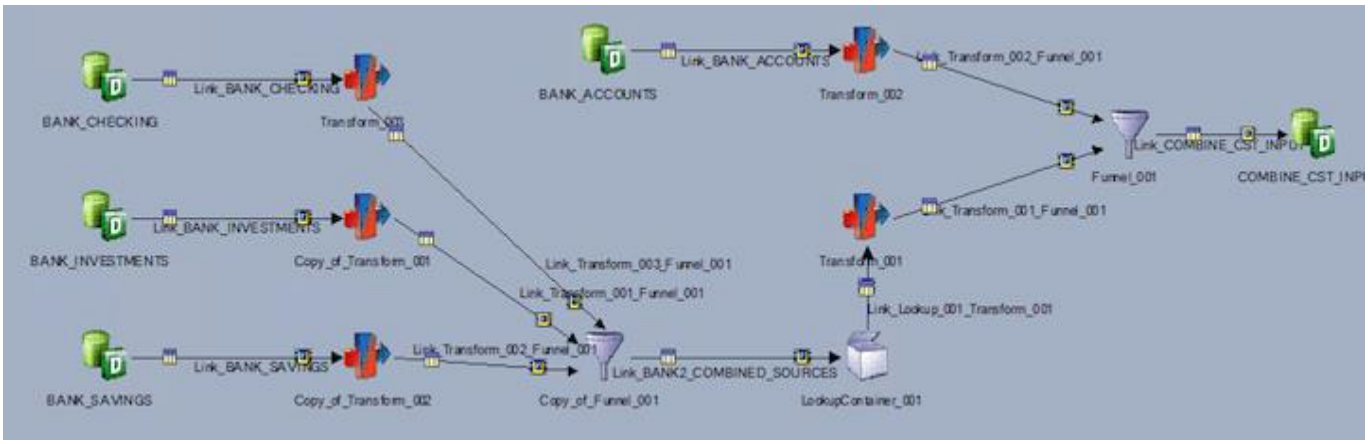
Follow the instructions in this pane to walk through the demo in the left pane.

Run a traditional Data Warehouse ETL job

First, you review a DataStage job that combines data from two lines of business into a table. You then run the job to create a combined data repository.
1. Click **Start demo now**
2. From the Start menu, click the IBM InfoSphere DataStage and QualityStage Designer icon.



The DS07\JK\BANK1\And\JK\BANK2\To\COMBINE\CST\INPUT job is displayed.



The job has these data sources:

* BANK1 ACCOUNTS data, which is labeled as `BANK_ACCOUNTS`.

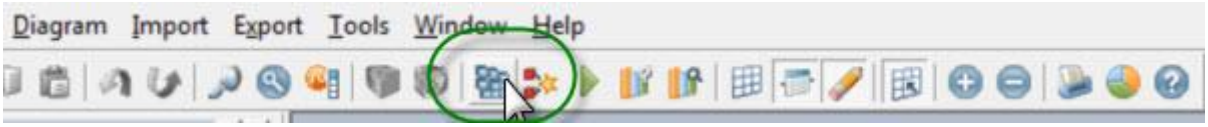
* BANK2 CHECKING customer data, which is labeled as `BANK_CHECKING`.

* BANK2 INVESTMENT customer data, which is labeled as `BANK_INVESTMENTS`.

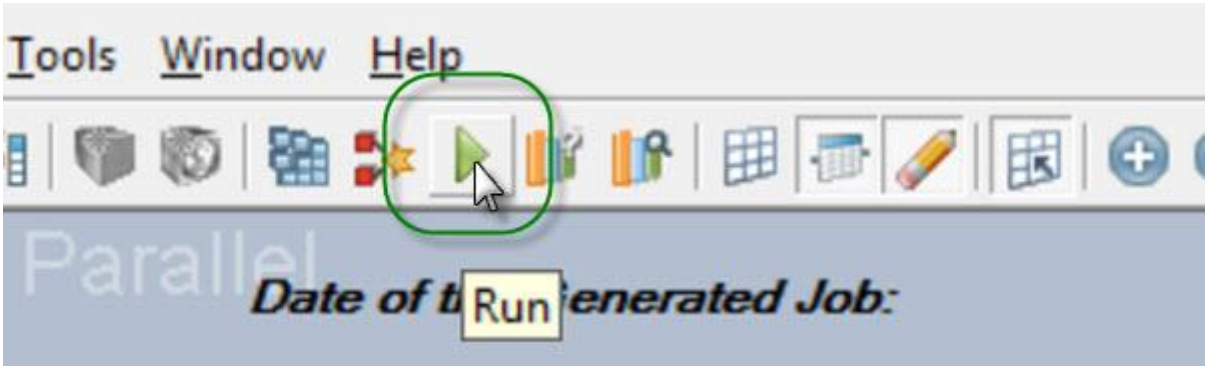
* BANK2 SAVINGS customer data, which is labeled as `BANK_SAVINGS`.

The job's data target is BANK1 and BANK2 combined customer data.

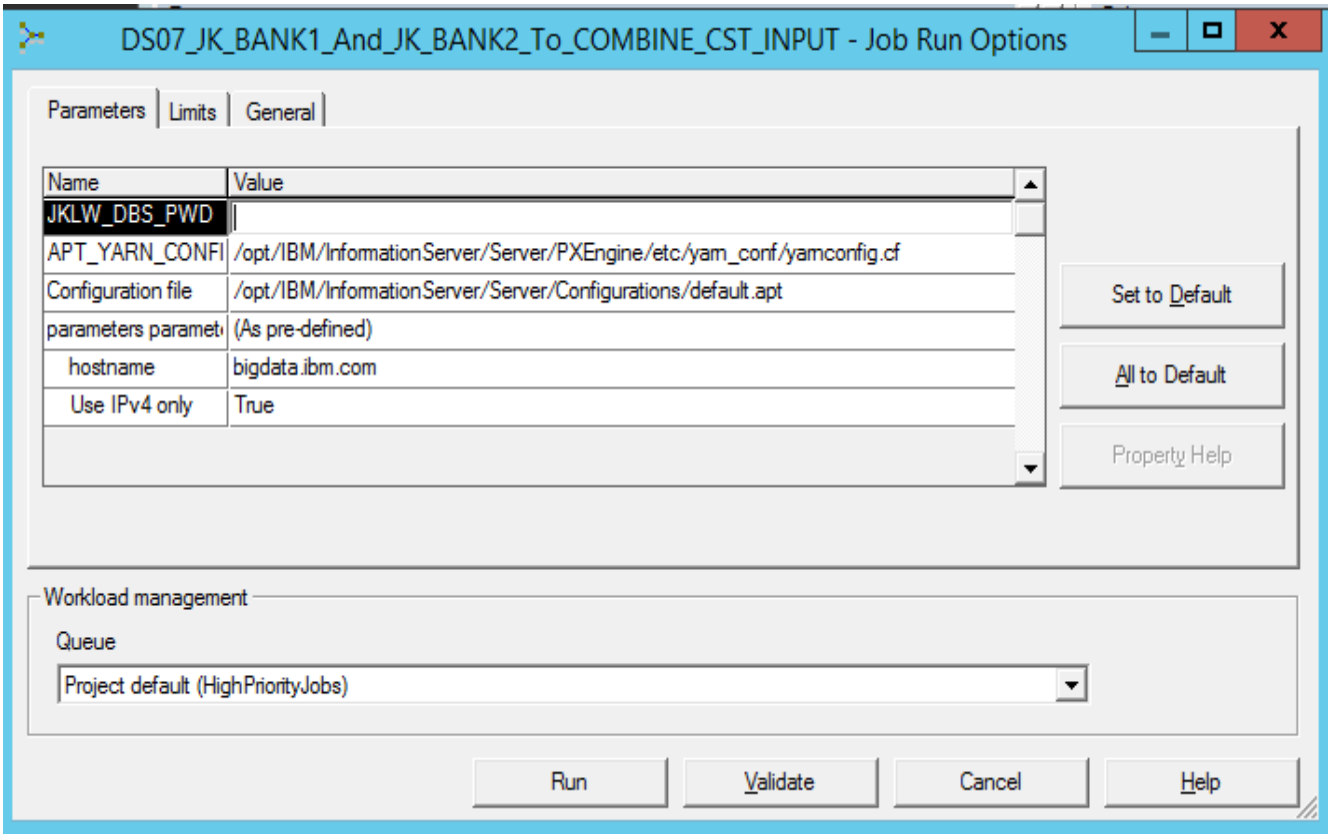
3. Compile the job by clicking the **Compile icon** on the toolbar.



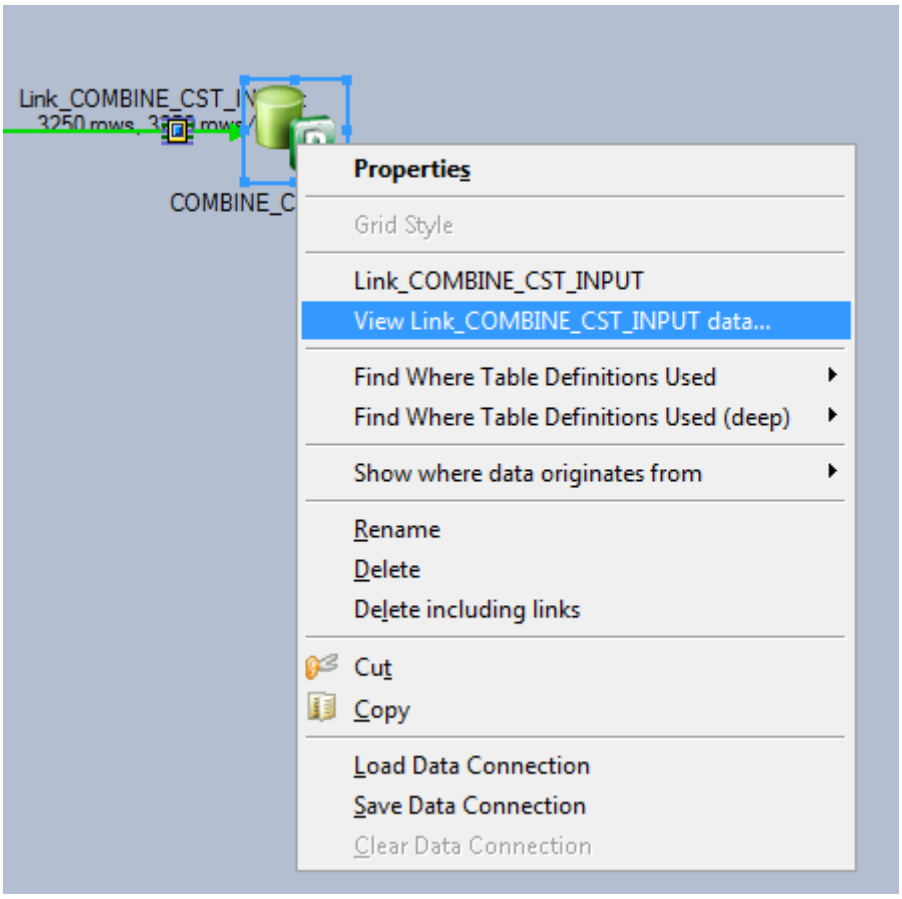
4. Run the job by clicking the **Run** icon on the toolbar.



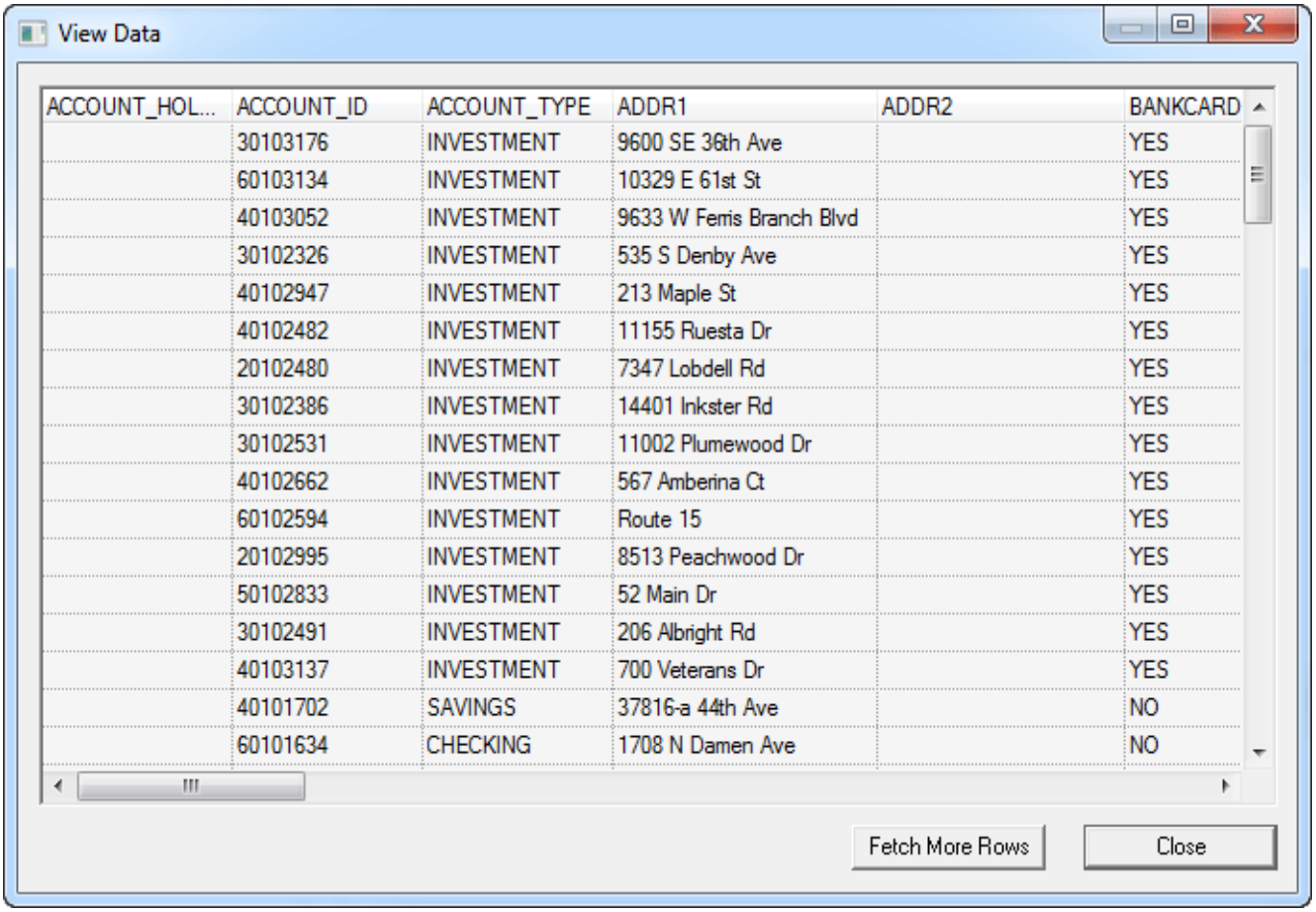
5. In the Job Run Options window, you can provide values for the runtime job parameters. Use the default values. For the `JKLW_DBS_PWD` parameter, type `inf0server`. Click **Run** to run the job.



6. After the job runs, verify the data from the output stage:
   a. Click the **COMBINE\CST\INPUT** output database connector and click View Link\COMBINE\CST\INPUT Data



The data that is returned represents a combined customer input set for JKBANK1 & JK\BANK2 line-of-business customers in a non-standardized and un-cleansed form.



b. Click **Close** to close the window.

Move data from Data Warehouse to Hadoop

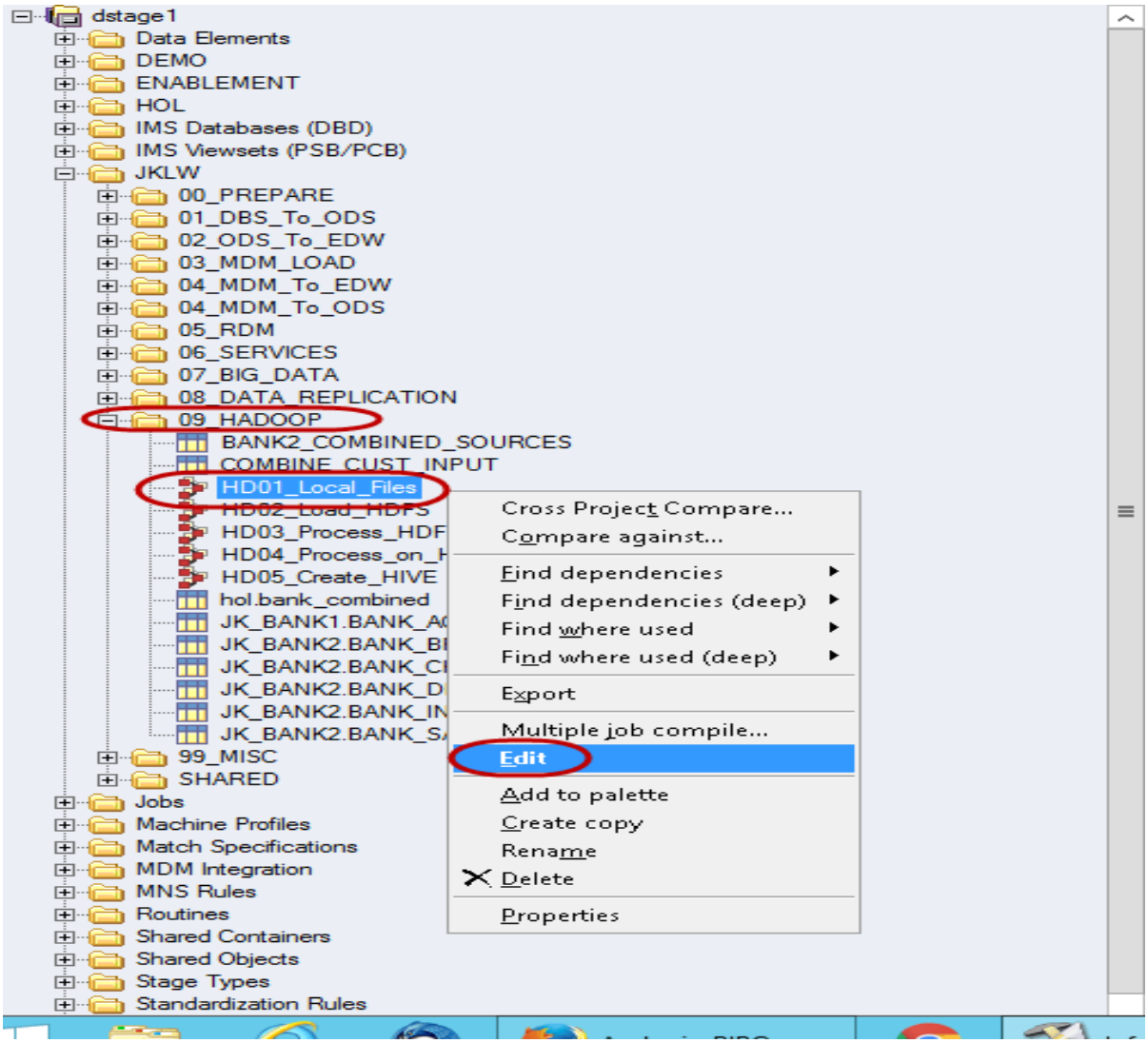
Explore how to use the scalability of the parallel engine and offload the processing from the enterprise data warehouse into a Hadoop cluster.

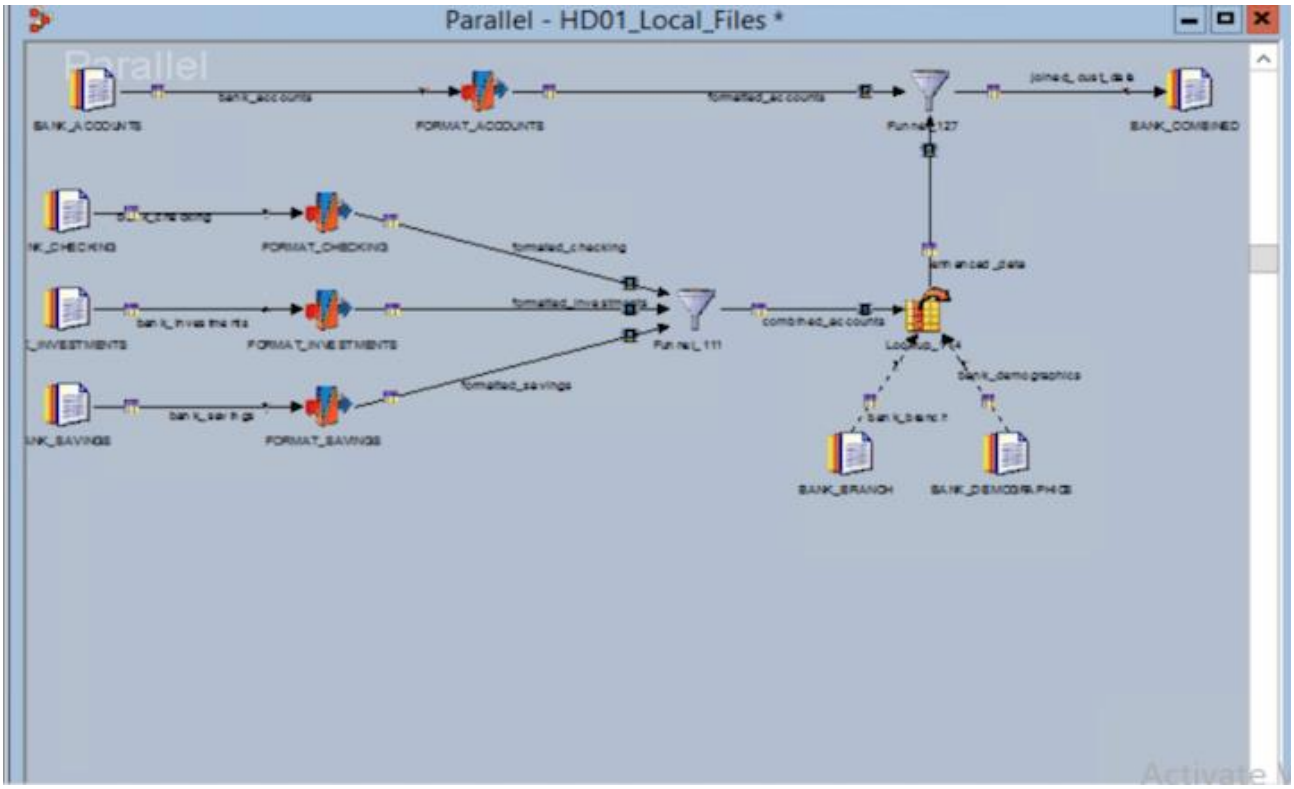
In this task, you complete these steps:

- Run a file processing job natively on the Linux host system
- Push the data to the Hadoop cluster

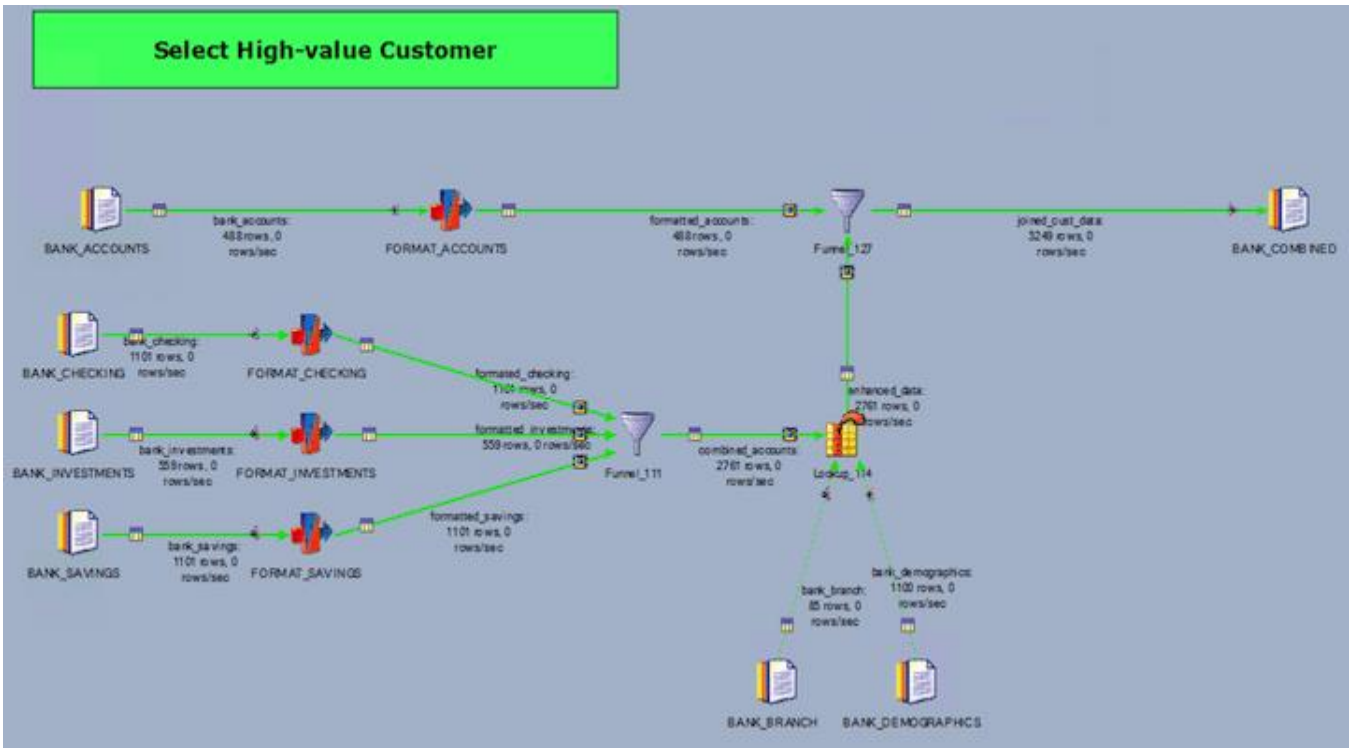
In this job, files that are based on Linux are ingested on the DataStage conductor node as the landing zone. After the files are ingested, DataStage processes the data files in a traditional DataStage configuration.

1. Click the **HD01\Local\Files** job and click **Edit**.

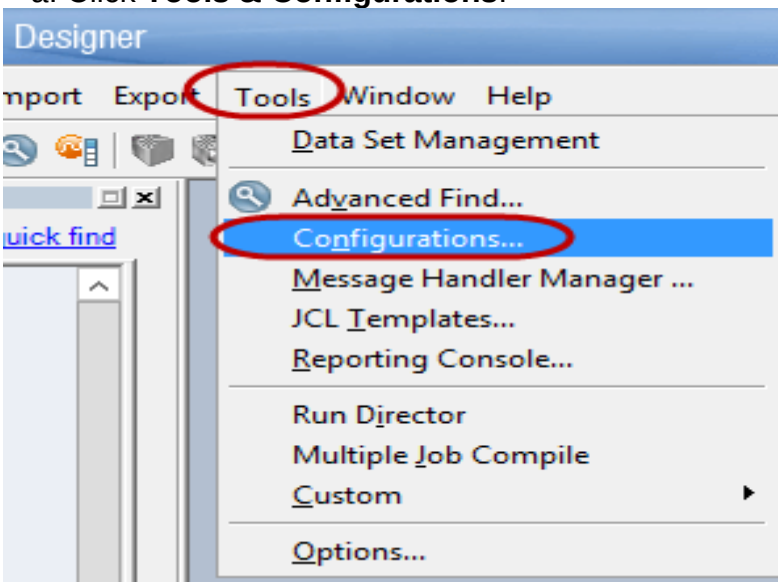

dstage1
Data Elements
DEMO
ENABLEMENT
HOL
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
JKLW
00_PREPARE
01_DBS_To_ODS
02_ODS_To_EDW
03_MDM_LOAD
04_MDM_To_EDW
04_MDM_To_ODS
05_RDM
06_SERVICES
07_BIG_DATA
08_DATA_REPLICATION
09_HADOOP
BANK2_COMBINED_SOURCES
COMBINE_CUST_INPUT
HD01_Local_Files
HD02_Load_HDFS
HD03_Process_HDF
HD04_Process_on_H
HD05_Create_HIVE
hol.bank_combined
JK_BANK1.BANK_AC
JK_BANK2.BANK_BI
JK_BANK2.BANK_CI
JK_BANK2.BANK_D
JK_BANK2.BANK_IN
JK_BANK2.BANK_SA
99_MISC
SHARED
Jobs
Machine Profiles
Match Specifications
MDM Integration
MNS Rules
Routines
Shared Containers
Shared Objects
Stage Types
Standardization Rules
Cross Project Compare...
Compare against...
Find dependencies
Find dependencies (deep)
Find where used
Find where used (deep)
Export
Multiple job compile...
Edit
Add to palette
Create copy
Rename
Delete
Properties


Parallel - HD01_Local_Files *
Parallel
BANK_ACCOUNTS
FORMAT_ACCOUNTS
BANK_COMBINED
FORMAT_CHECKING
FORMAT_INVESTMENTS
FORMAT_SAVINGS
BANK_BRANCH

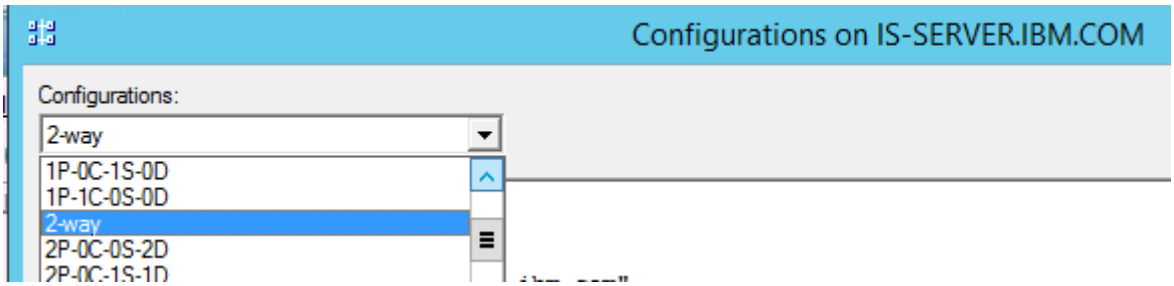
2. Compile the job by clicking the **Compile** icon on the toolbar.
3. Run the job by clicking the **Run** icon on the toolbar.
4. In the Job Run Options window, you can select the configuration file. For this demo, use the `default.apt` configuration file. Click **Run**.
When the job is finished, all the job links turn green and show the number of rows on each link.



5. Review the default configuration file:
a. Click **Tools & Configurations**.



b. From the Configurations list, select 2-way.

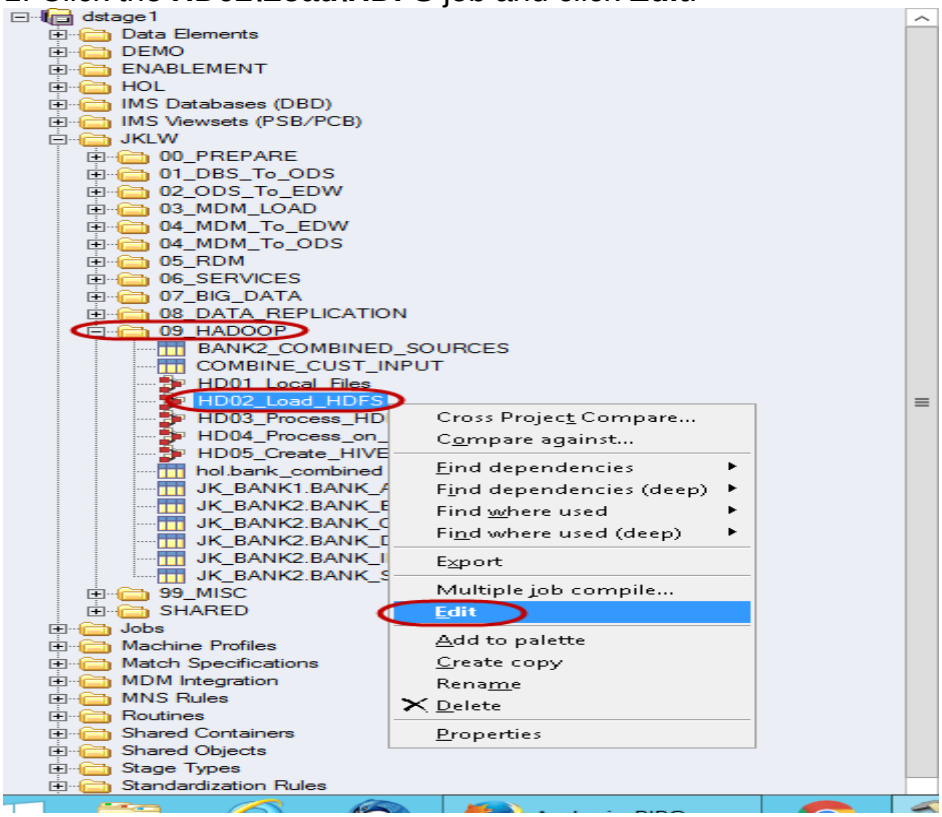


This file is a traditional configuration file, where the nodes and the location of the resources are named. A 2-node configuration file is shown, but you can have a 1-node configuration file.

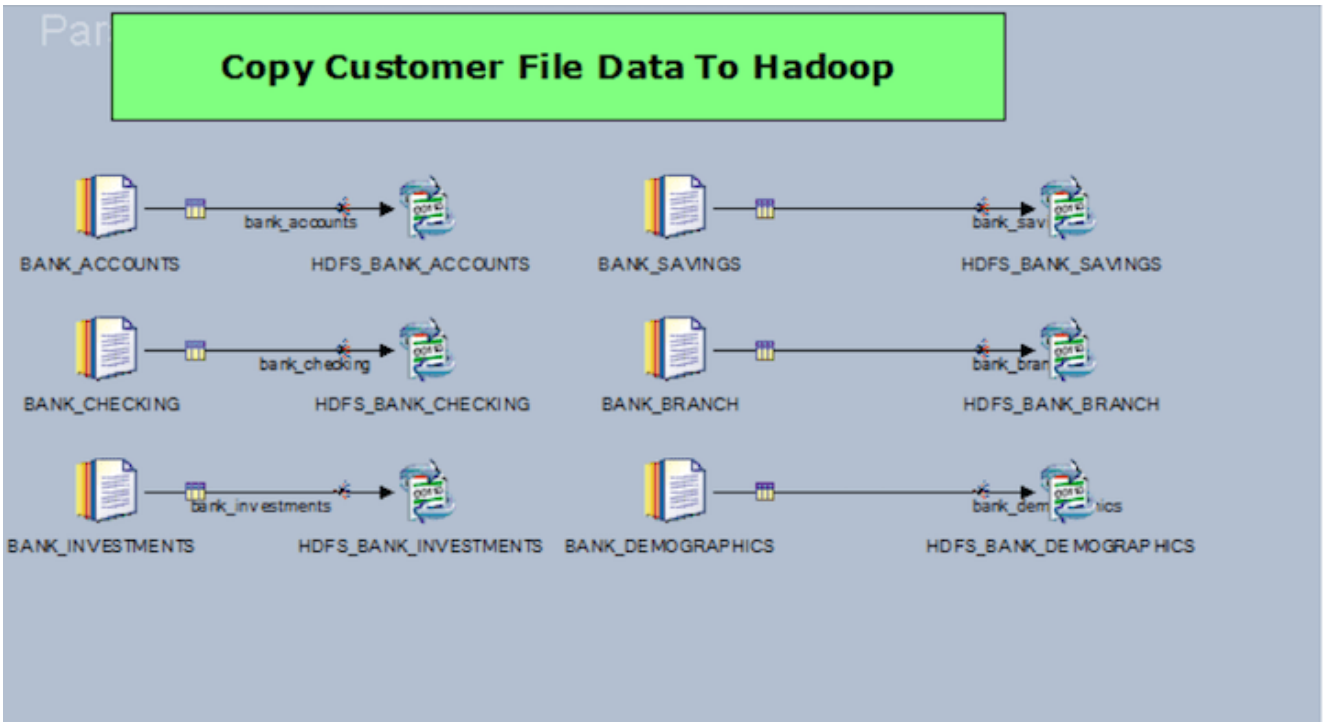
c. Click Close.

Load data into the Hadoop HDFS

1. Click the **HD02\Load\HDFS** job and click **Edit**.

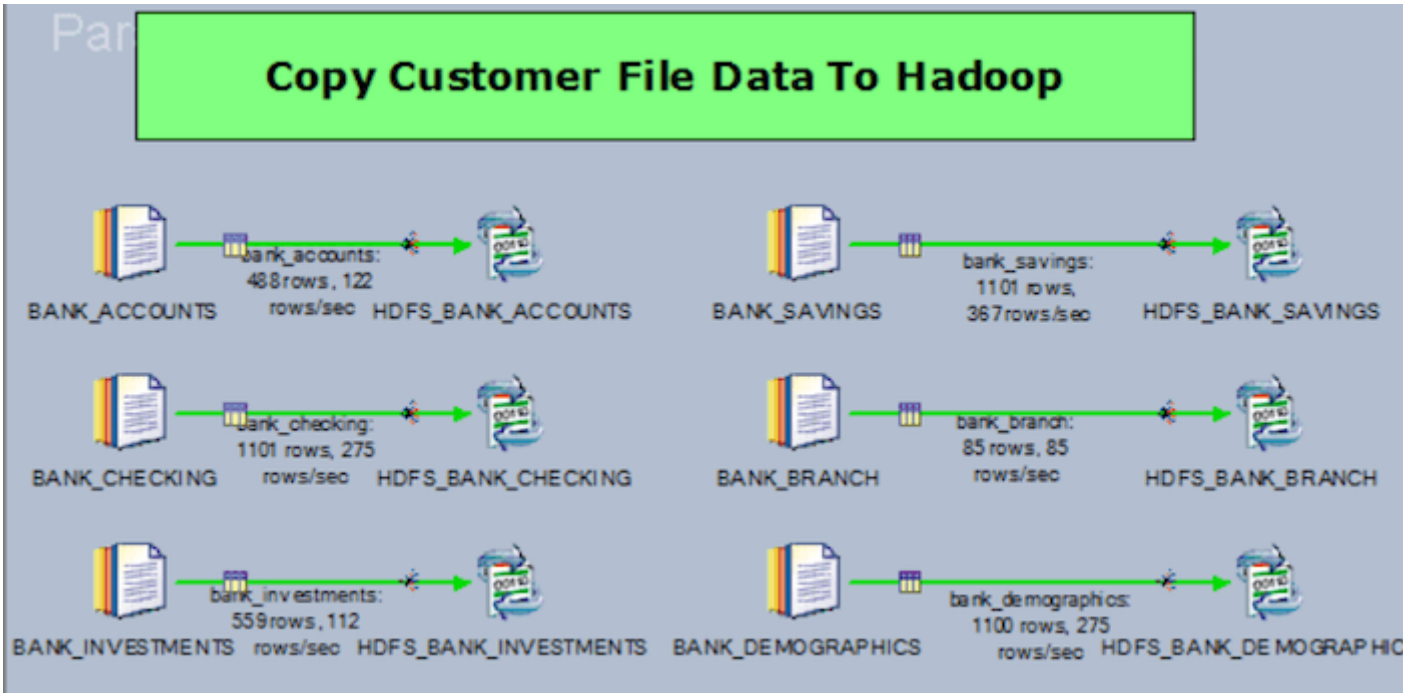


The HD02_Load_HDFS job loads your files onto HDFS for processing later.



2. Compile the job by clicking the Compile icon on the toolbar.
3. Run the job by clicking the Run icon on the toolbar.
4. In the Job Run Options window, you can select the configuration file. Use the `default.apt` configuration file. Click **Run**.

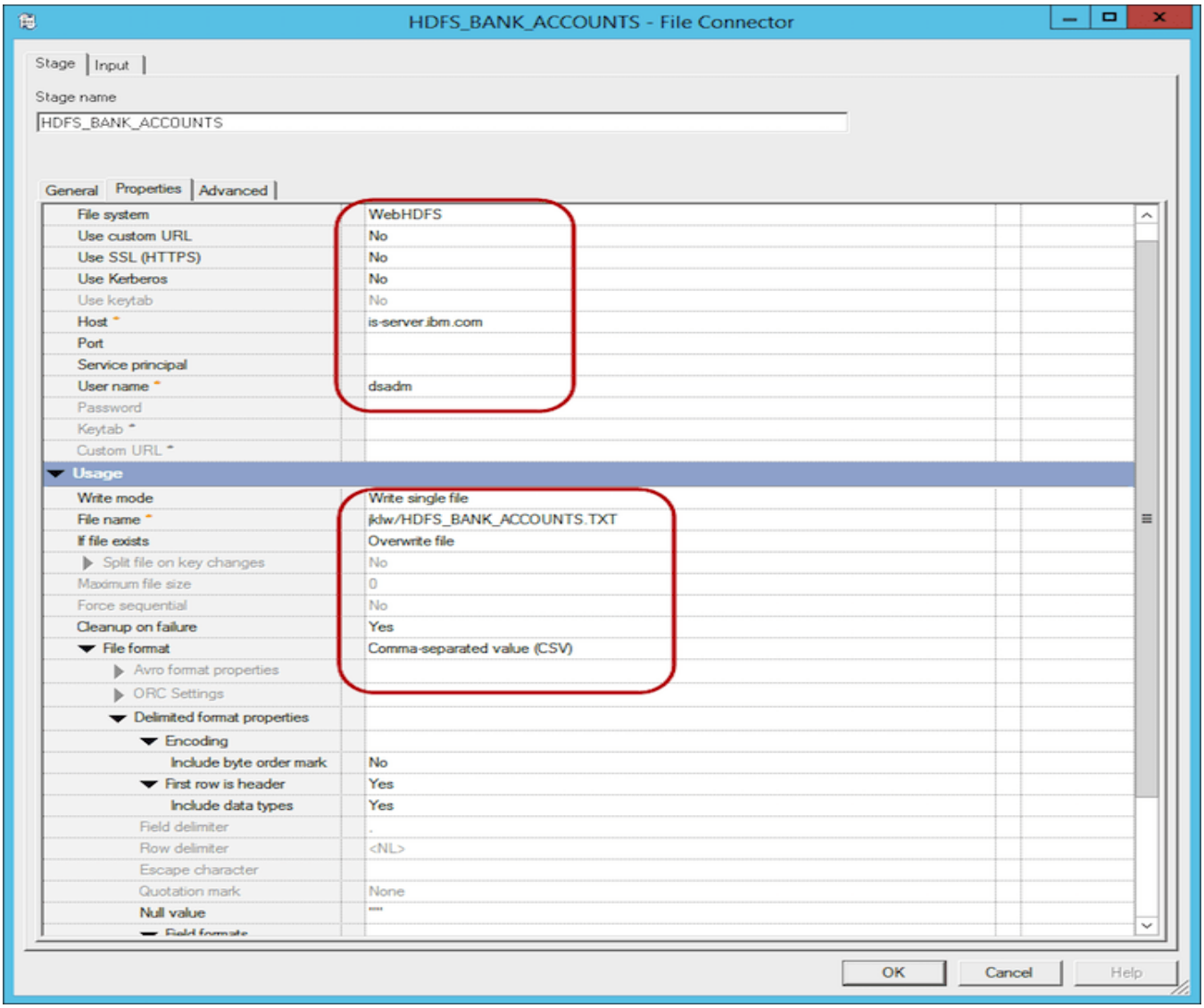
When the job is finished, all the job links turn green and show the number of rows on each link.



Review a File Connector property

1. Click the **HDFS\BANK\ACCOUNTS** file connector.

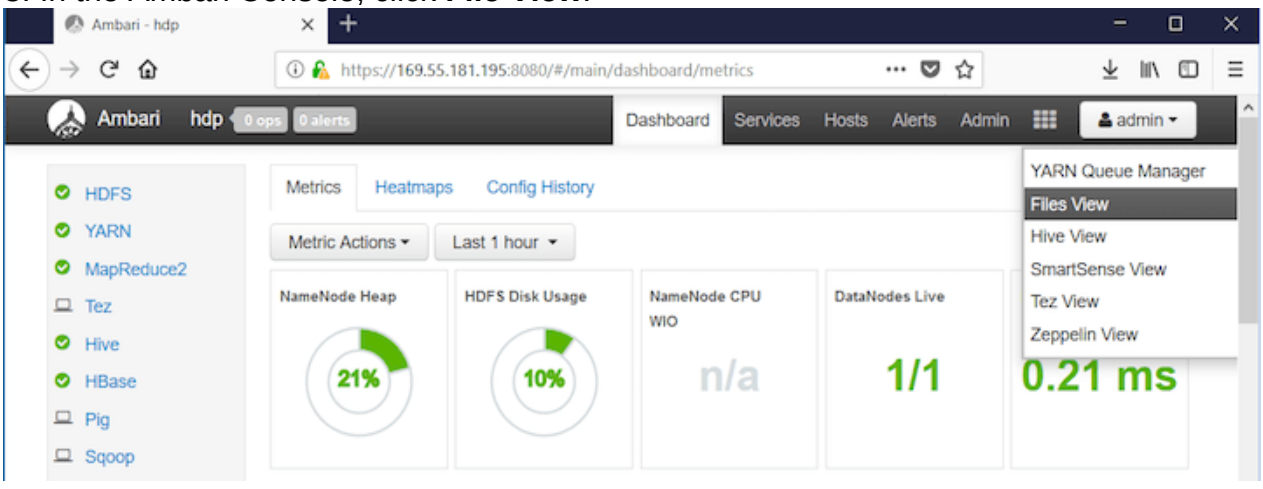
The File Connector window shows how you assigned the connectivity attributes for the big data server. You can also see the file attributes for where and how you write the data. Your host and file path might be different.



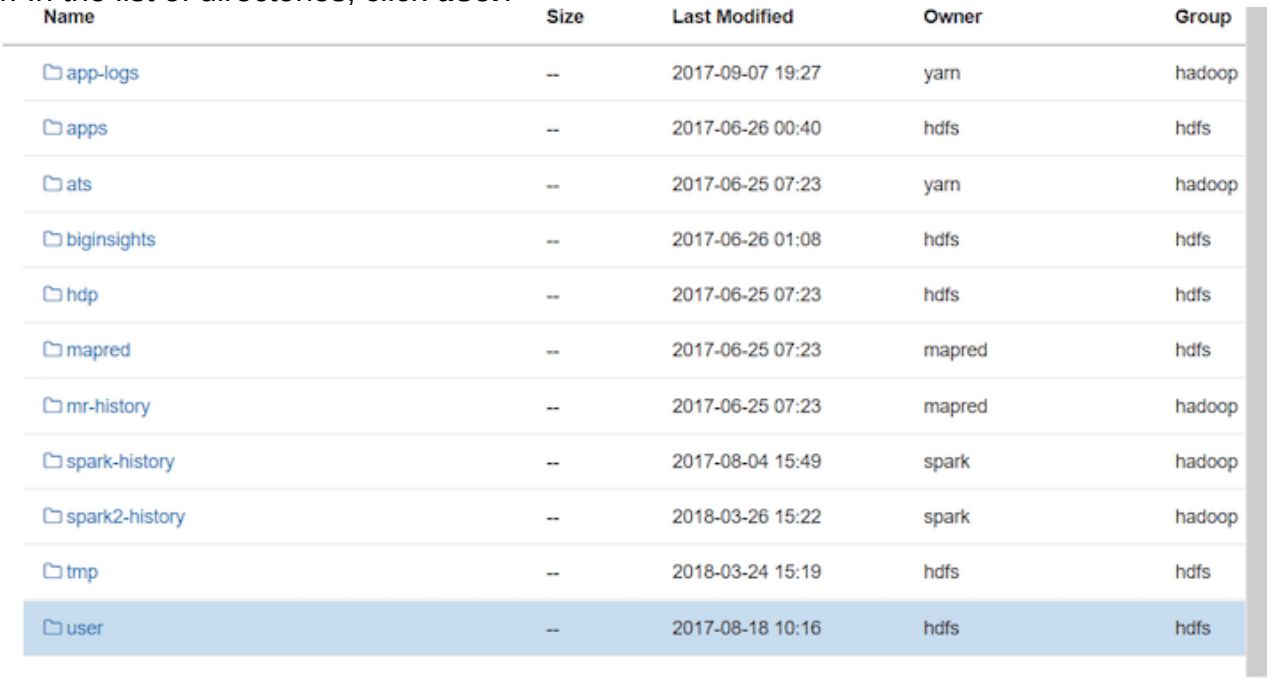
When you're finished reviewing the information, click **OK** to exit the properties window.

2. Review the data in the HDFS files system. The Ambari web console is open and the user admin is logged in.

3. In the Ambari Console, click **File View**.

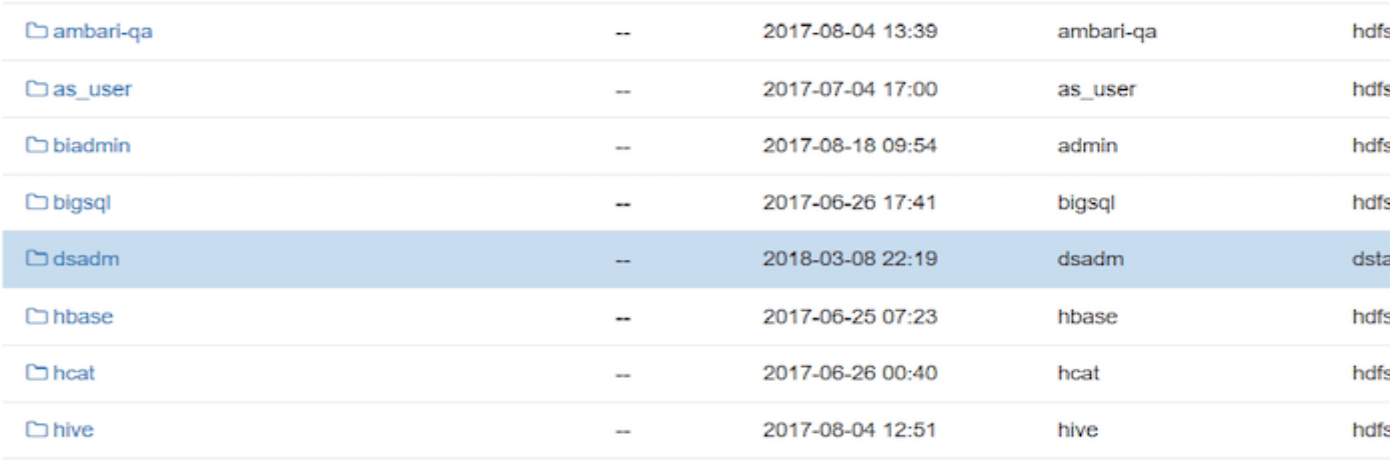


4. In the list of directories, click **user**.



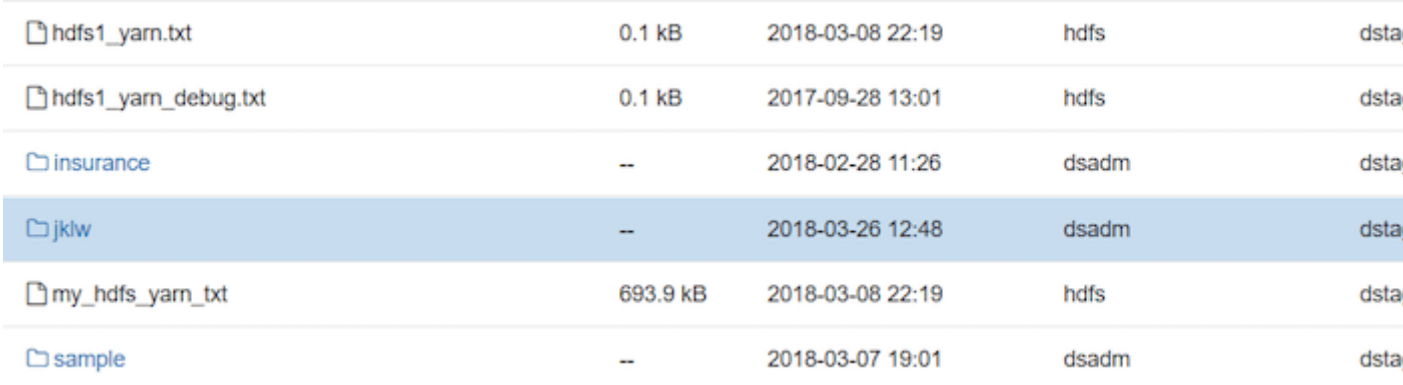
Name	Size	Last Modified	Owner	Group
app-logs	--	2017-09-07 19:27	yarn	hadoop
apps	--	2017-06-26 00:40	hdfs	hdfs
ats	--	2017-06-25 07:23	yarn	hadoop
biginsights	--	2017-06-26 01:08	hdfs	hdfs
hdp	--	2017-06-25 07:23	hdfs	hdfs
mapred	--	2017-06-25 07:23	mapred	hdfs
mr-history	--	2017-06-25 07:23	mapred	hadoop
spark-history	--	2017-08-04 15:49	spark	hadoop
spark2-history	--	2018-03-26 15:22	spark	hadoop
tmp	--	2018-03-24 15:19	hdfs	hdfs
user	--	2017-08-18 10:16	hdfs	hdfs

5. Click the **dsadm** directory.



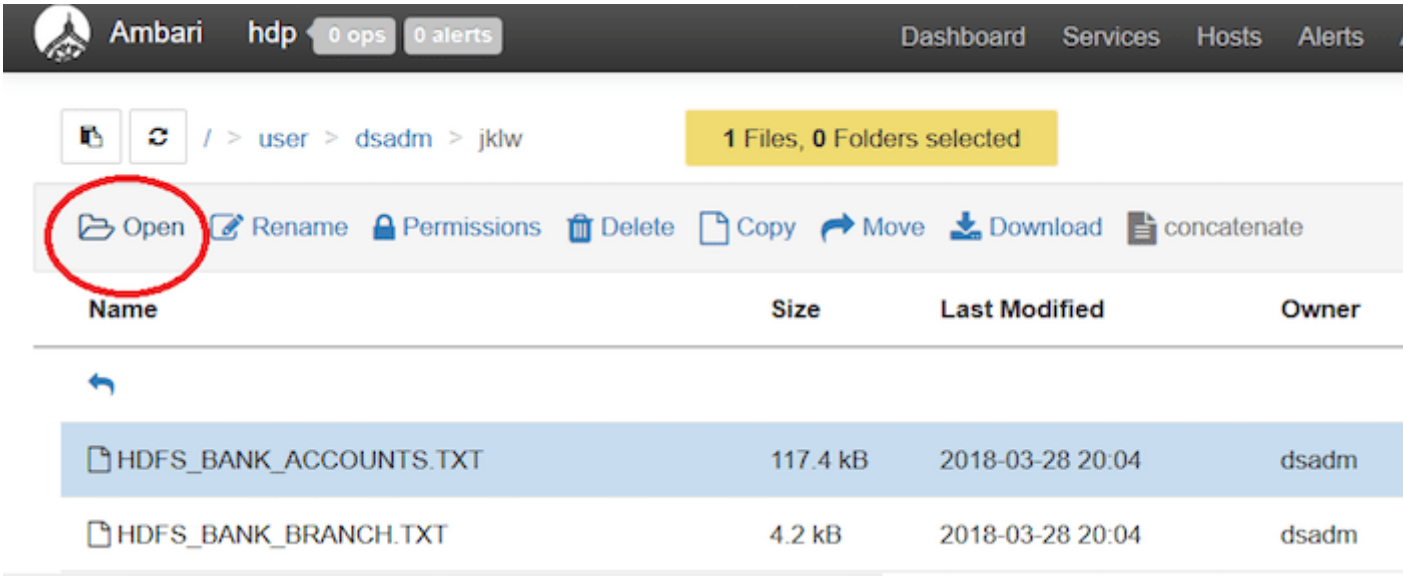
ambari-qa	--	2017-08-04 13:39	ambari-qa	hdfs
as_user	--	2017-07-04 17:00	as_user	hdfs
biadmin	--	2017-08-18 09:54	admin	hdfs
bigsql	--	2017-06-26 17:41	bigsql	hdfs
dsadm	--	2018-03-08 22:19	dsadm	dsta
hbase	--	2017-06-25 07:23	hbase	hdfs
hcat	--	2017-06-26 00:40	hcat	hdfs
hive	--	2017-08-04 12:51	hive	hdfs

6. Click the **jklw** folder.
   The files that you loaded into the HDFS file system are in the folder.

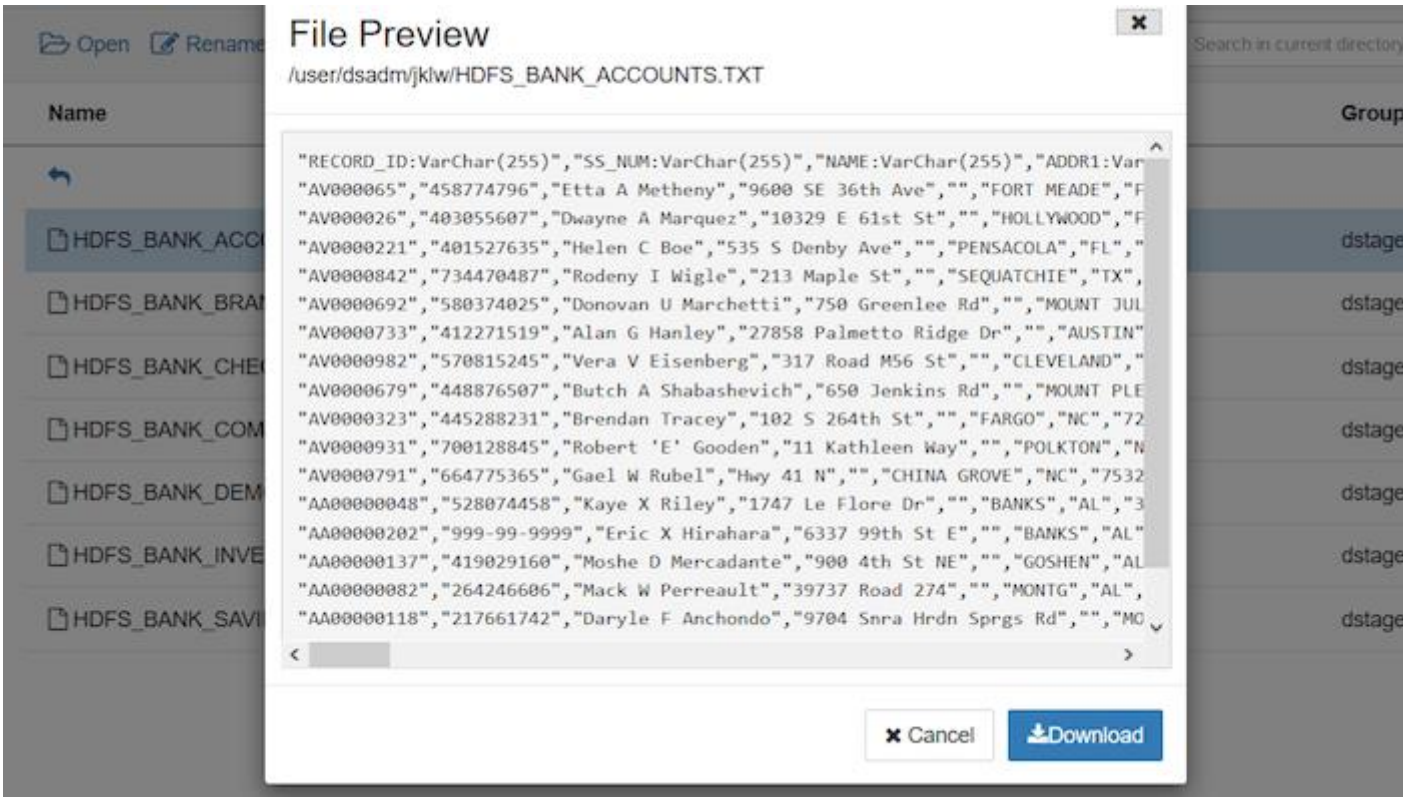


hdfs1_yarn.txt	0.1 kB	2018-03-08 22:19	hdfs	dsta
hdfs1_yarn_debug.txt	0.1 kB	2017-09-28 13:01	hdfs	dsta
insurance	--	2018-02-28 11:26	dsadm	dsta
jklw	--	2018-03-26 12:48	dsadm	dsta
my_hdfs_yarn_txt	693.9 kB	2018-03-08 22:19	hdfs	dsta
sample	--	2018-03-07 19:01	dsadm	dsta

For the purposes of this demo, you look at the HDFS_BANK_ACCOUNTS.TXT file.
7. Click **Open** on the top menu bar.



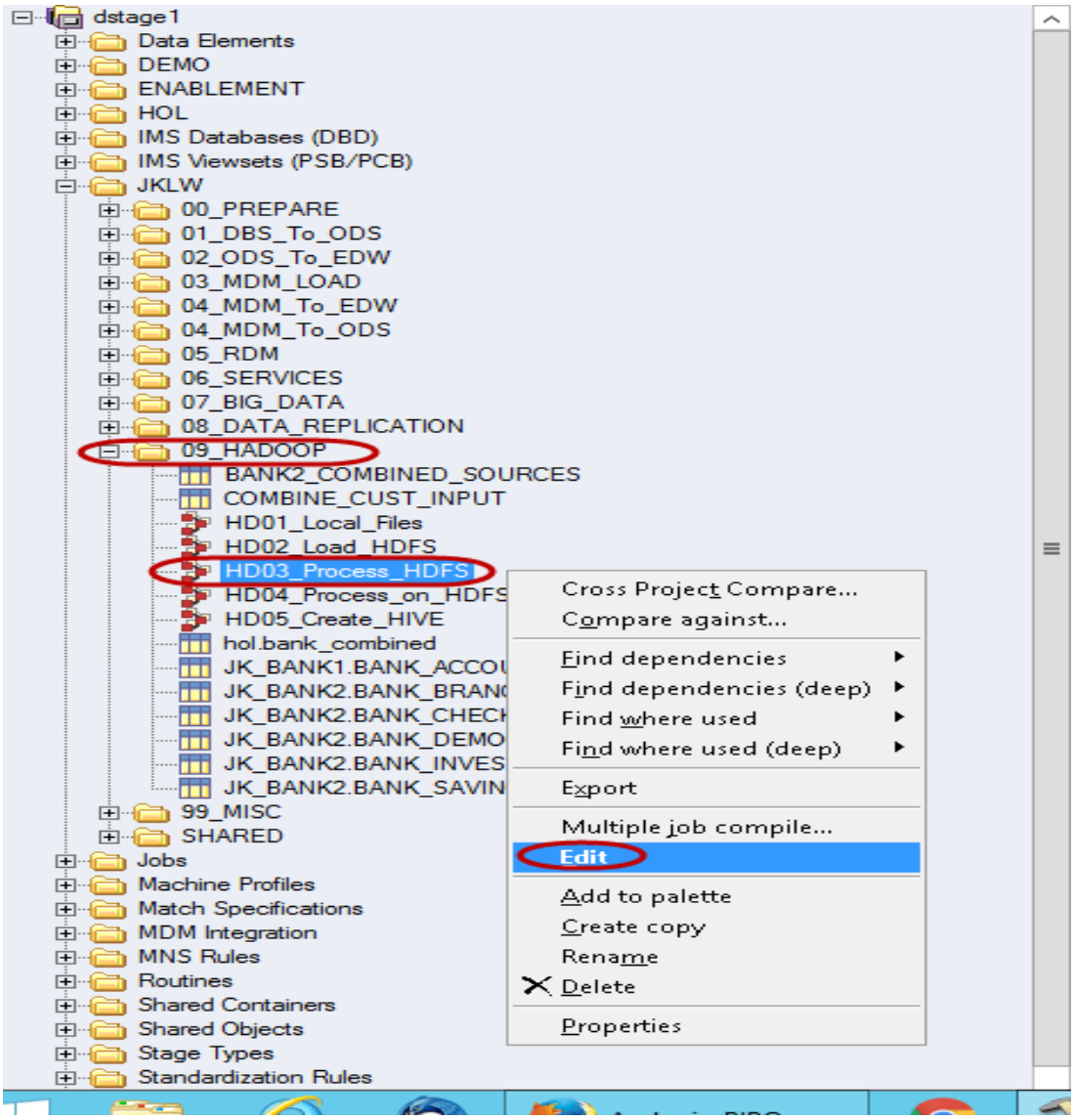
8. Review the data in the file. When you're finished, click **Cancel**.



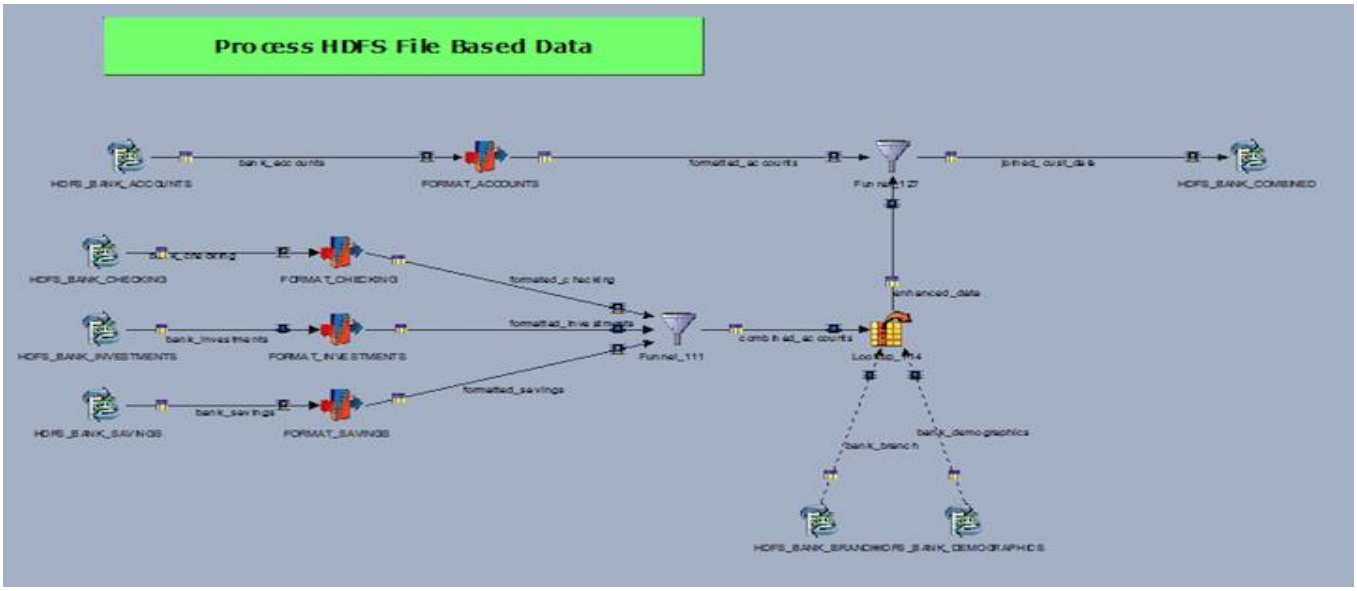
Process the Hadoop files

Process the HDFS-file-based data in a traditional DataStage configuration.

1. Click the **HD03\Process\HDFS** job and then click Edit.

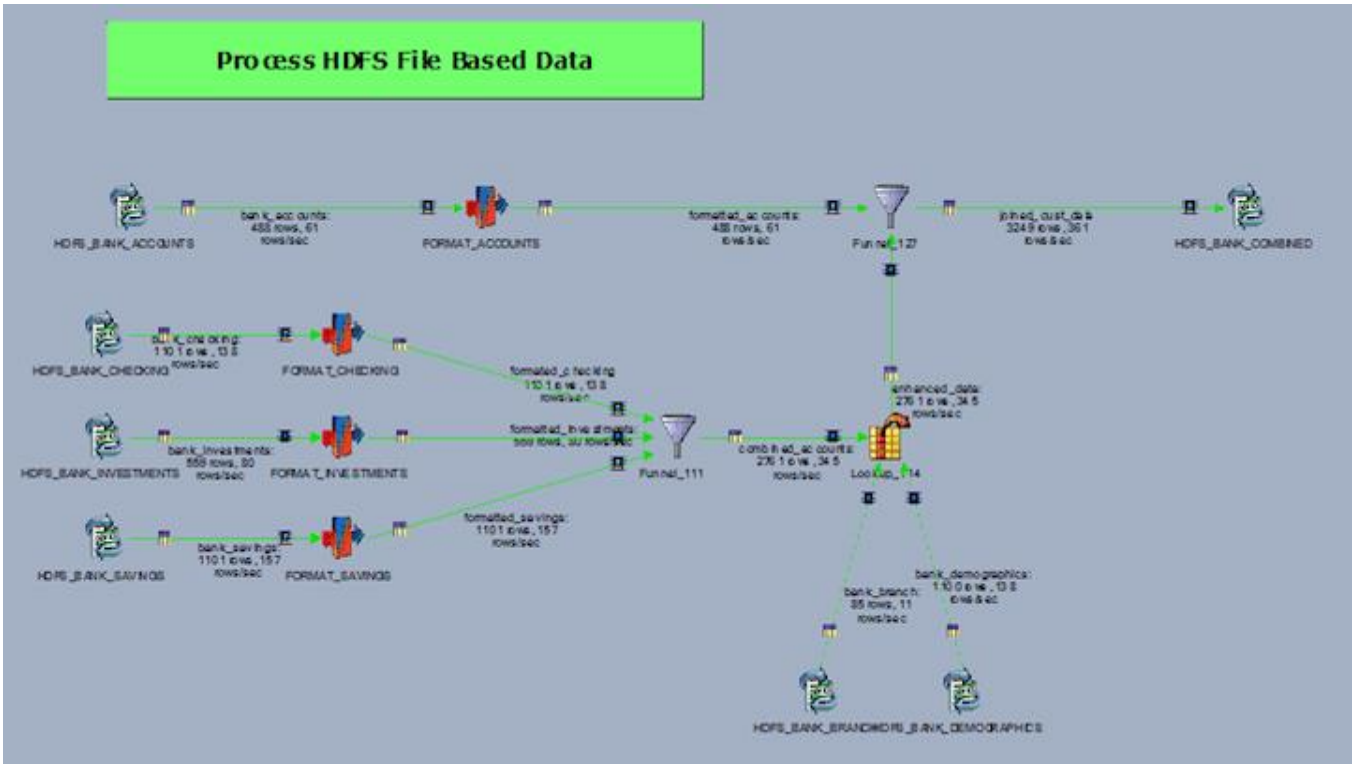


The HD03\Load\HDFS job reads data from your Hadoop HDFS file system, processes the data, and writes the consolidated customer data back to the Hadoop HDFS file system.



2. Compile the job by clicking the **Compile** icon on the toolbar.
3. Run the job by clicking the **Run** icon on the toolbar.
4. In the Job Run Options window, you can select the configuration file. Use the `default.apt` configuration file. Click **Run**.

When the job is finished, all the job links turn green and show the number of rows on each link.

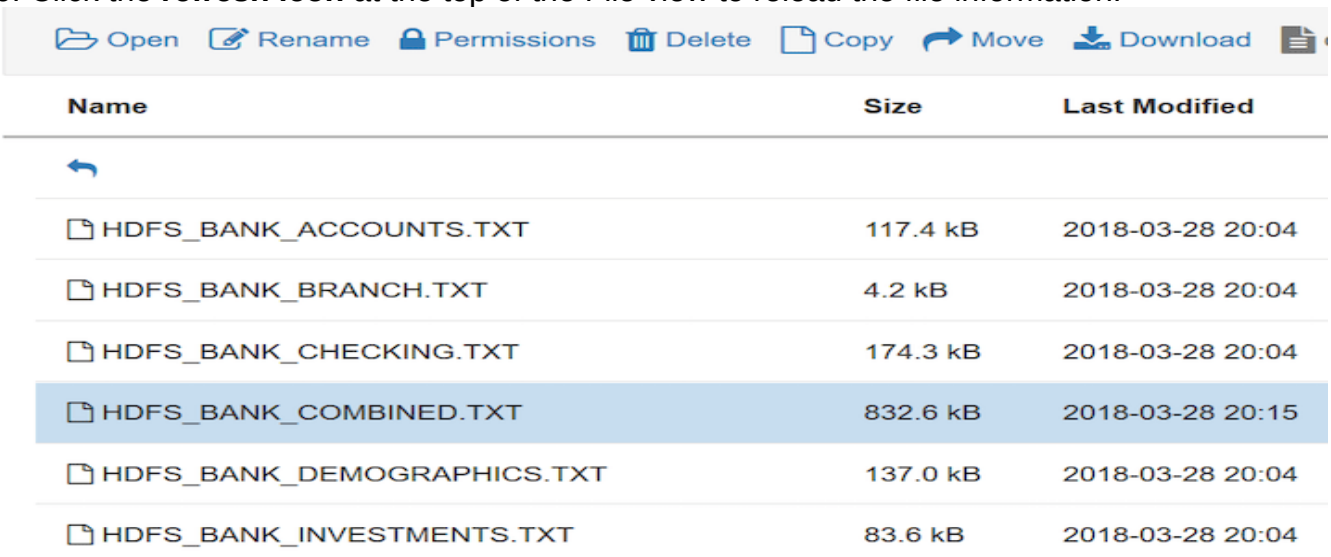


5. Return to the HDFS File view by clicking the Maximize Window icon.

HDFS_BANK_ACCOUNTS.TXT	117.4 kB	2018-03-28 20:04
HDFS_BANK_BRANCH.TXT	4.2 kB	2018-03-28 20:04
HDFS_BANK_CHECKING.TXT	174.3 kB	2018-03-28 20:04
HDFS_BANK_COMBINED.TXT	832.6 kB	2018-03-28 13:06
HDFS_BANK_DEMOGRAPHICS.TXT	137.0 kB	2018-03-28 20:04
HDFS_BANK_INVESTMENTS.TXT	83.6 kB	2018-03-28 20:04
HDFS_BANK_SAVINGS.TXT	161.4 kB	2018-03-28 20:04

Notice the time that the HDFS\BANK\COMBINED\TEXT file was written onto the Hadoop file system.

6. Click the **refresh icon** at the top of the File view to reload the file information.



The file time stamp is updated after the job ran.

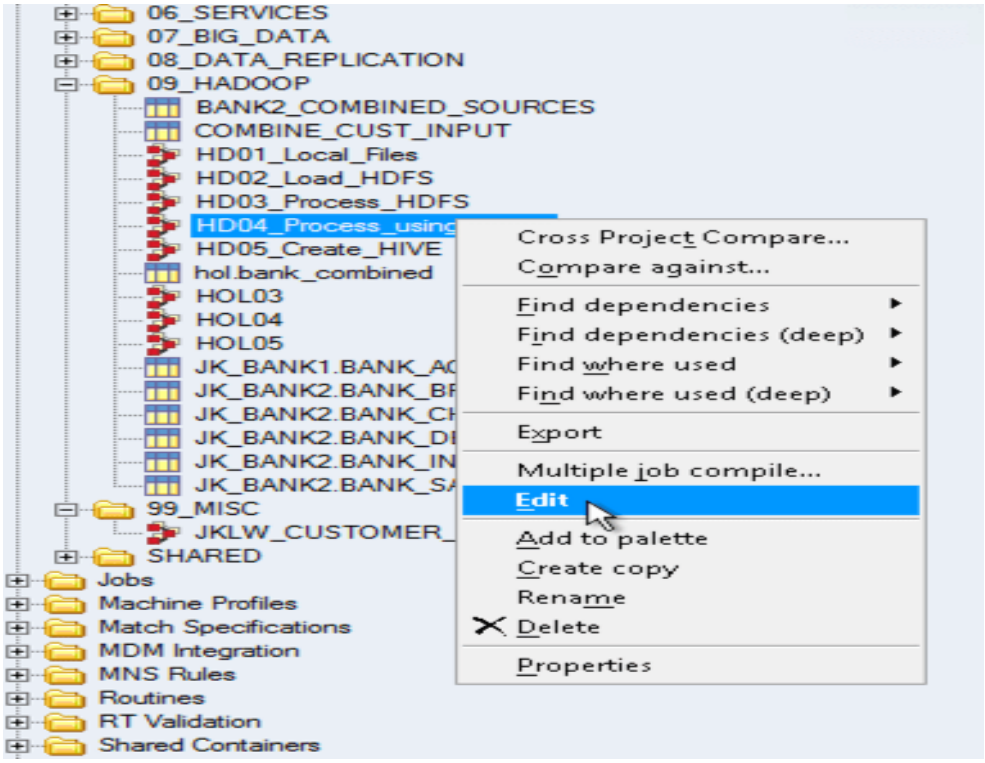
Close the File view by clicking the highlighted file.

7. Return to DataStage and close the job.

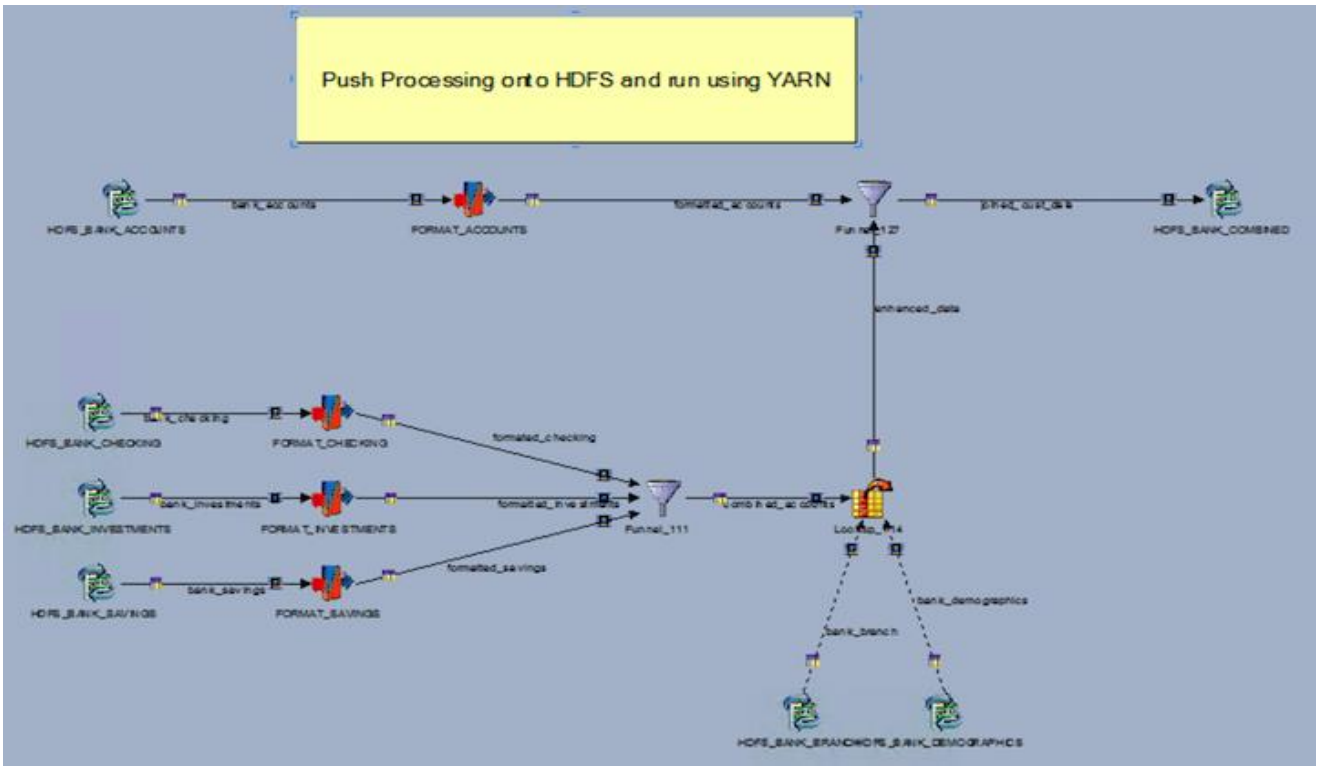
Run ETL processing inside Hadoop by using YARN

Process the HDFS-file-based data by pushing the processing to the Hadoop cluster and allowing the processing to run by using YARN.

1. Right-click the HD04\Process\using\YARN job and click **Edit**.



The HD04\Process\on\HDFS\using\YARN job reads data from your Hadoop HDFS file system, processes the data, and writes the consolidated customer data back to the Hadoop HDFS file system entirely in the Hadoop cluster.



2. Compile the job by clicking the **Compile** icon on the toolbar.
3. Run the job by clicking the **Run** icon on the toolbar.
   In the Job Run Options window, you can see that you have two properties.
   The first property points to the `yarnconfig.cfg` file. This file indicates to DataStage which settings it needs to communicate with and run the process on YARN.

   The second property points to the configuration file so that DataStage detects how many nodes to run and where the resources are.
4. Before you click **Run**, learn about the properties:
   * On the server, the `yarnconfig.cfg` file contains a number of settings to tune how DataStage processes are run with YARN. Two important settings are `APT_YARN_MODE` and `APT_YARN_USE_HDFS`.

   * The `APT_YARN_MODE` setting tells the engine where to run. A value of `false` tells the engine to run normally on the host system. A value of `true` tells the engine to hand the process to YARN for processing on the Hadoop cluster.

   * The `APT_YARN_USE_HDFS` setting tells the engine whether the data resources are being written to the local file system or the HDFS file system. A setting of `false` indicates that the data is written to the local file system. A setting of `true` indicates that the data is being written to the HDFS file system.

   * You can use many other settings to further tune the interaction. Each setting is described in the `yarnconfig.cfg` file.

```
##   Licensed Materials - Property of IBM
##   (c) Copyright IBM Corp. 2015
#
# DataStage PX Yarn Configuration
# ================================================
# Lines in this file are either comments, introduced by a # sign like this,
# or of the form "key=value". Key lines may be commented out below.
#
# IMPORTANT:
# Ensure when making changes to this file that it is saved with the encoding set to
# UTF-8. Please be aware if the encoding isn't set to UTF-8 this may produce undesired
# behaviour.

APT_YARN_MODE=true
# If defined and set to 1 or true runs the given PX job on
# the local Hadoop install in YARN mode.

APT_YARN_CONTAINER_VCORES=0
# Defines the number of virtual cores that the containers will request to run
# PX Section Leader and Player processes in.
# The default is 0 which means "Don't set it".

APT_YARN_CONTAINER_SIZE=64
# Defines the size in MBs of the containers that will be requested to run
# PX Section Leader and Player processes in.
# The default is 64MB if not set.

APT_YARN_CONTAINER_SIZE_AUTO=false
# When defined will automatically use the estimated container size for the largest partition
# as the container size if it is larger than the set container size.
# It accepts a value of true or false.

APT_YARN_BASE_PROCESS_SIZE=16
# Defines the base process size in MB to be used when estimating the size of containers
# to request from YARN.  The default value for this is 16MB.

APT_YARN_ALLOCATION_TIMEOUT=180
# Specifies the amount of time in seconds to wait for allocations of containers
```

* When parallel jobs run on Hadoop, the jobs request a set of containers from YARN. The containers represent the resources that the job was allocated. Each resource has a designated amount of virtual CPU and memory for each container. The number of containers that are requested is equal to the number of logical nodes that are defined in the `APT_CONFIG_FILE` file.

  * These files can be configured in one of three ways: static, dynamic, and mixed.

  * A static file looks like a regular `APT_CONFIG_FILE`, except the resource disk can be either on the local file system or in HDFS depending on the `APT_YARN_USE_HDFS` setting in the `yarnconfig.cfg` file.

```
{
   node "node1"
   {
      fastname "mymachine.domain.com"
      pools ""
      resource disk "/mydisk/tmp" {pools ""}
      resource scratchdisk "/myscratch/tmp" {pools ""}
   }
}
```

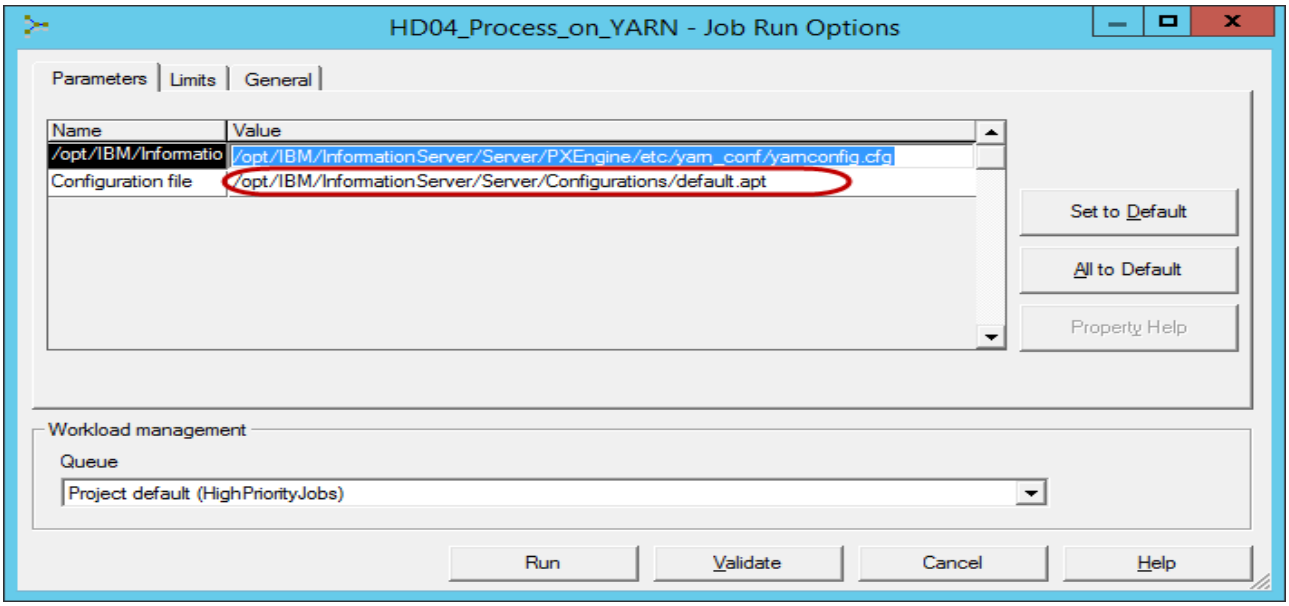
* A dynamic configuration file uses the same format as the static Information Server configuration files, which assign fixed nodes to the job. However, a dynamic configuration file uses a fastname value of `\$host`, as opposed to a static configuration file that usually contains a host name. One node in the configuration file must contain the engine tier node, but this node can be defined with a conductor node pool if you don't want to run data processing on the conductor node.

```
{ node "node0"
 {
  fastname "the-engine-tier-machine.domain.com"
  pools "conductor"
  resource disk "/sandbox/bsmith/tmp" {pools ""}
  resource scratchdisk "/scratch" {}
  }
   node "node1"
  {
   fastname "$host"
   pools ""
   resource disk "/sandbox/bsmith/tmp" {pool ""}
   resource scratchdisk "/sandbox/bsmith/tmp" {pools ""}
   instances 30
}
}
```

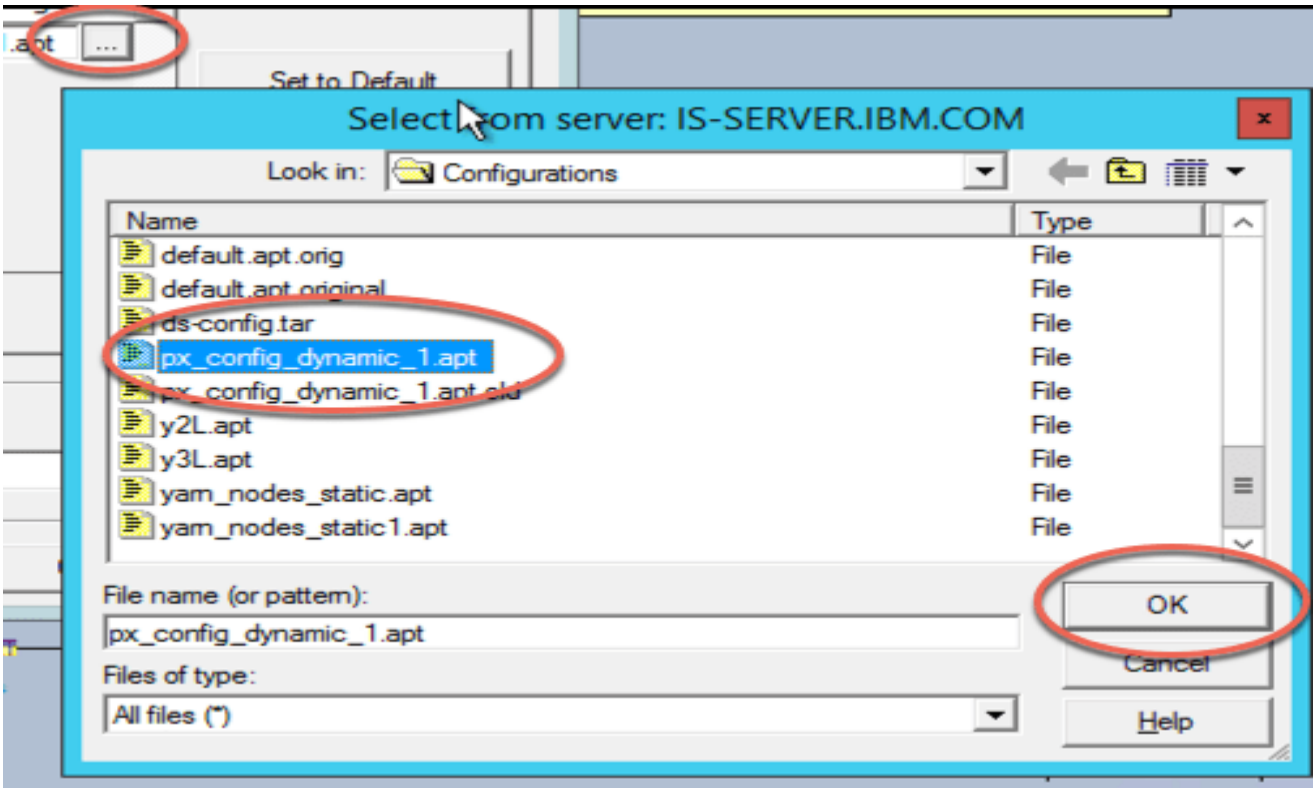
* A mixed configuration file contains a mix of static host names (an actual host name) and dynamic host names (fastname `\$host`). The following configuration file specifies that the first 30 nodes are defined by YARN. Then, the next 10 nodes are run on machineA, and the 41st node runs on machineB:

```
{
  node "node0"
{
  fastname "the-engine-tier-machine.domain.com"
  pools "conductor"
  resource disk "/sandbox/bsmith/tmp" {pools ""}
  resource scratchdisk "/scratch" {}
}
  node "node1"
 {
  fastname "$host"
  pools ""
  resource disk "/mydisk1/tmp" {pools ""}
  resource scratchdisk "/myscratchdisk1/tmp" {pools ""}
  instances 30
  }
   node "node31"
   {
   fastname "machineA.domain.com"
    pools ""
    resource disk "/mydisk2/tmp" {pools ""}
    resource scratchdisk "/myscratchdisk2/tmp" {pools ""}
    instances 10
   }
    node "node41"
   {
   fastname "machineB.domain.com"
    pools ""
    resource disk "/mydisk3/tmp" {pools ""}
    resource scratchdisk "/myscratchdisk3/tmp" {pools ""}
   }
}
```

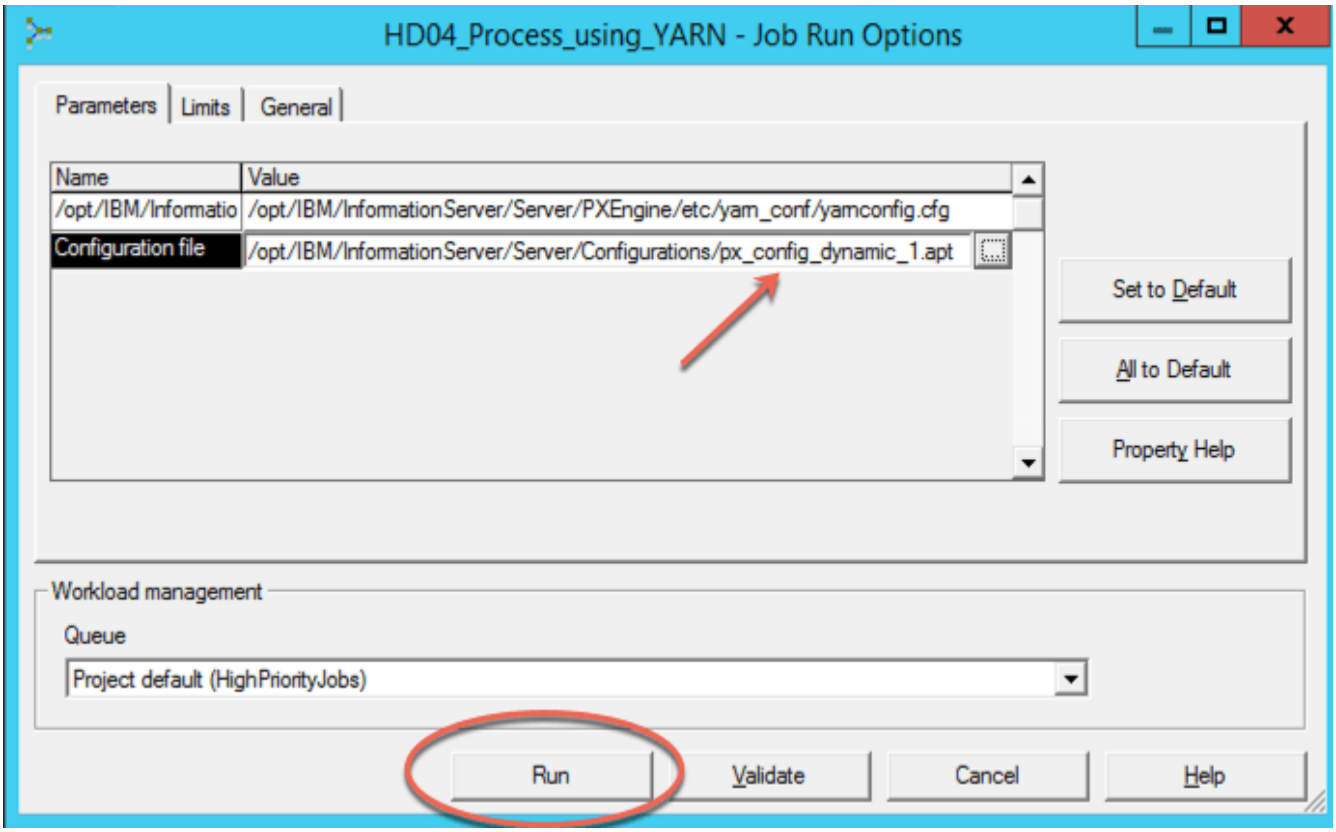
5. In the Job Run Options window, use a dynamic configuration file for your processing. Click the field that defines the configuration file to show the **Options** icon.



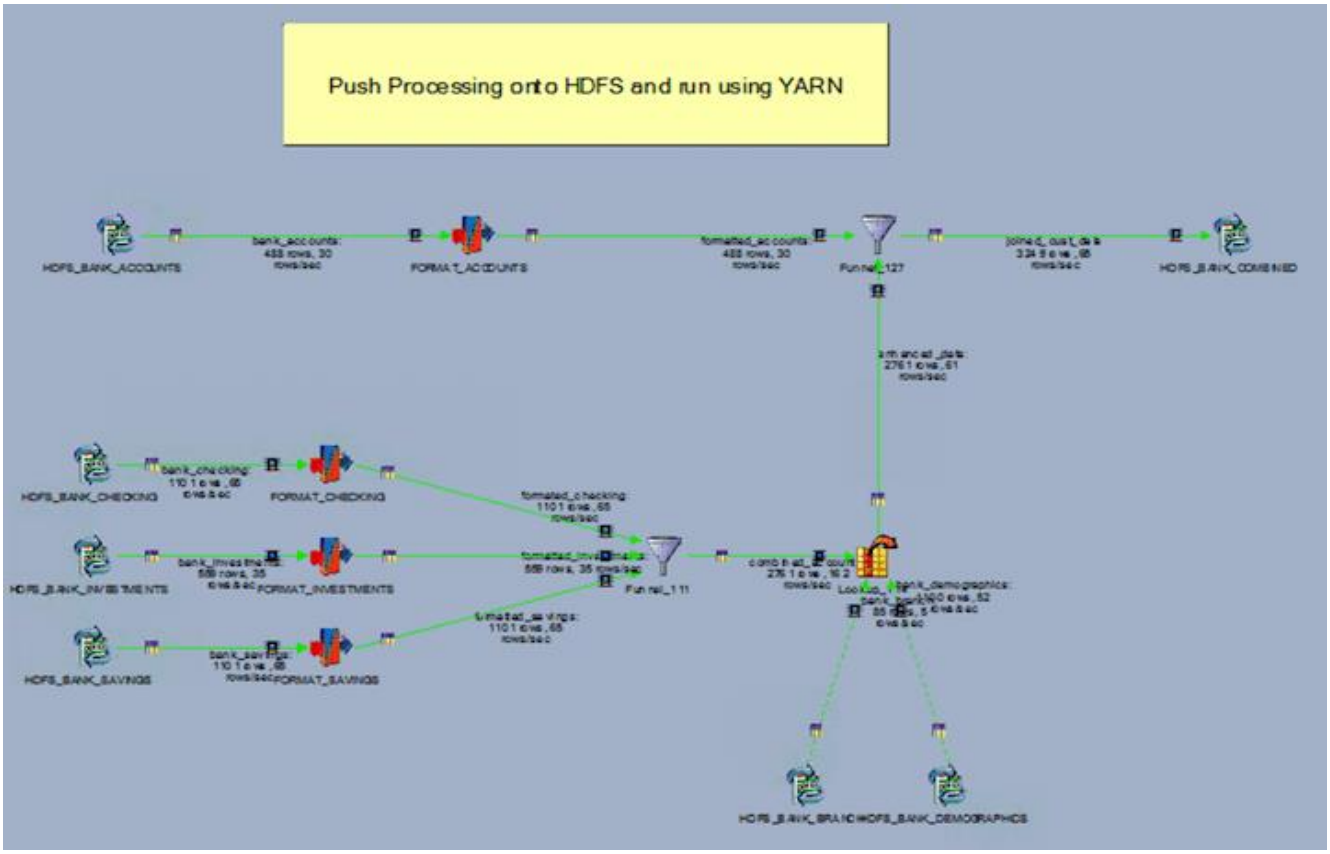
A file-browser window is displayed and the `px_config_dynamic_1.apt` file is selected.
6. Click **OK**.



7. Ensure that you're using the correct configuration file and click **Run**.



When the job is finished, all the job links turn green and show the number of rows on each link.



8. Click the center of the lower part of the page to raise the job log window.
9. Look for these entries:

```
Starting Job HD04_Process_using_YARN. (...)
Environment variable settings:  (...)
Parallel job initiated
OSH script  (...)
Parallel job default NLS map UTF-8, default locale OFF
main_program: IBM InfoSphere DataStage Enterprise Edition 11.5.0.8169  (...)
main_program: The open files limit is 1024; raising to 4096.
main_program: conductor uname: -s=Linux; -r=2.6.32-642.13.1.el6.x86_64; -v=#1 SMP Wed Nov 23 16:03:01 EST 2016; -n=is-serve...
main_program: orchgeneral: loaded (...)
main_program: Parallel Engine running in YARN execution mode.
main_program: IPv6 isn't currently supported by Hadoop/YARN and the required environment variable APT_USE_IPV4 is not set. It ...
HDFS_BANK_ACCOUNTS: Accessing file via WebHDFS file system.
HDFS_BANK_CHECKING: Accessing file via WebHDFS file system.
HDFS_BANK_INVESTMENTS: Accessing file via WebHDFS file system.
HDFS_BANK_SAVINGS: Accessing file via WebHDFS file system.
HDFS_BANK_DEMOGRAPHICS: Accessing file via WebHDFS file system.
HDFS_BANK_BRANCH: Accessing file via WebHDFS file system.
HDFS_BANK_COMBINED: The connector was configured to run in parallel on 3 nodes, but the Read/Write mode is Read/Write sin...
HDFS_BANK_COMBINED: Accessing file via WebHDFS file system.
main_program: APT configuration file: /opt/IBM/InformationServer/Server/Configurations/px_config_dynamic_1.apt (...)
```

Summary

You ran a traditional Data Warehouse ETL job and moved data from Data Warehouse to Hadoop. Then, you ran a Data Warehouse ETL by using Hadoop data and ran ETL processing inside Hadoop by using YARN.