Self-service data onboarding with Pentaho Metadata Injection

Script

# Resetting the demo

1. Rerun the onboard/setup/j\_create\_metadata\_db.kjb PDI job to rebuild the metadata database.
2. Delete the /metadata\_injection folder in HDFS
3. Delete the onboard/PDI/injected\* transformation(s)
4. Delete the files in onboard/data/

# Running the demo

Start by showing the slide deck to provide an overview of the topic of metadata injection.

## Business Case

A Human Resources SaaS software company, we’ll call HRaaS needs a smarter way to onboard data from their customers.

Customers have many different systems so data files comes in many different formats. Customers must also upload multiple types of files for different data. For example employee data, department hierarchies, office locations, and so on.

This required a manual process for staff to work with each customer to transform and load data into HRaaS’ multi-tenant database. The result is a separate data workflow for each file for each customer resulting in a massive amount of data engineering.

## Traditional ETL

Open onboard/PDI/hard\_coded\_metadata.ktr

Here’s a traditional data workflow that loads a department data file for a single customer into HDFS.

There are four steps to transform the customer file into the target format in HDFS:

1. The CSV file input step requires metadata to parse the data file. Things like the delimiter, the fields, data types, string lengths, and so on.
2. Then we rename the customer’s field to match the HRaaS nomenclature. Metadata is required here for the mapping.
3. Next we may need to transform the data to match the target schema. For example, we may want to specify a date format or numeric precision. Again, this is all about metadata.
4. Finally, we may want to order the output fields in the target or specify a sub-set of fields supplied by the customer. We may want to be able to gracefully ignore extra fields included in the customer’s data file.

## ETL Template

Next I’ll walk through the steps to develop a self-service solution that does not require your customer to be an ETL developer.

Open onboard/PDI/template.ktr

Here’s an example of the same workflow in template form. The metadata from all the steps has been removed. (edit each step)

## ETL Metadata Injection

Open onboard/PDI/t\_file2hdfs.ktr

The next step is to build another workflow that will perform the metadata injection. Here’s an example:

The metadata injection step does the actual injection. (edit the ETL metadata injection step)

Here we specify the workflow template. (File tab)

Here we specify what we want to do with the template. (Options tab)

* The default is to render and immediately run the template
* Here we specify that we want to save the rendered template to a file
* We also have the option of just saving the template and not running it. Perhaps we want to run it later. Or perhaps we want to want to run the rendered template in-cluster on Hadoop.

Finally, we have the Inject Metadata tab which lets us connect metadata from external sources into the individual steps in the template.

We see the four steps in the template: CSV file input, Rename, Transform data, and HDFS output.

We have two paths of metadata connecting into the ETL Metadata Injection step.

This tab lets us connect fields from these metadata feeds to individual settings and columns in the steps.

For example, the Source metadata step provides a list of fields, along with the data type, format, string length, etc. for the CSV file. Here we connect that to the CSV file input step in the template.

For the rename step we wire up the source field name and mapped field name.

## Parameterization

This data workflow accepts parameters for the customer and file type.

When I run this workflow I specify the customer and file type I want to ingest. The workflow will calculate the path to the file, the path for the output in HDFS and will query tables containing the source and target metadata for the specified customer and file type.

## User Interface

Now all that’s left is to provide a UI to customers to upload their file and specify the metadata.

That could look something like this. (show web form)

Here’s a very basic web form I created that allows me to upload a file and enter metadata about the file.

Let’s look at a sample departments file that I want to upload. (onboard/sample\_data/AcmeDepts.csv)

It’s got no header, is semi-colon delimited, with 2 fields: a parent department and child department.

In the form I can specify:

* the type of file, for example departments or employees.
* I enter the delimiter,
* if it has a header and
* the list of fields in the file. For each field I can specify
  + data type,
  + string length,
  + date format and other metadata.
* I map each field to the target schema so that the workflow knows what to rename the field to and how to transform the data, if needed
* I select my file and hit Go,
* the form will upload the file to the server and save the metadata to the database tables that are then queried by the data workflow.
* It can also call our web service API to start the workflow passing the customer and file type as parameters on the URL.

## Running the workflow

Now that we have uploaded a file and entered metadata for Acme departments, let’s run the workflow. (t\_file2hdfs.ktr)

We enter the customer name and file type.

The workflow renders the template and saves a copy of it to the file system. (e.g. onboard/PDI/injected\_department\_Acme.ktr)

We can open this up and verify that it looks right.

## Check Output

In Hue I can see that the data has landed in HDFS and it automatically created a path partitioned on the customer.

## Run a second customer file

Let’s repeat this with a separate departments file for the customer HiTech.

This file (onboard/sample\_data/HiTechDepts.csv) has a header and is comma delimited.

Let upload the file and hit Go.

We’ll run the workflow and now we can see a new rendered template and their data in the right format in HDFS.

## Further processing

Now that the data is in a consistent format this makes it easy to further process. We can go on to perform data validation, data profiling, data quality, enrichment and a whole host of other processes. Then can all be automated and orchestrated as part of a higher level workflow.

This is a very simple example compared to what our customers are doing but it serves to show how metadata injection works.

Let me share a few real customer use cases. (switch back to the deck)