

# Reloading support for auxiliary classes in HS2

## Overview

HIVE-7553 is talking about refreshing auxiliary jars for UDF or SerDes in Hive Server2. Specific Use case is that user/administrator needs to restart entire HiveServer2 if he needs to update any jar (custom UDF, output formats ...). As HS2 is a shared service this requires scheduling a maintenance window affecting all users on the cluster and hence is very troublesome.

### Current Workflow of Auxiliary Class Loading

The main workflow in hive server2 for jars loader is as below:

1. When starting the hive server2, the bootstrap script, the bash script under the \$HIVE\_HOME/bin, reads environment variable HIVE\_AUX\_JARS\_PATH from the system and adds jar files under the directory one by one via parsing function
2. The system class loader loads the jar files in step1.
3. When trying to create a UDF based on the added auxiliary jars, class FunctionTask will try to get a class from loaded classes by calling getUdfClass method in the class FunctionTask.

Conclusively, the loading of auxiliary classes is associated with the jars of hive library.

## Class loader Features

There are some features of class loader that need to discuss first.

- a) When finding a class, it will check the parent class loader first to see whether the class is loaded and then current class loader.
- b) Class Loader did not have the native mechanism for us to reload the cached class.

## Design

Based on the current workflow of auxiliary class loading and some features of class loader, I came up with the following solutions in three catalogs. Known that, hot swap for classes in a class loader may result in in consistence of data. Luckily, the auxiliary classes are stateless which are used for UDF, Ser/Des. For this reason, replacing the previous class loader may be a decent choice.

### 1. Postpone the auxiliary class loading phase and replace class loader when reloaded

#### Overview:

As hive server2 is positioned as a service, the auxiliary class loading phase should be decoupled from the loading phase of hive library jars. Current solution is aiming at loading auxiliary classes as well as hive library jars at the starting time of hive server2. For hive

library jars, there is no doubt loading them in the AppClassLoader at the booting time. And for the auxiliary jars, we can adopt a lazy loading strategy that they are loaded by the class loader held by HiveConf. Besides that, we shall provide the user a command like "Refresh resources" to reinforce reloading the jars from the path %HIVE\_AUX\_JARS\_PATH or from the location specified in hive-site.xml file. Each time refreshing resource command processor is called; it will reset the classloader saved in the HiveConf.

**Pros:**

- It will decouple the auxiliary jars loading phase from the initializing phase of hive server2 and will be stored in the custom class loader from HiveConf other than the AppClassLoader.

**Cons:**

- Reloading jars may make the previous cached class inaccessible which can be confusing for user.

It will change the previous initializing logical of hive server2.

## 2. Customize the class loader

We can customize the class loader into a parent-last/ child-first class loader which can avoid the changes in the initializing code logics. The parent class loader is the AppClassLoader which is used for loading jars from CLASSPATH at initializing step and the child class loader is the class loader created in HiveConf static block. HiveConf will hold the custom class loader. Similar to the 1<sup>st</sup> solution, we should provide a command to allow user trigger the reloading process which means creating a new classloader with latest jar files and make the class loader held by HiveConf replaced with the newly-created one.

**Pros:**

- Do nothing on the previous initializing script for hive-server2

**Cons:**

- Customized class loader will be opposite to the standard usage.
- Do have two classes with the same full qualified class name which one is held by the parent class loader and another by the child loader.

## Reference

<http://zeroturnaround.com/rebellabs/reloading-objects-classes-classloaders/>