# AA - R to Hive (ODBC)

This page serves as a platform for connecting to hive directly from the R studio environment through an ODBC Connection.  This pattern is tailored for the purpose of exporting the master table (after all data management has been done)

## Pre-requisites for Hive

The correct version of the hive ODBC driver needs to be installed locally. Please see details below:

| Hive ODBC driver version | Bit | Link location |
|---|---|---|
| 2.6.9 | 32 or 64 depending on your operating system | https://www.cloudera.com/downloads/connectors/hive/odbc/2-6-1.html |

**NB:**

This Link takes takes you through to the landing page. The user must select the appropriate version i.e. 2.6.9

Once the driver has been downloaded :

- Install ClouderaHiveODBCxx.msi (with xx replaced with 32 or 64 depending on your operating system).

## Pre-requisites for R

The following Pre-requisites are required on the R front once the Hive ODBC driver has been installed:

*Software requirements*

| Requirement | Description | version |
|---|---|---|
| Base R | The Code engine | latest version |
| R Studio | IDE for Base R | latest version |

*R Package requirements:*

Ensure that these packages are installed before the execution of the R script below. This will ensure that the connection to Hive is successful

| Package Name |
|---|
| odbc |
| getPass |
| DBI |

**R Script: R ODBC**

```
################################## Script to Create a Connection to Hive from R
############################################3

library(odbc)
library(getPass)
library(DBI)
## Make sure that the Hive ODBC v2.6.9 is installed on your local machine
## AuthMech = 3 - local user details defined.

## specify user id in lowercase

con <- DBI::dbConnect(odbc::odbc(),
Driver = "Cloudera ODBC Driver for Apache Hive",
Host = "cloudera.sanlam.co.za",
Port = "10000",
Schema = "<desired schema>",
AuthMech = 3,
UseSASL = 1,
HiveServerType = 2,
UID = getPass::getPass("Enter username: "),
PWD = getPass::getPass("Enter the password: "))



## Showcasing the tables present within the desired database.

DBI::dbGetQuery(con, "show tables")

test_data <- DBI::dbGetQuery(con, "select * from <schema.table>")

## Close connection
DBI::dbDisconnect(con)
```