Connecting to your Database Management System on UNIX
Content

1  What's New in Connecting to your Database Management System on UNIX.................5
1.1 Document History. .......................................................... 5

2  About this Guide. .................................................................7

3  ODBC Driver Manager Setup. ..................................................8
3.1 About ODBC. ................................................................. 8
3.2 Default Setup for Oracle and Teradata........................................8
3.3 Setup the unixODBC for Other DBMS........................................8
   Downloading unixODBC.......................................................... 9
   Setting Up unixODBC for Use with Automated Analytics......................9
   Activating unixODBC in Automated Analytics..................................10
3.4 Set Up the ODBC Driver and the ODBC Connection..........................10
   Installing and Setting Up a Suitable ODBC Connection for Automated Analytics. 10

4  SAP HANA .............................................................................12
4.1 Installing Prerequisite Software..................................................12
   Unarchiving SAP HANA Client Software......................................12
   Installation of SAP HANA Client Software on 64-bit Linux with Recent Kernel. 12
   Installation of SAP HANA Client software on Linux 64 bits with Kernel Version < 2.6.32 14
   Checking SAP HANA Native Client Connectivity................................14
   Checking SAP HANA ODBC Connectivity....................................15
4.2 Setting Up ODBC Connectivity with Automated Analytics..................16
4.3 Troubleshooting SAP HANA ODBC Connectivity..............................17
   Before Setting Up SAP HANA ODBC Driver for Automated Analytics.......17
   Activating the Full Debug Mode................................................17
   Deactivating the Full Debug Mode............................................18
   Checking the Status of the Trace Mode........................................18
4.4 SAP HANA as a Data Source...................................................18

5  SAP HANA Vora. .................................................................20
5.1 About SAP HANA Vora........................................................20
5.2 Installing the Simba Spark SQL ODBC Driver.................................20
5.3 Configuring SAP HANA Vora DSN...........................................20
5.4 Setting the ODBC Behaviour Flag............................................22

6  Oracle. ..................................................................................23
6.1 About Oracle Database..........................................................23
6.2 Setting Up Data Access for Oracle .......................................................... 23

7 Spark SQL ................................................................................................ 24
7.1 About Spark SQL .................................................................................. 24
7.2 Setting Up Data Access for Spark SQL ................................................ 24
7.3 Creating Data Manipulations using Date Functions ......................... 24
   Installation in Apache Spark SQL Server ........................................... 24
   Activation in SAP BusinessObjects Predictive Analytics ................. 25
7.4 Spark SQL Restrictions ....................................................................... 26
   Managing Primary Keys ................................................................... 26

8 Hive ......................................................................................................... 28
8.1 About Hive Database .......................................................................... 28
8.2 Setting Up Data Access for Hive .......................................................... 28
8.3 Creating Data Manipulations using Date Functions ......................... 29
   Installation in Apache Hive Server .................................................... 29
   Activation in SAP BusinessObjects Predictive Analytics ................. 29
8.4 Hive Restrictions.................................................................................. 30
   Using the Code Generator ................................................................ 30
   Aggregates ....................................................................................... 30
   Time-stamped Populations ................................................................ 30
   Inserting Data in Hive ...................................................................... 31
   Managing Primary Keys ................................................................... 31
   Tuning .............................................................................................. 32

9 Teradata ................................................................................................ 35
9.1 About Teradata Database .................................................................... 35
9.2 Using Data Access for Teradata ............................................................ 35
   Installing Prerequisite Software ....................................................... 35
9.3 Using Teradata ODBC Driver ............................................................... 37
9.4.1 Conventions .................................................................................. 37
9.4.2 Installing Teradata ODBC Driver .................................................. 37
9.4.3 Setting Up Teradata ODBC Driver ............................................... 38

10 IBM DB2 V9 ........................................................................................... 40
10.1 Installing DB2 Client ........................................................................... 40
10.2 Setting Up the DB2 Client for Connection to the Server ................... 40
10.3 Setting Up DB2 ODBC Driver .............................................................. 42

11 Sybase IQ ............................................................................................... 44
11.1 Installing Sybase IQ ODBC Driver ....................................................... 44
11.2 Setting Up Sybase IQ ODBC Driver .................................................... 45

12 Netezza ................................................................................................. 46
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1</td>
<td></td>
<td>Installing and Setting Up the Netezza ODBC Driver</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>PostgreSQL</strong></td>
<td>47</td>
</tr>
<tr>
<td>13.1</td>
<td></td>
<td>About PostgreSQL ODBC Driver</td>
<td>47</td>
</tr>
<tr>
<td>13.2</td>
<td></td>
<td>Installing the PostgreSQL ODBC Driver</td>
<td>47</td>
</tr>
<tr>
<td>13.3</td>
<td></td>
<td>Setting Up PostgreSQL ODBC Driver</td>
<td>47</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Vertica</strong></td>
<td>49</td>
</tr>
<tr>
<td>14.1</td>
<td></td>
<td>Installing and Setting Up Vertica ODBC Driver</td>
<td>49</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>SQL Server</strong></td>
<td>50</td>
</tr>
<tr>
<td>15.1</td>
<td></td>
<td>About SQL Server</td>
<td>50</td>
</tr>
<tr>
<td>15.2</td>
<td></td>
<td>Installing and Setting Up Microsoft SQL Server Driver</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>In-database Modeling</strong></td>
<td>52</td>
</tr>
<tr>
<td>16.1</td>
<td></td>
<td>Native Spark Modeling</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installation of Native Spark Modeling</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connection Setup</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configure the Spark Connection</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restrictions</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Performance Tuning</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Troubleshooting</td>
<td>72</td>
</tr>
<tr>
<td>16.2</td>
<td></td>
<td>Modeling in SAP HANA</td>
<td>73</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td><strong>Special Case</strong></td>
<td>75</td>
</tr>
<tr>
<td>17.1</td>
<td></td>
<td>Installing and Setting Up unixODBC on Suse 11</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Checking the Setup of the Software Package Repository</td>
<td>75</td>
</tr>
</tbody>
</table>
1 What's New in Connecting to your Database Management System on UNIX

Links to information about the new features and documentation changes for Connecting to your Database Management System on UNIX.

SAP BusinessObjects Predictive Analytics 3.1

<table>
<thead>
<tr>
<th>What's New</th>
<th>Link to More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Vora is now supported.</td>
<td>About SAP HANA Vora [page 20]</td>
</tr>
<tr>
<td>Spark 1.5 and 1.6 are now supported.</td>
<td>About Spark SQL [page 24]</td>
</tr>
<tr>
<td>Updated TTUF versions</td>
<td>Installing Prerequisite Software [page 35]</td>
</tr>
<tr>
<td>Updated installation and configuration procedures for Native SPark Modeling</td>
<td>Native Spark Modeling [page 52]</td>
</tr>
<tr>
<td>The connection to the following databases is no longer supported:</td>
<td>Setup the unixODBC for Other DBMS [page 8]</td>
</tr>
<tr>
<td>● MySQL</td>
<td></td>
</tr>
<tr>
<td>● Greenplum</td>
<td></td>
</tr>
<tr>
<td>Updated example to set a HANA connection</td>
<td>Setting Up ODBC Connectivity with Automated Analytics [page 16]</td>
</tr>
</tbody>
</table>

1.1 Document History

<table>
<thead>
<tr>
<th>Software Version</th>
<th>Link to More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP BusinessObjects Predictive Analytics 3.0</td>
<td>Added note about the management of table and column names</td>
</tr>
<tr>
<td></td>
<td>See About this Guide [page 7]</td>
</tr>
<tr>
<td>SAP BusinessObjects Predictive Analytics 3.0</td>
<td>All types of SAP HANA views are supported including the attribute views.</td>
</tr>
<tr>
<td></td>
<td>See SAP HANA as a Data Source [page 18]</td>
</tr>
<tr>
<td>Software Version</td>
<td>Link to More Information</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>SAP BusinessObjects Predictive Analytics 3.0</td>
<td>Smart Data Access virtual tables are supported. See <a href="#">SAP HANA as a Data Source</a> [page 18]</td>
</tr>
</tbody>
</table>
| SAP BusinessObjects Predictive Analytics 3.0 | Teradata database: the platform Solaris 10X64 is no longer supported.  
  - [About Teradata Database](#) [page 35]  
  - [Using Data Access for Teradata](#) [page 35]  
  - [Using Teradata ODBC Driver](#) [page 37] |
| SAP BusinessObjects Predictive Analytics 3.0 | Following the updated of the configuration file KxProfile.sh, the connector name has been updated.  
  - [Deactivating DataAccess ODBC Driver Manager](#) [page 10]  
  - [Setting Up ODBC Connectivity with Automated Analytics](#) [page 16]  
  - [Setting Up Data Access for Oracle](#) [page 23]  
  - [Setting Up Data Access for Spark SQL](#) [page 24]  
  - [Setting Up Data Access for Hive](#) [page 28]  
  - [Setting Up Data Access for Teradata](#) [page 36] |
| SAP BusinessObjects Predictive Analytics 2.5 | Native Spark Modeling  
  See [Native Spark Modeling](#) [page 52] |
| SAP BusinessObjects Predictive Analytics 2.5 | Modeling in SAP HANA  
  See [Modeling in SAP HANA](#) [page 73] |
| SAP BusinessObjects Predictive Analytics 2.5 | Update on SAP HANA  
  [SAP HANA as a Data Source](#) [page 18] |
| SAP BusinessObjects Predictive Analytics 2.5 | Update on Spark SQL  
  See:  
  - [Setting Up Data Access for Spark SQL](#) [page 24]  
  - [Spark SQL Restrictions](#) [page 26] |
| SAP BusinessObjects Predictive Analytics 2.5 | Update on Hive  
  See:  
  - [Creating Data Manipulations using Date Functions](#) [page 29]  
  - [Hive Restrictions](#) [page 30] |
2 About this Guide

This document indicates how to connect Automated Analytics to a database management system (DBMS) by creating an ODBC connection.

Configuring steps are detailed for the following databases on UNIX OS:
- SAP HANA
- Oracle
- Spark SQL
- Hive
- Teradata
- MySQL 5
- IBM DB2
- Sybase IQ
- Netezza
- PostgreSQL
- Vertica
- SQLServer
- Greenplum
- Native Spark Modeling

All the ODBC drivers used in the examples can be purchased via their respective editors.

This document explains how to install the drivers used to create the ODBC connections when they are not delivered with the OS. It also shows how to configure them so that they suit Automated Analytics requirements.

⚠️ Caution

Automated Analytics does not use quotes around table and column names by default. So if they contain mixed cases or special characters including spaces, they can be misinterpreted by the DBMS. To change this behavior, you need to set the CaseSensitive option to "True" as explained in the ODBC Fine Tuning guide.
3 ODBC Driver Manager Setup

3.1 About ODBC

The ODBC software is based on a two-part architecture:

- An ODBC Driver Manager: this manager is in charge of dispatching ODBC calls to the right DBMS driver.
- ODBC drivers: these drivers are actually implementing ODBC calls. Each DBMS needs its own ODBC driver.

An ODBC driver cannot function without the ODBC Driver Manager and ODBC Driver Manager is common for several ODBC drivers. While an ODBC driver is supposed to be compatible with any ODBC Driver Manager, some of them are strongly bound to their preferred ODBC Driver Manager.

3.2 Default Setup for Oracle and Teradata

Data Access for Oracle and Data Access for Teradata are delivered with their own ODBC Driver Manager properly set up.

Automated Analytics can use the Teradata ODBC driver provided with the Teradata software client package. In this case, the ODBC Driver Manager delivered by Teradata is mandatory. See the related topic.

Related Information

Using Data Access for Teradata [page 35]

3.3 Setup the unixODBC for Other DBMS

The ODBC Driver Manager unixODBC has been tested and validated for the following DBMS:

- SAP HANA
- SAP HANA Vora
- postgresQL
- Sybase IQ
- DB2
- Netezza
Checking that unixODBC is Installed

The command `isql` allows you to check if the unixODBC component is installed and available for any user:

```
isql --version
unixODBC 2.2.14
```

If this command fails, you have to install the component unixODBC.

3.3.1 Downloading unixODBC

UNIX OS is often delivered without any ODBC software layer. To download the ODBC Driver Manager, go to the unixODBC website: http://www.unixodbc.org.

On Linux, usual package repositories allow to easily find a unixODBC package. Modern Linux distributions also have a package manager with a prebuilt unixODBC package that can easily be installed.

```
Note

The installation procedure depends on the version of the UNIX OS. See section Installing and Setting Up unixODBC on Suse 11.
```

```
Example

On Ubuntu 13.4, unixODBC can be downloaded and installed with this single command:

```
sudo apt-get install unixodbc
```
```
```

3.3.2 Setting Up unixODBC for Use with Automated Analytics

For technical reasons, the library libodbc.so.1 must be available in the library path so that the application can be executed.

Depending on the actual version of unixODBC, a library with this exact name may not be available. In that case, you must create a symbolic link to libodbc.so.

```
Note

Depending on the OS, unixODBC binaries may not be available and therefore unixODBC sources must be compiled. Do not hesitate to contact the support for assistance on this subject.
```

Connecting to your Database Management System on UNIX

ODBC Driver Manager Setup
On Ubuntu 13.04, use the command:

```bash
sudo ln -s /usr/lib/x86_64-libux-gnu/libodbc.so /usr/lib/x86_64-libux-gnu/libodbc.so.1
```

### 3.3.3 Activating unixODBC in Automated Analytics

The default setup of standard Automated Analytics deliveries is to use the ODBC Driver Manager embedded in DataAccess.

An additional step is needed to deactivate it and allow unixODBC Driver Manager to be used.

#### 3.3.3.1 Deactivating DataAccess ODBC Driver Manager

1. Edit the file `<Installation Dir>/KxProfile.sh`.
2. Comment the line `KXEN_ODBC_DRIVER_MANAGER=Kxen`.
3. Uncomment the line `KXEN_ODBC_DRIVER=`. Note that when you keep the value empty, the driver provided by the OS is used by default.

### 3.4 Set Up the ODBC Driver and the ODBC Connection

Each ODBC connection to a DBMS requires its corresponding ODBC driver. The ODBC connection must be declared with DBMS specific connection options in a reference file: `odbc.ini`.

Automated Analytics is delivered with a directory (`odbcconf`) containing template files for each DBMS and providing an easy way to specify the `odbc.ini` file to use.

#### 3.4.1 Installing and Setting Up a Suitable ODBC Connection for Automated Analytics

1. Install and set up a DBMS client software (depending on the DBMS, this step may be optional).
2. Install the corresponding ODBC driver.
3. Copy the file `<Installation Dir>/odbcconf/odbc.ini.<dbms>` as `odbc.ini` (in the same directory).
**Code Syntax**

```
cp odbc.ini.<dbms> odbc.ini
```

where `<dbms>` corresponds to your database management system.

4. Replace the `<xxx>` values in `odbc.ini` with the real values. All the tricky ODBC options have already been filled in the provided templates so you only have to fill some options that are easy to find. In a general manner, it is a good practice to do this operation with the help of your database administrator.
4 SAP HANA

4.1 Installing Prerequisite Software

Before setting up the connection to SAP HANA, you need to install additional software.

4.1.1 Unarchiving SAP HANA Client Software

HANA Client sp6 rev67 minimum is mandatory for Automated Analytics.

These revisions can be found in the Service Market Place (SMP) and are typically delivered as a SAP proprietary archive .sap.

These archives can be extracted with the SAP tool sapcar by using the following command:

```
sapcar -xvf <archive name>.sar
```

This command will extract all files in a folder with the same name as the archive.

4.1.2 Installation of SAP HANA Client Software on 64-bit Linux with Recent Kernel

Recent 64-bit Linux distributions require the installation of the HDB_CLIENT_LINUX_X86_64 software package.

```
cd HDB_CLIENT_LINUX_X86_64
chmod +x hdbinst hdbsetup hdbuninst instruntime/sbd
sudo ./hdbinst -a client
```

If the installation is successful, the following message appears:

```
SAP HANA Database Client installation kit detected.
SAP HANA Database Installation Manager - Client Installation 1.00.68.384084
***************************************************************************
Enter Installation Path [/usr/sap/hdbclient]:<RETURN>
Checking installation...
Preparation package 'Python Runtime'...
Preparation package 'Product Manifest'...
Preparation package 'SQLDBC'...
Preparation package 'REPOTOOLS'...
Preparation package 'Python DB API'...
Preparation package 'ODBC'...
Preparation package 'JDBC'...
Preparation package 'Client Installer'...
Installing SAP HANA Database Client to /usr/sap/hdbclient...
Installing package 'Python Runtime'...
```
Installing package 'Product Manifest' ...
Installing package 'SQLDBC' ...
Installing package 'REPOTOOLS' ...
Installing package 'Python DB API' ...
Installing package 'ODBC' ...
Installing package 'JDBC' ...
Installing package 'Client Installer' ...
Installation done

Log file written to '/var/tmp/hdb_client_2013-11-12_16.06.06/hdbinst_client.log'

If an error occurs, the SAP HANA installer does its best to display a comprehensive report about the error.

SAP HANA Database Client installation kit detected.
Installation failed
  Checking installation kit failed
  Checking package ODBC failed
    Software isn't runnable on your system:
      file magic: elf 64-bit lsb X86-64 shared object
      MAKE SYSTEM:
        architecture = X86-64
        c_runtime = GLIBC 2.11.1
        processor_features = sse sse2 cmpxchg16b
        system = Linux
        version = 2.6.32.59
        subversion = 0.7-default
      YOUR SYSTEM:
        architecture = X86-64
        c_runtime = GLIBC 2.5
        processor_features = sse sse2 cmpxchg16b
        system = Linux
        version = 2.6.18
        subversion = 8.el5
    Checking package SQLDBC failed
    Software isn't runnable on your system:
      file magic: elf 64-bit lsb X86-64 shared object
      MAKE SYSTEM:
        architecture = X86-64
        c_runtime = GLIBC 2.11.1
        processor_features = sse sse2 cmpxchg16b
        system = Linux
        version = 2.6.32.59
        subversion = 0.7-default
      YOUR SYSTEM:
        architecture = X86-64
        c_runtime = GLIBC 2.5
        processor_features = sse sse2 cmpxchg16b
        system = Linux
        version = 2.6.18
        subversion = 8.el5

In this example, the report shows that this client needs at least version 2.6.32 of the kernel and version 2.11.1 of GLIBC, whereas the current system provides only kernel 2.6.18 and GLIBC 2.5. The installed Linux version, HDB_CLIENT_LINUX_X86_64, is outdated and should be replaced by the HDB_CLIENT_LINUX_X86_64_SLES9 package.
4.1.3 Installation of SAP HANA Client software on Linux 64 bits with Kernel Version < 2.6.32

Old versions of Linux require the installation of the HDB_CLIENT_LINUX_X86_64_SLES9 software package.

In this example, the report shows that this client needs at least version 2.6.32 of the kernel and version 2.11.1 of GLIBC, whereas the current system provides only kernel 2.6.18 and GLIBC 2.5. The installed Linux version, HDB_CLIENT LINUX X86 64, is outdated and should be replaced by the HDB_CLIENT_LINUX_X86_64_SLES9 package.

If the installation is successful, the following message appears:

```
SAP HANA Database Client installation kit detected.
SAP HANA Database Installation Manager - Client Installation 1.00.68.384084
***************************************************************************
Enter Installation Path [/usr/sap/hdbclient]:
Checking installation...
Preparing package "Product Manifest"...
Preparing package "SQLDBC"...
Preparing package "ODBC"...
Preparing package "JDBC"...
Preparing package "Client Installer"...
Installing SAP HANA Database Client to /usr/sap/hdbclient...
Installing package 'Product Manifest' ... 
Installing package 'SQLDBC' ... 
Installing package 'ODBC' ... 
Installing package 'JDBC' ... 
Installing package 'Client Installer' ... 
Installation done
Log file written to '/var/tmp/hdb_client_2013-11-12_16.20.37/hdbinst_client.log'
```

4.1.4 Checking SAP HANA Native Client Connectivity

The SAP HANA Client package provides tools to test native SAP HANA connectivity without involving any other software layer, especially any ODBC layer. We strongly advise you to use these tools before going further.

```
Code Syntax

    cd /usr/sap/hdbclient
    ./hdbsql
    hdbsql> \c -i <instance number -n <host name> -u <user> -p <password>

Example

To test SAP HANA running at Hana-host as instance number 00 allowing user myName/MyPass, the commands to use are:

    cd /usr/sap/hdbclient
    ./hdbsql
    hdbsql> \c -i 00 -n Hana-host -u myName -p MyPass
    Connected to BAS@Hana-host:30015
```
4.1.4.1 Missing libaio-dev Package

On recent Linux kernels, some software packages might be missing, making it impossible to use SAP HANA Native Client Connectivity.

In such a case, when running `hdbsql`, the following error message is triggered:

```
hdbsqlerror while loading shared libraries: libaio.so
```

Install the `libaio-dev` package to solve this issue. The command to install the package depends on the Linux version but is usually:

```
sudo apt-get install libaio-dev
```

4.1.5 Checking SAP HANA ODBC Connectivity

If the previous check is successful, another SAP tool allows checking the ODBC connectivity layer without any other software layer than the SAP ones.

We strongly advise you to use these tools before going further.

**Code Syntax**

```
export LD_LIBRARY_PATH=/usr/sap/hdbclient:$LD_LIBRARY_PATH
/usr/sap/hdbclient/odbcreg <Servernode> <ServerDB> <UID> <PWD>
```

**Example**

For SAP HANA running at Hana-host as instance number 00 allowing user MyName/MyPass on database BAS, the commands to use are:

```
/usr/sap/hdbclient/odbcreg Hana-host:30015 KXN MyName MyPass
ODBC Driver test.
Connect string: 'SERVERNODE=Hana-host:
30015;SERVERDB=BAS;UID=MyName;PWD=MyPass;'.
retcode: 0
outString(68): SERVERNODE={Hana-host:
30015};SERVERDB=BAS;UID=MyName;PWD=MyPass;
Driver version SAP HDB 1.00 (2013-10-15).
Select now(): 2013-11-12 15:44:55.272000000 (29)
```

The last line displays the current time on the HANA server using SQL with ODBC connectivity and allows the full validation of SAP HANA ODBC connectivity to the given HANA server.
4.2 Setting Up ODBC Connectivity with Automated Analytics

1. Check that unixODBC is properly installed.

   **Note**
   
   The unixODBC component is mandatory. The exact process to install this package, if missing, depends on the exact version of UNIX. See the related topic on setting up ODBC on Suse 11.

2. Set up the application so it uses unixODBC provided by the OS.

3. Edit the KxProfile.sh file located in the Automated Analytics installation folder.

4. Uncomment the line #KXEN_ODBC_DRIVER=so the file ends with:

   ```
   #KXEN_ODBC_DRIVER_MANAGER=Kxen
   #KXEN_ODBC_DRIVER_MANAGER=tdodbc
   KXEN_ODBC_DRIVER_MANAGER=
   fi
   export KXEN_ODBC_DRIVER_MANAGER
   # -- Define here additional variables required for all InfiniteInsight instances...
   #Force our unixODBC Driver Manager
   ```

5. Copy the file `<installation folder>/odbconfig/odbc.ini.HANA` to `<installation folder>/odbconfig/odbc.ini` and edit the actual values.

**Example**

The following `odbc.ini` file describes a connection name `MyHANA12=HANA sp12` that allows the connection to the HANA host named `Hana-host` running instance `00` with the typical path for the HANA ODBC driver.

```
[ODBC Data Sources]
MyHANA12=HANA sp12
[MyHANA12]
Driver=HDBODBC
ServerNode=Hana-host:30015
```

**Related Information**

- Setup the unixODBC for Other DBMS [page 8]
- Installing and Setting Up unixODBC on Suse 11 [page 75]
4.3 Troubleshooting SAP HANA ODBC Connectivity

SAP provides some tools to set up trace and debug information of SAP HANA ODBC Driver.

⚠️ Caution

Such debug and trace information is costly to generate and must be activated only for debug and tests purposes. It is important to switch back to regular configuration when debugging is done.

4.3.1 Before Setting Up SAP HANA ODBC Driver for Automated Analytics

If previous checks are successful, SAP HANA ODBC connectivity can be set up for the application.

The application GUI displays the list of available ODBC connections and therefore allows not typing any connection name. Should you choose to type yourself the connection name or use the command line tool KxShell, be sure to use exactly the same upper/lower cases as declared in odbc.ini. Otherwise, SAP HANA ODBC driver will not be able to connect. This behavior is specific to the SAP HANA ODBC Client.

💡 Example

If MyHana6 is described in odbc.ini, using the connection name MYHANA6 will display an error message with the following ODBC diagnostic:

```
Connection failed: [08S01][SAP AG][LIBODBCHDB SO][HDBODBC] Communication link failure;-10709 Connect failed (no reachable host left).
```

4.3.2 Activating the Full Debug Mode

- Enter the following commands.

```
cd /usr/sap/hdbclient
./hdbodbc_cons CONFIG TRACE PACKET ON
./hdbodbc_cons CONFIG TRACE DEBUG ON
./hdbodbc_cons CONFIG TRACE API ON
./hdbodbc_cons CONFIG TRACE SQL ON
./hdbodbc_cons CONFIG TRACE FILENAME <filename>
```

After this sequence of commands, any usage of the SAP HANA ODBC driver will fill comprehensive debug and trace information in the file called <filename>.
4.3.3 Deactivating the Full Debug Mode

- Enter the following commands.

```
cd /usr/sap/hdbclient
./hdbodbc_cons CONFIG TRACE PACKET OFF
./hdbodbc_cons CONFIG TRACE DEBUG OFF
./hdbodbc_cons CONFIG TRACE API OFF
./hdbodbc_cons CONFIG TRACE SQL OFF
```

4.3.4 Checking the Status of the Trace Mode

- Enter the following commands.

```
hd /usr/sap/hdbclient
hdbodbc_cons SHOW ALL
```

4.4 SAP HANA as a Data Source

You can use SAP HANA databases as data sources for all types of modeling analyses in Modeler: Classification/Regression, Clustering, Time Series, Association Rules, Social, and Recommendation.

<table>
<thead>
<tr>
<th>SAP HANA tables or SQL views</th>
<th>found in the Catalog node of the SAP HANA database</th>
</tr>
</thead>
<tbody>
<tr>
<td>All types of SAP HANA views</td>
<td>found in the Content node of the SAP HANA database. An SAP HANA view is a predefined virtual grouping of table columns that enables data access for a particular business requirement. Views are specific to the type of tables that are included, and to the type of calculations that are applied to columns. For example, an analytic view is built on a fact table and associated attribute views. A calculation view executes a function on columns when the view is accessed.</td>
</tr>
</tbody>
</table>

**Restriction**

- Analytic and calculation views that use the variable mapping feature (available starting with SAP HANA SPS 09) are not supported.
- You cannot edit data in SAP HANA views using Automated Analytics.
- SAP HANA views are not supported in Data Manager.
Smart Data Access virtual tables

Thanks to Smart Data Access, you can expose data from remote sources tables as virtual tables and combine them with HANA regular tables. This allows you to access data sources that are not natively supported by the application, or to combine data from multiple heterogeneous sources.

⚠️ Caution

To use virtual tables as input data sets for training or applying a model or as output data sets for applying a model, you need to check that the following conditions are met:

- The in-database application mode is not used.
- The destination table for storing the predicted values exists in the remote source before applying the model.
- The structure of the remote table, that is the column names and types, must match exactly what is expected with respect to the generation options; if this is not the case an error will occur.

⚠️ Caution

In Data Manager, use virtual tables with caution as the generated queries can be complex. Smart Data Access may not be able to delegate much of the processing to the underlying source depending on the source capabilities. This can impact performance.

Prerequisites

You must know the ODBC source name and the connection information for your SAP HANA database. For more information, contact your SAP HANA administrator.

In addition to having the authorizations required for querying the SAP HANA view, you need to be granted the `SELECT` privilege on the `_SYS_BI` schema, which contains metadata on views. Please refer to SAP HANA guides for detailed information on security aspects.
5 SAP HANA Vora

5.1 About SAP HANA Vora

Native Spark Modeling optimizes the modeling process on SAP HANA Vora data sources.

To setup an ODBC connection to SAP HANA Vora you need to follow the steps listed below:

1. Install the Simba Spark SQL ODBC driver.
2. Configure a SAP HANA Vora DSN with the Simba Spark SQL driver.
3. Set the ODBC behaviour flag for the DSN to SAP HANA Vora.

For more information, refer to the section Native Spark Modeling in the related information below.

Related Information

Native Spark Modeling [page 52]

5.2 Installing the Simba Spark SQL ODBC Driver

The Simba Spark SQL ODBC driver can be obtained from the Simba website.

Follow the instructions to install the driver.

i Note

HortonWorks customers can use the HortonWorks Spark SQL ODBC driver provided by Simba.

5.3 Configuring SAP HANA Vora DSN

1. Install the unixODBC package corresponding to your operating system.
2. Install the Simba Spark ODBC Driver and its required dependencies for your operating system based on the shipped instructions.
3. Go to the Setup folder located in the installation folder of the Simba drivers, for example, /opt/simba/spark/Setup.
4. Copy the files listed in the following table to your home directory and rename them as specified in the table.

<table>
<thead>
<tr>
<th>File to Copy</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>odbcinst.ini</td>
<td>.odbcinst.ini</td>
</tr>
<tr>
<td>odbc.ini</td>
<td>.odbc.ini</td>
</tr>
</tbody>
</table>

5. Update the `.odbcinst.ini` file in your home directory to point to the correct driver installation path. Change the driver path to the correct installation location on your system.

```
Sample Code

.odbcinst.ini

[ODBC Drivers]
Simba Spark ODBC Driver 64-bit=Installed

[Simba Spark ODBC Driver 64-bit]
Description=Simba Spark ODBC Driver (64-bit)
Driver=/opt/simba/sparkodbc/lib/64/libsparkodbc_sb64.so
```

6. Update the `.odbc.ini` file in your home directory.
   a. Add the connection information you have obtained from your cluster administrator
   b. Change the driver path to the correct installation location on your system.

```
Sample Code

.odbc.ini

[ODBC Data Sources]
VORA1.2=Simba Spark ODBC Driver 64-bit

[VORA1.2]
Description=Simba Spark ODBC Driver (64-bit) DSN
Driver=/opt/simba/sparkodbc/lib/64/libsparkodbc_sb64.so
ErrorMessagesPath=/opt/simba/sparkodbc/ErrorMessages
HOST=host.example.com
PORT=49155
Schema=default
SparkServerType=3
AuthMech=2
ThriftTransport=1
UseNativeQuery=1
CatalogSchemaSwitch=0
GetTablesWithQuery=1
UID=username
```

7. Go to the folder `lib/64` in the driver installation directory, for example, `/opt/simba/sparkodbc/lib/64`.

8. Edit the `simba.sparkodbc.ini` file and set `ErrorMessagesPath` as shown in the example below changing the path `/opt/simba/sparkodbc` to the correct installation location on your system.

```
Sample Code

simba.sparkodbc.ini

[Driver]
```
5.4 Setting the ODBC Behaviour Flag

The ODBC Behaviour flag is required to enable SAP HANA Vora ODBC connectivity with the Simba Spark SQL ODBC driver to switch the ODBC behavior from Spark SQL to SAP HANA Vora.

1. Edit the SparkConnector\Spark.cfg file.
2. Set the Behaviour property for either a specific Vora DSN name or for all DSNs as shown in the examples below.

**Note**

For OEM installations there is no SparkConnector folder. You need to set the Behaviour flag in the KxShell.cfg configuration file.

**Example**

Spark.cfg set up for a single Vora DSN called MY_VORA_ODBC_DSN

```
# SAP HANA Vora ODBC Connectivity
# Set the "Behaviour" option for all Vora ODBC DSNs here
# example 1: for a specific DSN called MY_VORA_ODBC_DSN
ODBCStoreSqlMapper.MY_VORA_ODBC_DSN.Behaviour=Vora
```

Spark.cfg set up to use Vora behavior for all DSNs, which means that every DSN will be treated as a Vora DSN.

```
# SAP HANA Vora ODBC Connectivity
# Set the "Behaviour" option for all Vora ODBC DSNs here
# example 2: to use Vora for all DSNs
ODBCStoreSqlMapper.*.Behaviour=Vora
```
6 Oracle

6.1 About Oracle Database

Automated Analytics standard installation includes a component named Data Access for Oracle. This component allows connecting to Oracle DBMS without the need to install any additional Oracle software. On UNIX systems, this is the only component supported by Automated Analytics to connect on Oracle.

Another advantage of this component is that a special bulk mode can be activated with this driver. Using this mode, writes in Oracle DBMS are boosted by a 20 factor. Depending on the algorithms that have been used to build the model, such acceleration is mandatory when scoring. Social models are a typical example of such algorithms where scoring cannot be done with the feature In-database Apply. Note that this component is a regular Automated Analytics feature and therefore, is liable to licensing.

The driver must be set up after the installation so that it suits the application requirements. Once the driver is set up, users can start using the application.

6.2 Setting Up Data Access for Oracle

In the odbcconfig directory (located in the root directory):

1. Copy the file `odbc.ini.KxenConnectors.oracle` as `odbc.ini` (in the same directory). This can be done typing the following commands:

   ```bash
   cd <Installation dir>/odbcconfig
   cp odbc.ini.KxenConnectors.oracle odbc.ini
   ```

2. Edit the `odbc.ini` file and replace `<MyDSN>` with the name of the connection you want to use.

3. Replace all the `<xxx>` with your actual parameters. The first parameters set up the actual connectivity to your Oracle DBMS. There are numerous ways to connect to an Oracle DBMS, so you have to request assistance from your Oracle database administrator to setup the proper parameters.

   However, a very common way to define Oracle connectivity is to setup the first two parameters:
   - **HostName**: the host name of the Oracle server or its IP address.
   - **ServiceName**: the service name of the Oracle DBMS on the Oracle server.

   It is not mandatory but we advise to also update the InstallDir parameters with the full path of Automated Analytics installation as described.

4. Open the script `KxProfile.sh` located in the root directory.

5. Check that the line `#KXEN_ODBC_DRIVER_MANAGER=Kxen` is not commented.

6. The server processes can now be launched using the script `kxen.server`, located in the root directory.
7 Spark SQL

7.1 About Spark SQL

SAP BusinessObjects Predictive Analytics supports the Apache Spark framework in order to perform large-scale data processing with the Automated Predictive Server.

Note
The latest supported version of Spark SQL is 1.6.1.

7.2 Setting Up Data Access for Spark SQL

In the odbcconfig directory (located in the root directory):

1. Copy the file odbc.ini.KxenConnectors.spark as odbc.ini (in the same directory). This can be done typing the following commands:

   Code Syntax
   
   ```
   cd <Installation Dir>/odbcconfig
   cp odbc.ini.KxenConnectors.spark odbc.ini
   ```

2. Edit the odbc.ini file and replace <MyDSN> with the name of the connection you want to use.
3. Replace all the <xxx> with your actual parameters:
   - HOSTNAME: the host name of the Spark SQL Thrift Server or its IP address.
   - PORTNUMBER: the port number of the Spark SQL Thrift Server
   - DATABASE: the database to use in the Spark SQL Thrift Server
4. Open the script KxProfile.sh located in the root directory.
5. Check that the line #KXEN_ODBC_DRIVER_MANAGER=Kxen is not commented.
6. The server processes can now be launched using the script kxen.server, located in the root directory.

7.3 Creating Data Manipulations using Date Functions

SAP BusinessObjects Predictive Analytics provides UDF extensions to Spark SQL allowing the management of dates. Apache Spark SQL provides very few date functions natively.
Installation of UDFs for Spark SQL requires access to the Apache Spark SQL server.

7.3.1 Installation in Apache Spark SQL Server

The Apache Spark SQL UDFs for the application are located in:

{SAP BusinessObjects Predictive Analytics folder}/resources/KxenHiveUDF.jar

You need to copy this file into the local file system of the Apache Spark SQL server. The jar to deploy is the same used for Hive.

7.3.2 Activation in SAP BusinessObjects Predictive Analytics

In this section, <server_local_path_to_jar> designates the local path to the copied KxenHiveUDF.jar file inside the Apache Spark SQL server.

On the computer running the application, locate the configuration file (.cfg) corresponding to SAP BusinessObjects Predictive Analytics product you want to use:

- KxCORBA.cfg when using an SAP BusinessObjects Predictive Analytics server
- KJWizard.cfg when using an SAP BusinessObjects Predictive Analytics workstation

Add these lines to the proper configuration file using following syntax:

```
ODBCStoreSQLMapper.*.SQLOnConnect1="ADD JAR <server_local_path_to_jar>"
ODBCStoreSQLMapper.*.SQLOnConnect2="CREATE TEMPORARY FUNCTION KxenUDF_add_year
as 'com.kxen.udf.cUDFAdd_year'"
ODBCStoreSQLMapper.*.SQLOnConnect3= "CREATE TEMPORARY FUNCTION KxenUDF_add_month
as 'com.kxen.udf.cUDFAdd_month'"
ODBCStoreSQLMapper.*.SQLOnConnect4= "CREATE TEMPORARY FUNCTION KxenUDF_add_day
as 'com.kxen.udf.cUDFAdd_day'"
ODBCStoreSQLMapper.*.SQLOnConnect5= "CREATE TEMPORARY FUNCTION KxenUDF_add_hour
as 'com.kxen.udf.cUDFAdd_hour'"
ODBCStoreSQLMapper.*.SQLOnConnect6= "CREATE TEMPORARY FUNCTION KxenUDF_add_min
as 'com.kxen.udf.cUDFAdd_min'"
ODBCStoreSQLMapper.*.SQLOnConnect7="CREATE TEMPORARY FUNCTION KxenUDF_add_sec
as 'com.kxen.udf.cUDFAdd_sec'"
```

These lines will activate the UDFs extension for all DBMS connections. If you are using other DBMS connections than Spark SQL, you can activate the UDFs extensions only for a specific connection by replacing the star (*) with the actual name of the ODBC connection you have defined.

Example

```
ODBCStoreSQLMapper.My_DSN.SQLOnConnect1="ADD JAR <server_local_path_to_jar>"
ODBCStoreSQLMapper.My_DSN.SQLOnConnect2="CREATE TEMPORARY FUNCTION KxenUDF_add_year
as 'com.kxen.udf.cUDFAdd_year'"
ODBCStoreSQLMapper.My_DSN.SQLOnConnect3= "CREATE TEMPORARY FUNCTION KxenUDF_add_month
as 'com.kxen.udf.cUDFAdd_month'"
ODBCStoreSQLMapper.My_DSN.SQLOnConnect4= "CREATE TEMPORARY FUNCTION KxenUDF_add_day
as 'com.kxen.udf.cUDFAdd_day'"
```
7.4 Spark SQL Restrictions

Restrictions for using Spark SQL with SAP BusinessObjects Predictive Analytics.

7.4.1 Managing Primary Keys

Spark SQL does not publish the primary keys of a table.

The usual workaround is to use a description file in which the primary keys are properly described. This description can be loaded either by the user or automatically by the application. In this case, for a table XX, the application automatically reads the description file named KxDesc_XX.

Unfortunately, it may not be easy to push new files in Spark SQL and due to other limitations of Spark SQL, the application is not able to save these files in Spark SQL. It is still possible to use description files stored in a standard text repository but all descriptions must be explicitly read.

As a convenience, with Spark SQL, the application uses a new heuristic to guess the primary keys of a table. Field names are compared to patterns, allowing the detection of names commonly used for primary keys.

The default setup for the application on Spark SQL is to manage the fields listed below as primary keys:

- Starting with ‘KEY’ or ‘ID’. For example: KEY_DPTMT or IDCOMPANY
- Ending with ‘KEY’ or ‘ID’.; For example: DPTMKEY or COMPANY_ID

**Note**

The list of patterns for primary keys can be tuned by the user.

**PrimaryKeyRegExp**

This option allows the specification of a list of patterns that will be recognized as primary keys. The syntax follows the convention described in the ODBC Fine Tuning guide on SAP Help Portal at [http://help.sap.com/pa](http://help.sap.com/pa). The patterns are using the regexp formalism.

**Example**

```sql
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=KEY$
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.2=^KEY
```
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=ID$
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1:^ID

**Note**

This set of patterns is the default one for SAP BusinessObjects Predictive Analytics and Spark SQL.

### NotPrimaryRegExp

Patterns described by `PrimaryKeyRegExp` may match too many field names. This option is applied after the `PrimaryKeyRegExp` patterning and allows the explicit description of field names that are not primary keys. These patterns are using the regexp formalism.

**Example**

```
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^KEY
ODBCStoreSQLMapper.*.NotPrimaryRegExp.1=^CRICKEY
```

All field names ending by KEY will be managed as primary keys except field names ending with CRICKEY.
8 Hive

8.1 About Hive Database

Automated Analytics provides a DataDirect driver for Hive. This component lets you connect to a Hive server without the need to install any other software. Hive technology allows you to use SQL statements on top of a Hadoop system. This connectivity component supports:

- Hive 0.13, 1.01, 1.1 and 1.2.1.
- Hive server 1 and Hive server 2: The server can be set up in 2 modes. Both modes are transparently managed, however, Hive server 2 is preferred as it provides more authentication and multi-connection features.

The driver must be set up after the installation so that it suits the application requirements. Once the driver is set up, users can start using the application. Native Spark Modeling optimizes the modeling process on Hive data sources. For more information, refer to Native Spark Modeling [page 52]

8.2 Setting Up Data Access for Hive

In the odbcconfig directory (located in the root directory):

1. Copy the file odbc.ini.KxenConnectors.hive as odbc.ini (in the same directory). This can be done typing the following commands:

   ```
   cd <Installation Dir>/odbcconfig
   cp odbc.ini.KxenConnectors.hive odbc.ini
   ```

2. Edit the odbc.ini file and replace <MyDSN> with the name of the connection you want to use.
3. Replace all the <xxx> with your actual parameters:
   - **HOSTNAME**: the host name of the Hive Server or its IP address.
   - **PORTNUMBER**: the port number of the Hive Server
   - **DATABASE**: the database to use in the Hive Server
4. Open the script KxProfile.sh located in the root directory.
5. Check that the line #KXEN_ODBC_DRIVER_MANAGER=Kxen is not commented.
6. The server processes can now be launched using the script kxen.server, located in the root directory.
8.3 Creating Data Manipulations using Date Functions

SAP BusinessObjects Predictive Analytics provides UDF extensions to Hive allowing the management of dates. Apache Hive provides very few date functions natively.

**Note**
Installation of UDFs for Hive requires access to the Apache Hive server.

8.3.1 Installation in Apache Hive Server

The Apache Hive UDFs for the application are located in:

```
{SAP BusinessObjects Predictive Analytics folder}/resources/KxenHiveUDF.jar
```

You need to copy this file into the local file system of the Apache Hive server.

8.3.2 Activation in SAP BusinessObjects Predictive Analytics

In this section, `server_local_path_to_jar` designates the local path to the copied KxenHiveUDF.jar file inside the Apache Hive server.

On the computer running the application, locate the configuration file corresponding to SAP BusinessObjects Predictive Analytics product you want to use:

- KxCORBA.cfg when using an SAP BusinessObjects Predictive Analytics server
- KJWizard.cfg when using an SAP BusinessObjects Predictive Analytics workstation

Add these lines to the proper configuration file:

```
ODBCStoreSQLMapper.*/SQLOnConnect1="ADD JAR <server_local_path_to_jar>"
ODBCStoreSQLMapper.*/SQLOnConnect2="CREATE TEMPORARY FUNCTION KxenUDF_add_year as 'com.kxen.udf.cUDFAdd_year'"
ODBCStoreSQLMapper.*/SQLOnConnect3= "CREATE TEMPORARY FUNCTION KxenUDF_add_month as 'com.kxen.udf.cUDFAdd_month'"
ODBCStoreSQLMapper.*/SQLOnConnect4= "CREATE TEMPORARY FUNCTION KxenUDF_add_day as 'com.kxen.udf.cUDFAdd_day'"
ODBCStoreSQLMapper.*/SQLOnConnect5= "CREATE TEMPORARY FUNCTION KxenUDF_add_hour as 'com.kxen.udf.cUDFAdd_hour'"
ODBCStoreSQLMapper.*/SQLOnConnect6= "CREATE TEMPORARY FUNCTION KxenUDF_add_min as 'com.kxen.udf.cUDFAdd_min'"
ODBCStoreSQLMapper.*/SQLOnConnect7= "CREATE TEMPORARY FUNCTION KxenUDF_add_sec as 'com.kxen.udf.cUDFAdd_sec'"
```

These lines will activate the UDFs extension for all DBMS connections. If you are using a DBMS connections other than Hive, you can activate the UDFs extensions for a specific connection by replacing the star (*) with the actual name of the ODBC connection you have defined.

Connecting to your Database Management System on UNIX

Hive
8.4 Hive Restrictions

Hive technology is built on top of Hadoop technology and allows access to the Hadoop world with classic SQL statements. Hive’s goal is to provide a standard SQL DBMS on top of a Hadoop system. However, Hive is not yet a full SQL DBMS and has some restrictions.

8.4.1 Using the Code Generator

The code generator is compatible with Apache Hive, however, due to some restrictions of the current ODBC driver provided by DataDirect, the SQL code generated by the code generator cannot be executed when the model contains a question mark (?) as a significant category.

8.4.2 Aggregates

Since COUNT DISTINCT is not supported in analytical functions by Hadoop Hive, the application does not support the COUNT DISTINCT aggregate.

Note that aggregates with sub queries syntax are not supported by Hive.

8.4.3 Time-stamped Populations

Hive does not support functions (Union, Except, Intersection, Cross Product) that are used by the application for building Compound and Cross Product time-stamped populations. As a result, these two types of time-stamped populations cannot be used with Hive. Filtered time-stamped populations, however, can be used with Hive.
8.4.4 Inserting Data in Hive

With Hive, it is possible to create new tables and insert new records using complex statements but there is no way to push a single record using usual `INSERT INTO <table> VALUES(...).`

For example, the statement below works:

```
CREATE TABLE TODEL (ID INT)
INSERT INTO TABLE TODEL SELECT ID FROM ADULTID
```

Whereas the following statement does not work:

```
INSERT INTO TABLE TODEL VALUES (1)
```

The in-database application feature and SAP BusinessObjects Predictive Analytics code generator are not impacted by this limitation but several other features of the application are blocked:

- Scoring with models not compatible with the in-database application feature
- Saving models, data manipulations, variable pool, and descriptions in Hive DBMS
- Transferring data with the Data Toolkit
- Generating distinct values in a data set

8.4.5 Managing Primary Keys

Hive does not publish the primary keys of a table. The usual workaround with is to use a description file in which the primary keys are properly described. This description can be loaded either by the user or automatically by the application. In this case, for a table XX, the application automatically reads the description file named KxDesc_XX.

Unfortunately, it may not be easy to push new files in Hive and due to other limitations of Hive, the application is not able to save these files in Hive. It is still possible to use description files stored in a standard text repository but all descriptions must be explicitly read.

As a convenience, with Hive, the application uses a new heuristic to guess the primary keys of a table. Field names are compared to patterns, allowing the detection of names commonly used for primary keys.

The default setup for the application on Hive is to manage the fields listed below as primary keys:

- Starting with ‘KEY’ or ‘ID. For example: KEY_DPTMT or IDCOMPANY
- Ending with ‘KEY’ or ‘ID.: For example: DPTMKEY or COMPANY_ID

**Note**

The list of patterns for primary keys can be tuned by the user.
**PrimaryKeyRegExp**

This option allows the specification of a list of patterns that will be recognized as primary keys. The syntax follows the convention described in the *ODBC Fine Tuning* guide. The patterns are using the regexp formalism.

```plaintext
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=KEY$
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^KEY
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=ID$
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^ID
```

**Note**

This set of patterns is the default one for SAP BusinessObjects Predictive Analytics and Hive.

**NotPrimaryKeyRegExp**

Patterns described by `PrimaryKeyRegExp` may match too many field names. This option is applied after the `PrimaryKeyRegExp` patterning and allows the explicit description of field names that are not primary keys. These patterns are using the regexp formalism.

```plaintext
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^KEY
ODBCStoreSQLMapper.*.NotPrimaryKeyRegExp.1=^CRICKEY
```

All field names ending by KEY will be managed as primary keys except field names ending with CRICKEY.

---

**8.4.6 Tuning**

You can tune the Hive connection by disabling the KxDesc mechanism or by modifying the heap size of the JVM or Apache Hadoop client application.

**PrimaryKeyRegExp**

This option allows the specification of a list of patterns that will be recognized as primary keys. The syntax follows the convention described in the *ODBC Fine Tuning* guide. The patterns are using the regexp formalism.

```plaintext
ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=KEY$
```
**ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^KEY**
**ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=ID$**
**ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^ID**

---

**Note**

This set of patterns is the default one for SAP BusinessObjects Predictive Analytics and Hive.

---

**NotPrimaryKeyRegExp**

Patterns described by *PrimaryKeyRegExp* may match too many field names. This option is applied after the *PrimaryKeyRegExp* patterning and allows the explicit description of field names that are not primary keys. These patterns are using the regexp formalism.

---

**Example**

**ODBCStoreSQLMapper.*.PrimaryKeyRegExp.1=^KEY**
**ODBCStoreSQLMapper.*.NotPrimaryKeyRegExp.1=^CRICKEY**

All field names ending by KEY will be managed as primary keys except field names ending with CRICKEY.

---

**SupportKxDesc**

Even if pushing such description files in Hive may not be easy, the automatic KxDesc_XXX is a convenient feature. This is why the application still provides it on Hive. However, accessing Hive’s metadata may be heavy and may uselessly slow the processes down.

The **SupportKxDesc** option allows the deactivation of the KxDesc mechanism and thus possibly speeds up usage with Hive.

---

**Example**

**ODBCStoreSQLMapper.*.SupportKxDesc=false**

---

**JVM Heap Size**

When using Hive server 2 with a wide dataset, you need to make sure to increase the heap size of the Java Virtual Machine (JVM) for the Hive Metastore service.
Apache Hadoop Client Application Heap Size

The out-of-the-box Apache Hadoop installation sets the heap size for the client applications to 512 MB. Leaving the memory size at its default value can cause out-of-memory errors when accessing large tables. To avoid this, you need to increase the heap size to 2 GB by changing the ‘HADOOP_CLIENT_OPTS’ variable setting within the usr/local/hadoop/hadoop-env.sh script.
9 Teradata

9.1 About Teradata Database

There are two ways to connect to a Teradata database with Automated Analytics.

If there is no plan to use FastWrite, the recommended solution is to set up Data Access (refer to section Using Data Access for Teradata); however when this feature cannot be implemented (due to the company IT policy for example), you can skip to the Teradata ODBC driver installation (refer to section Using Teradata ODBC Driver).

9.2 Using Data Access for Teradata

Automated Analytics standard installation includes a component named Data Access for Teradata.

This component allows connecting easily to Teradata DBMS. You do not need to install a Teradata ODBC Driver. Note that this component is a regular Automated Analytics feature and therefore, is liable to licensing.

9.2.1 Installing Prerequisite Software

To use Data Access for Teradata, you need to install the following Teradata client packages first:

- tdicu
- TeraGSS
- cliv2

These components are part of the Teradata Tools and Utility Files and are commonly installed on any computer that needs accessing a Teradata server. Data Access for Teradata is compatible with TTUF13.1 to TTUF15.10.

9.2.1.1 Installing these Packages (if Needed)

Note

We recommend installing the latest patches from Teradata. These patches must be installed after the CD installation. They are available in the client patches section of Teradata website, after connecting to your user account.
1. Insert the TTUF (Teradata Tools and Utilities Files) CD that contains the correct Teradata Client Software to install. When inserted, the TTUF CD will allow selecting the Teradata package to install. The TTUF CD to use depends on the Teradata DBMS version:

<table>
<thead>
<tr>
<th>Teradata DBMS version</th>
<th>TTUF version</th>
</tr>
</thead>
<tbody>
<tr>
<td>V15.10</td>
<td>TTUF 15.10</td>
</tr>
<tr>
<td>V15.0</td>
<td>TTUF 15.0</td>
</tr>
<tr>
<td>V14.1</td>
<td>TTUF 14.10</td>
</tr>
<tr>
<td>V14.0</td>
<td>TTUF 14.0</td>
</tr>
<tr>
<td>V13.1</td>
<td>TTUF 13.1</td>
</tr>
</tbody>
</table>

2. Select cliv2, which is the only top-level package needed for Automated Analytics. The dependent packages, teraGSS and tdicu, will be automatically selected.

### 9.2.1.2 Setting Up Data Access for Teradata

In the `odbcconfig` directory (located in the root directory):

1. Copy the file `odbc.ini.KxenConnectors.teradata` as `odbc.ini` (in the same directory). This can be done typing the following commands:

   ```
   cd <Installation Dir>/odbcconfig
   cp odbc.ini.KxenConnectors.teradata odbc.ini
   ```

2. Edit the `odbc.ini` file and replace `<MyDSN>` with the name of the connection you want to use.

3. Replace all the `<xxx>` with your actual parameters:
   - `DBCNAME`: the host name of the Teradata Server or its IP address.
   - `DATABASE`: the database to use in the Teradata Server

4. Open the script `KxProfile.sh` located in the root directory.

5. Check that the line `#KXEN_ODBC_DRIVER_MANAGER=Kxen` is not commented.

6. The server processes can now be launched using the script `kxen.server`, located in the root directory.
9.3 Using Teradata ODBC Driver

When Data Access for Teradata is not available or if Teradata ODBC driver is preferred for any reason, the application supports the standard Teradata ODBC driver.

9.4.1 Conventions

To facilitate the reading, naming conventions are used in this section.

They are presented in the following table:

<table>
<thead>
<tr>
<th>The acronym...</th>
<th>Refers to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Installation dir&gt;</td>
<td>the installation directory of Automated Analytics.</td>
</tr>
<tr>
<td>&lt;Tdodbc dir&gt;</td>
<td>the installation directory of Teradata ODBC package. Usually /usr/odbc.</td>
</tr>
</tbody>
</table>

9.4.2 Installing Teradata ODBC Driver

**Recommendation**

We recommend installing the latest patches from Teradata. These patches must be installed after the CD installation. They are available in the client patches section of the Teradata website, after connecting to your user account.

1. Insert the TTUF (Teradata Tools and Utilities Files) CD, which contains the Teradata Client Software to install. When inserted, the TTUF CD will allow selecting the Teradata package to install.

<table>
<thead>
<tr>
<th>Teradata DBMS version</th>
<th>TTUF version</th>
</tr>
</thead>
<tbody>
<tr>
<td>V15.1</td>
<td>TTUF 15.1</td>
</tr>
<tr>
<td>V15.0</td>
<td>TTUF 15.0</td>
</tr>
<tr>
<td>V14.1</td>
<td>TTUF 14.1</td>
</tr>
<tr>
<td>V14.0</td>
<td>TTUF 14.0</td>
</tr>
<tr>
<td>V13.1</td>
<td>TTUF 13.1</td>
</tr>
</tbody>
</table>

2. Select tdodbc, which is the only top-level package needed for Automated Analytics. The dependent packages, teraGSS and tdicu, will be automatically selected.

3. Deactivate UnixODBC and activate the Teradata ODBC Driver manager.
   
   For any platforms except AIX:
9.4.3 Setting Up Teradata ODBC Driver

1. To specify the Teradata ODBC libraries directory (for example, `/usr/odbc/lib`), configure the environment variable corresponding to your operating system as listed in the following table.

<table>
<thead>
<tr>
<th>For the Operating System...</th>
<th>Set the Environment Variable...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SunOS</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>Linux</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>AIX</td>
<td>LIBPATH</td>
</tr>
</tbody>
</table>

To configure these variables, edit the script `KxProfile.sh`, located in the root directory, as shown in the following example.

```bash
LD_LIBRARY_PATH=/usr/odbc/lib
export LD_LIBRARY_PATH
```

2. In the directory `odbcconfig` (located in the root directory), copy the file `odbc.ini.teradata` as `odbc.ini` (in the same directory) by typing the following commands:

```bash
cd <Installation dir>/odbcconfig
```
3. Edit the `odbc.ini` file.

4. Replace all the `<xxx>` with the actual parameters. The main parameters to modify are:
   ○ `Driver`: the exact path to the file `tdata.so` installed by Teradata.
   ○ `DBCName`: the host name of the Teradata server or its IP address.
   ○ `Database & DefaultDatabase`: the current Teradata database if the default one associated to the current user is not correct.

5. Edit the script `KxProfile.sh` located in the root directory.

6. Uncomment the line `#KXEN_ODBC_DRIVER_MANAGER=tdodbc`.

7. The server processes can now be launched using the script `kxen.server`, located in the root directory.
10 IBM DB2 V9

10.1 Installing DB2 Client

1. Type the following command lines:

```
$ gunzip v9fp7_linuxx64_client.tar.gz
$ tar xvf v9fp7_linuxx64_client.tar
$ cd client
$ su root
# ./db2setup
```

The setup window opens.

2. Click **Install a Product**.

3. Click **Install New**.

4. In the **Installation type** menu, check **Custom** and click **Next**.

   The menus allowing customizing the installation are activated.

5. In the **Installation action** menu, check **Install IBM Data Server Client** and click **Next**.

6. In the **Installation action** menu, select the items shown below and click **Next**.

7. In the **Instance setup** menu, check **Create a DB2 instance** and click **Next**.

8. As a **New user**, enter `db2inst1` as **User name** and `db2grp1` as **Group name**.

9. Click **Next**.

   The last panel is displayed summarizing the installation parameters.

10. Click **Finish** to begin copying files and complete the installation procedure.

10.2 Setting Up the DB2 Client for Connection to the Server

1. Catalog the DB2 server by typing the following commands:

```
$ su - /port
$ cd /opt/ibm/db2/V9.1/bin
$ db2
> catalog tcpip node <logical name of DBMS> remote <DBMS server hostname> server 50000
```

```
DB20000I  The CATALOG TCPIP NODE command completed successfully.
```

```
DB21056W  Directory changes may not be effective until the directory cache is refreshed.
```
2. To catalog a specific database for the DBMS, type the following command:

```
db2 => catalog database <database name> at node <logical name of DBMS>
```

**Output Code**

DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is refreshed.

3. Commit and publish the parameters previously set by typing the following commands:

```
db2 => db2stop
```

**Output Code**

DB20000I  The DB2STOP command completed successfully.

```
db2 => db2start
```

**Output Code**

DB20000I  The DB2START command completed successfully.

4. Test the connectivity by typing the following command:

```
db2 => quit
```

**Output Code**

DB20000I  The QUIT command completed successfully.

```
$ db2
```

```
db2 => connect to <database name> user <user name> using <password>
```

**Output Code**

Database Connection Information  
Database server = DB2/NT 9.5.0  
SQL authorization ID = <user name>  
Local database alias = <database name>  
select * from  
db2 => list tables  
db2 => LIST NODE DIRECTORY  
Node Directory  
Number of entries in the directory = 1  
Node 1 entry:  
Node name = <logical name of DBMS>  
Comment =  
Directory entry type = LOCAL  
Protocol = TCPIP  
Hostname = <DBMS server hostname>  
Service name = 50000
10.3 Setting Up DB2 ODBC Driver

The driver must be configured to suit Automated Analytics requirements and use the application.

1. **Open the script** `KxLauncher.sh` **located in the root directory**, which is used to start Automated Analytics processes.

2. **Configure the environment variable** `<DB2INSTANCE>` **with the name of your database instance**.

   ```
   Sample Code
   
   DB2INSTANCE=port
   export DB2INSTANCE
   
   Where `port` is the name of the database instance, for example `db2inst1`.
   ```

3. **Edit the** `.odbc.ini` **file**.

4. **Add the following lines**:

   ```
   $ cat $ODBCINI
   [ODBC Data Sources]
   [IBM DB2 ODBC Driver]
   Description = DB2 on your server
   Driver =/opt/ibm/db2/V9.1/lib64/libdb2.so
   Host = <DBMS server hostname>
   ServerType = Windows
   FetchBufferSize = 99
   UserName =
   Password =
   Database = <database name>
   ServerOptions =
   ConnectOptions =
   Options =
   ReadOnly = no
   ```

   **Caution**

   The DSN name must be identical to the name of your IBM DB2 database.
5. Export the .odbc.ini file in the script KxLauncher.sh. Note that for IBM DB2 UDB v7.2, it is possible to use the driver /opt/IBMdb2/V7.1/lib/libdb2_36.so.

6. You can now start the server processes.
11 Sybase IQ

11.1 Installing Sybase IQ ODBC Driver

1. Download Sybase IQ client 15.4 software on the Sybase website.

   Note
   The actual name of the installer depends on the selected version. Next steps show the example of a Sybase IQ 15.4 client network ESD #2 64 bits installer. Sybase IQ 15.2 client network is the minimal version supported.

2. Enter the following commands to decompress the installer:

   ```
   gunzip EBF20595.tgz
   tar xvf ebf20595.tar
   cd ebf20595
   ```

3. Enter the command `./setup.bin` to configure XWindows and execute the installer.

   Caution
   Depending on the location where the client installation is stored, this may require root privileges.

   The installation wizard opens.

4. Click Next.

5. Choose the installation path and click Next.

6. Select the Custom option and click Next.

7. In the panel Choose Install Set, select the options Sybase IQ Client Sybase and IQ ODBC Driver.

8. In the panel Software License Type Selection, check the Evaluate Sybase OQ Client Suite [...] option. Note that the Network client is not submitted to licensing, there is no problem with the evaluation edition.

9. Select a localized End-user license agreement and check I agree to the terms [...].

10. Click Next. A summary panel for the current installation process is displayed.

11. Check the installation information and click Install.

   At the end of the installation process, the Install Complete panel is displayed. The installation procedure is complete.

12. Click Done to close the installation wizard.
11.2 Setting Up Sybase IQ ODBC Driver

1. To specify the Sybase libraries directory (for example, /usr/SybaseIQ/IQ-15_4/lib64), configure the environment variable corresponding to your operating system as listed in the following table.

<table>
<thead>
<tr>
<th>For the Operating System...</th>
<th>Set the Environment Variable...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SunOS</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>Linux</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>AIX</td>
<td>LIBPATH</td>
</tr>
</tbody>
</table>

To configure these variables, edit the script KxProfile.sh, located in the root directory, as shown in the following example.

```
Sample Code

LD_LIBRARY_PATH= /usr/SybaseIQ/IQ-15_4/lib64
export LD_LIBRARY_PATH
```

2. In the directory odbcconfig (located in the root directory), copy the file odbc.ini.sybaseiq as odbc.ini (in the same directory) by typing the following command:

```
Code Syntax

cp odbc.ini.sybaseiq odbc.ini
```

3. Replace all the <xxx> with your actual parameters. The main parameters to change are:
   - **Driver**: the exact path to libdbodbc12.so file installed by Sybase
   - **CommLinks**: the network parameters of Sybase IQ engine (host name and port)
   - **EngineName**: the name of the Sybase IQ engine running on server described by CommLinks
   - **Database**: the actual Sybase IQ database if several database are managed by the Sybase IQ engine

4. Use the script kxen.server, located in the root directory, to launch the server processes.
12 Netezza

12.1 Installing and Setting Up the Netezza ODBC Driver

1. Install the Netezza ODBC archive provided by Netezza.
2. To specify the Netezza ODBC libraries directory configure the environment variable corresponding to your operating system as listed in the following table. For a standard installation of Netezza, the Netezza ODBC libraries directory is /usr/local/nz/lib64.

<table>
<thead>
<tr>
<th>For the Operating System...</th>
<th>Set the Environment Variable...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SunOS</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>Linux</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>AIX</td>
<td>LIBPATH</td>
</tr>
</tbody>
</table>

To configure these variables, edit the script KxProfile.sh, located in the root directory, as shown in the following example.

```bash
LD_LIBRARY_PATH=/usr/local/nz/lib64
export LD_LIBRARY_PATH
```

3. In the directory odbcconfig/ located in the root directory, copy the odbc.ini.netezza file as odbc.ini file by typing the following command:

```bash
cp odbc.ini.netezza odbc.ini
```

4. Replace all the <xxx> with your actual parameters. The main parameters to change are:
   - Driver: the exact path to libnzodbc.so file installed by Netezza. If you have executed a standard installation of Netezza without modifying any parameters, keep the default value set up in the odbc.ini file.
   - ServerName: the host name of the Netezza server or its IP address.
   - Database: the actual Netezza database.

   **Note**

   To use another odbc.ini file, set the <NZ_ODBC_INI_PATH> variable located in the KxProfile.sh file. The line containing the variable is preceded by a comment describing how to change its value.

5. The server processes can now be launched using the script kxen.server, located in the root directory.

   **Note**

   If you plan to manage data sets using foreign character sets with Netezza, comment the line LC_ALL=C in the KxProfile.sh file.
13 PostgreSQL

13.1 About PostgreSQL ODBC Driver

Automated Analytics has been tested and validated with PostgreSQL ODBC Driver 9.00.0200-1 for Red Hat Enterprise 5 64 bits.

This driver and its associated support library can be found on the PostgreSQL website (http://yum.pgrpms.org/9.0/redhat/rhel-5Server-x86_64/).

The exact files to download are:

- postgresql90-libs-9.0.4-2PGDG.rhel5.x86_64.rpm
- postgresql90-odbc-09.00.0200-1PGDG.rhel5.x86_64.rpm

13.2 Installing the PostgreSQL ODBC Driver

The driver is delivered as an rpm package.

Enter the following commands:

```
rpm -i postgresql90-libs-9.0.4-2PGDG.rhel5.x86_64.rpm
rpm -I postgresql90-odbc-09.00.0200-1PGDG.rhel5.x86_64.rpm
```

**Note**

These commands must be executed with root’s administration rights.

13.3 Setting Up PostgreSQL ODBC Driver

1. Copy the file `odbc.ini.postgresql.ini` as `odbc.ini` (in the same directory) using the following command:
   
   ```
   cd <Installation dir>/odbcconfig
   cp odbc.ini.postgresql odbc.ini
   ```

2. Edit the `odbc.ini` file and replace `<MyDSN>` with the connection name you want to use.

3. Replace all the `<xxx>` with the actual parameters. The first two parameters set up the connectivity to your PostgreSQL DBMS:
- **Servername** is the name of the computer running the PostgreSQL DBMS
- **Database** is the name of the database you need to access
14 Vertica

14.1 Installing and Setting Up Vertica ODBC Driver

1. Install the Vertica ODBC archive provided by Vertica.

2. To specify the Vertica libraries directory (for example, `/opt/vertica/lib64`), configure the environment variable corresponding to your operating system as listed in the following table.

<table>
<thead>
<tr>
<th>For the Operating System</th>
<th>Set the Environment Variable...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SunOS</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>Linux</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>AIX</td>
<td>LIBPATH</td>
</tr>
</tbody>
</table>

To configure these variables, edit the script `KxProfile.sh`, located in the root directory, as shown in the following example.

```bash
# Sample Code
LD_LIBRARY_PATH= /opt/vertica/lib64
export LD_LIBRARY_PATH
```

3. In the directory `odbcconfig` (located in the root directory), copy the `odbc.ini.vertica` file as `odbc.ini` file in the same directory. This can be done typing the following command:

```bash
cp odbc.ini.vertica odbc.ini
```

4. Replace all the `<xxx>` with your actual parameters. The main parameters to change are:
   - **Driver**: the exact path to the file `libverticaodbc_unixodbc.so` installed by Vertica.
   - **ServerName**: the host name of the Vertica server or its IP address.
   - **Database**: the actual Vertica database.

5. The server processes can now be launched using the script `kxen.server`, located in the root directory.
15  SQL Server

15.1  About SQLServer

Microsoft provides an SQLServer ODBC driver for Linux OS.

⚠️ Caution
The connection of Automated Analytics to SQLServer 2005 using this driver has not been validated. Only recent versions of SQLServer have been validated.

15.2  Installing and Setting Up Microsoft SQLServer Driver

Steps in this process assume that the Linux server can connect to the Internet. If an internet connection is not possible on the Linux server, download the unixodbc 2.3. archive (ftp://ftp.unixodbc.org/pub/unixODBC/unixODBC-2.3.0.tar.gz), and copy the file to the Linux server.

1. Download the driver for the appropriate version of Linux OS from the Microsoft web site.
2. Unpack it using the following command: tar xvfz msodbcsql-11.0.22x0.0.tar.gz.
3. Install unixODBC 2.3.

ℹ️ Note
Microsoft strongly recommends using version 2.3 of unixODBC package. Depending on the version of Linux OS, this package may be already installed. If it is missing, it can be installed from standard package repositories. However, the Microsoft installer may not be able to find it even if it exists on the machine. The Microsoft package provides a script allowing you to download, compile and install unixODBC 2.3.

⚠️ Caution
This will potentially install the new driver manager over any existing unixODBC driver manager.

a. As a super user, execute the script:

    build_dm.sh

    Or specify the archive in the arguments: ./build_dm.sh --download-url=file:///<path to archive>/t/unixODBC-2.3.0.tar.gz

b. To install the driver manager, run the command:

    cd /tmp/unixODBC.5744.19587.1964/unixODBC-2.3.0; make install
4. Install the ODBC driver using the following command:

   ./install.sh install

5. Type in "Yes" to accept the license.

   Enter YES to accept the license or anything else to terminate the installation: YES
   Checking for 64 bit Linux compatible OS ................................................. OK
   Checking required libs are installed ......................................................... OK
   unixODBC utilities (odbc_config and odbcinst) installed ......................... OK
   unixODBC Driver Manager version 2.3.0 installed .................................. OK
   unixODBC Driver Manager configuration correct ...................................... OK*
   Microsoft ODBC Driver 11 for SQL Server already installed .............. NOT FOUND
   Microsoft ODBC Driver 11 for SQL Server files copied ......................... OK
   Symbolic links for bcp and sqlcmd created ........................................... OK
   Microsoft ODBC Driver 11 for SQL Server registered ......................... INSTALLED
   Install log created at /tmp/msodbcsql. 5744.19587.1965/install.log.
   One or more steps may have an *. See README for more information regarding
   these steps.

6. In the directory odbcconfig (located in the root directory), copy the odbc.ini.SQLServer file as
   odbc.ini file in the same directory by typing the following command:

   cp odbc.ini.sqlserver odbc.ini

7. Replace all the <xxx> with your actual parameters. The main parameters to change are:

   ○ **Driver**: the exact path to the driver library (usually /opt/Microsoft/msodbcsql/
     libmsodbcsql-11.0.so.2270.0).
   ○ **Server**: the host name of the SQLServer server or its IP address.
   ○ **Database**: the actual SQLServer database.

8. The server processes can now be launched using the script kxen.server, located in the root directory.
16 In-database Modeling

16.1 Native Spark Modeling

You use the Native Spark Modeling feature of SAP BusinessObjects Predictive Analytics to delegate the data-intensive modeling steps to Apache Spark.

Prerequisite: You have installed the recommended version of Hive as Native Spark Modeling is enabled by default for Hive connections. Refer to the section About Hive Database [page 28]

Native Spark Modeling uses a native Spark application developed in the Scala programming language. This native Spark Scala approach ensures that the application can leverage the full benefits offered by Spark including parallel execution, in-memory processing, data caching, resilience, and integration into the Hadoop landscape.

Native Spark Modeling improves:

- Data transfer: as the data intensive processing is done close to the data, this reduces the data transfer back to the SAP BusinessObjects Predictive Analytics server or desktop.
- Modeling performance: the distributed computing power of Apache Spark improves the performance.
- Scalability: the training process evolves with the size of the Hadoop cluster and so enables more model trainings to be completed in the same time and with bigger/wider datasets. It is optimized for more columns in the training dataset.
- Transparency: the delegation of the modeling process to Apache Spark is transparent to the end user and uses the same automated modeling process and familiar interfaces as before.

Note

A more general term, "In-database Modeling", refers to a similar approach that delegates the data processing steps to a database. The term "In-database Modeling" is sometimes used in the Automated Analytics configuration and messages to refer to this broader approach.

16.1.1 Installation of Native Spark Modeling

The feature Native Spark Modeling in SAP BusinessObjects Predictive Analytics requires the following four main components of Hadoop:

- Core Hadoop
- YARN: The YARN resource manager helps control and monitor the execution of the Native Spark Modeling Spark application on the cluster. For more information, refer to the section Configure the YARN Port [page 70].
- Hive: A data warehouse based on data distributed in Hadoop Distributed File System (HDFS). For more information, refer to the section Setting Up Data Access for Hive [page 28].
- Spark: A data processing framework.
As a prerequisite, you need to download the artefacts required to run Spark as they are not packaged with the SAP BusinessObjects Predictive Analytics installer.

**Caution**

Any elements manually added through the installation process need to be manually removed on uninstallation.

### 16.1.1.1 Hadoop platform

The installation process differs depending on the operating system and choice of Hive or SAP HANA Vora as a data source:

- Hive is supported on HortonWorks and Cloudera clusters with Spark version 1.6.1
- SAP HANA Vora 1.2 is supported on HortonWorks and Cloudera clusters with Spark version 1.5.2 on the UNIX operating system only. In a client-server installation, the client can be on Windows with the server on UNIX.
16.1.1.2 Recommendation for Automated Analytics Server Location

Native Spark Modeling shares the processing between the Automated Analytics Server or Workstation and the Hadoop cluster in an interactive Spark session - this is called YARN client mode. In this mode, the driver application containing the SparkContext coordinates the remote executors that run the tasks assigned by it. The recommended approach for performance and scalability is to co-locate the driver application within the cluster.

Note
You cannot run Native Spark Modeling against multiple versions of Spark at the same time on the same installation of Automated Analytics. As a workaround you can change the configuration to switch between versions.

Note
For performance and scalability install the Automated Analytics Server on a jumpbox, edge node or gateway machine co-located with the worker nodes on the cluster.

A Hadoop cluster involves a very large number of similar computers that can be considered as four types of machines:

- Cluster provisioning system with Ambari (for HortonWorks) or Cloudera Manager installed.
- Master cluster nodes that contain systems such as HDFS NameNodes and central cluster management tools (such as the YARN resource manager and ZooKeeper servers).
- Worker nodes that do the actual computing and contain HDFS data.
- Jump boxes, edge nodes, or gateway machines that contain only client components. These machines allow users to start their jobs from the cluster.

We recommend that you install Automated Analytics server on a jump box to get the following benefits:

- Reduction in latency:
  The recommendation for Spark applications using YARN client mode is to co-locate the client (in this case the Automated Analytics server) with the worker machines.

- Consistent configuration:
  A jump box contains a Spark client and Hive client installations managed by the cluster manager web interface (Ambari or Cloudera Manager). As Native Spark Modeling uses YARN and Hive, it requires three XML configuration files (`yarn-site.xml`, `core-site.xml` and `hive-site.xml`). A symbolic link can be used to the client XML files so they remain synchronized with any configuration changes made in cluster manager web interface.
Recommended Setup

This setup uses an Automated Analytics server co-located with the worker nodes in the cluster.

Limited Setup with Workstation Installation

* Note

This setup should only be used in a non-production environment.
Related Information

- How Spark runs on clusters
- Spark submit client mode recommendations

### 16.1.1.3 Install the Apache Spark Jar Files

Download the Apache Spark "pre-built for Hadoop 2.6 and later" version that is relevant for your enterprise Hadoop platform.

The Spark lib directory is located in the compressed Spark binary file as shown in the last column of the tables below:

<table>
<thead>
<tr>
<th>Cluster Type</th>
<th>Direct Download URL</th>
<th>Folder to Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Vora (Spark version 1.5.2)</td>
<td><a href="http://archive.apache.org/dist/spark/spark-1.5.2/spark-1.5.2-bin-hadoop2.6.tgz">http://archive.apache.org/dist/spark/spark-1.5.2/spark-1.5.2-bin-hadoop2.6.tgz</a></td>
<td>spark-1.5.2-bin-hadoop2.6\lib</td>
</tr>
<tr>
<td>Apache Hive (Spark version 1.6.1)</td>
<td><a href="http://archive.apache.org/dist/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz">http://archive.apache.org/dist/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz</a></td>
<td>spark-1.6.1-bin-hadoop2.6\lib</td>
</tr>
</tbody>
</table>

Use a file compression/uncompression utility to extract the lib folder contents from the Spark binary download to the SparkConnector/jars folder.
Only the lib folder needs to be extracted and the spark-examples jar can be removed.

Directory structure before:

- SparkConnector/jars/DROP_HERE_SPARK_ASSEMBLY.txt
- SparkConnector/jars/idbm-spark-1_5.jar
- SparkConnector/jars/idbm-spark-1_6.jar

Example

Directory structure after extracting the Spark 1.5.2 lib folder content:

- SparkConnector/jars/DROP_HERE_SPARK_ASSEMBLY.txt
- SparkConnector/jars/idbm-spark-1_5.jar
- SparkConnector/jars/idbm-spark-1_6.jar
- SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib/datanucleus-api-jdo-3.2.6.jar
- SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib/datanucleus-core-3.2.10.jar
- SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib/datanucleus-rdbms-3.2.9.jar
- SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib/spark-assembly-1.5.2-hadoop2.6.0.jar
- SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib/spark-examples-1.5.2-hadoop2.6.0.jar -> this can be removed

Example

Directory structure after extracting the Spark 1.6.1 lib folder content:

- SparkConnector/jars/DROP_HERE_SPARK_ASSEMBLY.txt
- SparkConnector/jars/idbm-spark-1_5.jar
- SparkConnector/jars/idbm-spark-1_6.jar
- SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib/datanucleus-api-jdo-3.2.6.jar
- SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib/datanucleus-core-3.2.10.jar
- SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib/datanucleus-rdbms-3.2.9.jar
- SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib/spark-assembly-1.6.1-hadoop2.6.0.jar
- SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib/spark-examples-1.6.1-hadoop2.6.0.jar -> this can be removed

16.1.1.4 Upload Spark Assembly Jar

The relatively large Spark assembly jar is uploaded to HDFS each time a Spark job is submitted. To avoid this, you can manually upload the Spark assembly jar to HDFS. Refer to the sections Edit the SparkConnections.ini File and Configure the YARN Port in the related information below to know how to specify the location of the assembly jar in the SparkConnections.ini configuration file.

1. Logon to a cluster node.
2. Put the jar on HDFS.
Example

Sample Code
To put the Spark 1.5.2 assembly jar into the /jars directory on HDFS on a using SAP HANA Vora cluster when the spark assembly jar is in the /tmp directory, use the following command:

```bash
hdfs dfs -mkdir /jars hdfs dfs -copyFromLocal /tmp/spark-assembly-1.5.2-hadoop2.6.0.jar /jars
```

Example

Sample Code
To put the Spark 1.6.1 assembly jar file into the /jars directory on HDFS on a Apache Hive cluster when the spark assembly jar is in the /tmp directory, use the following command:

```bash
hdfs dfs -mkdir /jars hdfs dfs -copyFromLocal /tmp/spark-assembly-1.6.1-hadoop2.6.0.jar /jars
```

Related Information

Configure the YARN Port [page 70]
Edit the SparkConnections.ini File [page 66]

16.1.1.5 Copy SAP HANA Vora jar from Cluster

You are connecting to SAP HANA Vora 1.2 as a data source. An additional jar file, spark-sap-datasources-1.2.33-assembly.jar, is required to connect to SAP HANA Vora 1.2.

Note

The jar file location is set in the Spark.VoraJarFile property of the Spark.cfg file. Refer to the Edit the Spark.cfg section in the related information below and the SAP HANA Vora examples provided.

1. Get the jar file using one of the following ways:
   - Extract the jar file from the SAP HANA Vora download package (tgz). It is located in the following directory: vora-base\package\lib\vora-spark\lib\spark-sap-datasources-1.2.33-assembly.jar
   - Logon to a cluster node, preferably the jumpbox.
     By default the jar file can be found at the following location:
<table>
<thead>
<tr>
<th>Hadoop vendor</th>
<th>Default Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloudera</td>
<td></td>
</tr>
</tbody>
</table>

2. Copy the jar file to the SparkConnector/jars folder.

### 16.1.2 Connection Setup

Native Spark Modeling is enabled by default for Hive connections. (For more information, refer to the section Setting Up Data Access for Hive [page 28]).

It requires specific connection configuration entries and three Hadoop client configuration files (hive-site.xml, core-site.xml and yarn-site.xml).

As a prerequisite, you have installed a server with basic configuration. Refer to the Server Installation Guide for UNIX for more information.

If you run the installation on a non-Windows jumpbox on the cluster, create symbolic links to the hive-site.xml, core-site.xml and yarn site.xml files.

If you run the installation outside the cluster, download the hive-site.xml, core-site.xml and yarn site.xml files.

#### 16.1.2.1 Download the HortonWorks Client Configuration Files from Ambari

You want to download the Hadoop client configuration files to enable the YARN connection to a HortonWorks cluster.

You have access to the Ambari web User Interface.

Download the client configuration files (hive-site.xml, core-site.xml and yarn-site.xml at a minimum) from the HortonWorks Ambari web User Interface:

1. Log on to the Ambari web User Interface.
2. Download the YARN client configuration files (including the core-site.xml and yarn-site.xml files)
   a. Go to the menu Services and click on Yarn service.
   b. On the right side, click on Service actions.
   c. From the dropdown list, select Download Client_configs.
      The downloaded file contains the core-site.xml and yarn-site.xml files.
3. Download the Hive client configuration files including the hive-site.xml file:
   a. Go to the Services menu and click on Hive service.
   b. On right side, click on Service actions.
16.1.2.2 Download the Cloudera Client Configuration Files from Cloudera Manager

You want to download the Hadoop client configuration files to enable the YARN connection to a Cloudera cluster.

Download the following client configuration files from the Cloudera Manager web User Interface:

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>hive-site.xml</td>
<td>Hive</td>
</tr>
<tr>
<td>core-site.xml</td>
<td>YARN</td>
</tr>
<tr>
<td>yarn-site.xml</td>
<td>YARN</td>
</tr>
</tbody>
</table>

1. Log on to the Cloudera Manager web User Interface.
2. Download the YARN client configuration files.
   a. From the Home menu, select the Yarn service.
   b. On right side, click on Actions.
   c. From the dropdown list, select Download Client Configuration.
      The downloaded file contains the core-site.xml and the yarn-site.xml.
3. Download the Hive client configuration files.
   a. From the Home menu, select the Hive service.
   b. On right side, click on Actions.
   c. From the dropdown list, select Download Client Configuration.

16.1.2.3 Create a Directory for the Client Configuration Files

You have downloaded the three hadoop client configuration XML files.

Create a directory to store the files:
1. Change directory to SparkConnector/HadoopConfig directory from the installation root directory.
2. Create a directory with the name of the Hive ODBC connection.
3. Copy the three hadoop client configuration files to the new directory.
4. Change the ownership of the directory and files to kxenadmin:kxenusers.

16.1.3 Configure the Spark Connection

You configure Spark on YARN connection using the two property files located under the SparkConnector directory listed in the table below.
The two tables below describe the properties available in each property file.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default/Expected Values</th>
</tr>
</thead>
</table>
| **SparkAssemblyLibFolder** | Mandatory. Location of the downloaded Spark lib folder                      | Apache Hive (default): 
  
  \"../SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib\"  
  
  SAP HANA Vora:  
  
  \"../SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib\" |
| **SparkConnectionsFile** | Mandatory. Location of the connections file                                 | \"../SparkConnector/SparkConnections.ini\" |
| **JVMPath**             | Mandatory. Location of the Java Virtual Machine used by the Spark connection. This is OS dependent. | \"../jre/lib/{OS}/server\" (default)  
  
  \"../../j2re/bin/server\" |
| **IDBMJarFile**         | Mandatory. Location of the Native Spark Modeling application jar            | Apache Hive (default):  
  
  \"../SparkConnector/jars/idbm-spark-1_6.jar\"  
  
  SAP HANA Vora:  
  
  \"../SparkConnector/jars/idbm-spark-1_5.jar\" |
| **VoraJarFile**         | Location of SAP HANA Vora data source jar file.                            | \"../SparkConnector/jars/spark-sap-databasesources-1.2.33-assembly.jar\" |
| **.native."property_name"** | A Native Spark property can be specified by adding native and quotes to the property name and value, for example: | Apache Hive (Spark 1.6.1): Refer to http://spark.apache.org/docs/1.6.1/configuration.html  
  
  SAP HANA Vora (Spark 1.5.2): Refer to http://spark.apache.org/docs/1.5.2/configuration.html |
<p>| <strong>HadoopHome</strong>          | Mandatory. Internal Hadoop Home containing a bin subdirectory. Mainly used for Windows (for winutils.exe). | &quot;../SparkConnector/&quot; |
| <strong>HadoopConfigDir</strong>     | Mandatory. Location where the Hadoop client configuration XML files is copied during modeling. | /tmp |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default/Expected Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>log4jOutputFolder</strong></td>
<td>Location of the Native Spark Modeling log output</td>
<td>“/tmp”</td>
</tr>
<tr>
<td><strong>log4jConfigurationFile</strong></td>
<td>Location of the log4j logging configuration property file</td>
<td>“SparkConnector/log4j.properties” (default)</td>
</tr>
<tr>
<td></td>
<td>Change this to control the level of logging or log recycling behavior.</td>
<td></td>
</tr>
<tr>
<td><strong>JVMPermGenSize</strong></td>
<td>The permanent generated size of the Java Virtual Machine used by the Spark connection</td>
<td>Recommended value is 256 (for 256 MB)</td>
</tr>
<tr>
<td><strong>DriverMemory</strong></td>
<td>Increase the driver memory if you encounter out-of-memory errors.</td>
<td>recommended value is 8192 (for 8192 MB).</td>
</tr>
<tr>
<td><strong>CreateHadoopConfigSubFolder</strong></td>
<td>If the value is &quot;true&quot;, a subfolder is created for each modeling process ID containing the Hadoop client configuration files in the HadoopConfigDir folder. If the value is “false” no subfolder is created and HadoopConfigDir folder is used.</td>
<td>true (default) or false.</td>
</tr>
<tr>
<td><strong>AutoPort</strong></td>
<td>Enables or disables automatic configuration of YARN ports.</td>
<td>true (default) or false</td>
</tr>
<tr>
<td></td>
<td>When the value is &quot;true” the port range can be configured with the MinPort and MaxPort properties. When it is “false” YARN automatically assigns ports. This helps when configuring a YARN connection across a network firewall.</td>
<td></td>
</tr>
<tr>
<td><strong>MinPort</strong></td>
<td>See AutoPort property</td>
<td>55000</td>
</tr>
<tr>
<td><strong>MaxPort</strong></td>
<td>See AutoPort property</td>
<td>56000</td>
</tr>
<tr>
<td><strong>AutoIDBMActivation</strong></td>
<td>Allows Native Spark Modeling to be activated when matching entries exist in the SparkConnections.ini file for the ODBC connection.</td>
<td>true (default) or false.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HadoopConfigDir</strong></td>
<td>Mandatory. Location of the Hadoop client configuration XML files (hive-site.xml, core-site.xml, yarn-site.xml)</td>
<td>../SparkConnector/hadoopConfig/$DATASOURCENAME</td>
</tr>
<tr>
<td><strong>HadoopUserName</strong></td>
<td>Mandatory. User with privilege to run Spark jobs.</td>
<td>hive</td>
</tr>
</tbody>
</table>
### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionTimeOut</td>
<td>Period (in seconds) after which a Spark job will time out. It is useful to change the value to minutes in case of troubleshooting.</td>
<td>1000000 (seconds)</td>
</tr>
<tr>
<td>.native.&quot;property_name&quot;</td>
<td>A Native Spark property can be specified by adding native and quotes to the property name and value e.g. Spark-Connection.$MyDSN.native.&quot;sparkexecutor.memory&quot;=&quot;2g&quot;.</td>
<td>Cloudera (Spark 1.5.0): Refer to <a href="http://spark.apache.org/docs/1.5.0/configuration.html">http://spark.apache.org/docs/1.5.0/configuration.html</a> †&lt;br&gt;HortonWorks (Spark 1.4.1): Refer to <a href="http://spark.apache.org/docs/1.4.1/configuration.html">http://spark.apache.org/docs/1.4.1/configuration.html</a> †.</td>
</tr>
</tbody>
</table>

### Related Information

**Edit the Spark.cfg File [page 63]**

#### 16.1.3.1 Edit the Spark.cfg File

You want to:
- change a default setting relating to all Spark connections
- connect to a SAP HANA Vora data source instead of the default, which is Apache Hive.

1. Change directory to the SparkConnector directory from the install root directory.
2. Edit the Spark.cfg file and add the necessary properties. All property names begin with Spark..
   
   Set the version of the jar files depending on the data source:
   - Hive is supported with Spark version 1.6.1
   - SAP HANA Vora 1.2 is supported with Spark version 1.5.2

   **Note**
   Most of the settings can remain at the default values.

3. For Workstation installation only: restart the client after making any changes to the Spark.cfg file.

**Example**

*Sample Code*

```
# Configuration file for SAP HANA Vora ODBC connectivity and Native Spark Modeling settings.
# See SparkConnections.ini for DSN specific settings.
```
# SAP HANA Vora ODBC Connectivity
# Set the "Behaviour" option for all Vora ODBC DSNs here
# example 1: for a specific DSN called MY_VORA_ODBC_DSN
#ODBCStoreSQLMapper.MY_VORA_ODBC_DSN.Behaviour=Vora
# example 2: to use Vora for all DSNs
#ODBCStoreSQLMapper.*.Behaviour=Vora
#
# Native Spark Modeling settings.
# Switch the properties depending on the data source type: either Hive or SAP HANA Vora 1.2
#
# ##### HIVE on HortonWorks HDP 2.4.2 or Cloudera CDH 5.8.x (Spark 1.6.1) #####
Spark.SparkAssemblyLibFolder="../SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib"
Spark.IDBMJarFile="../SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib"
#
# ##### SAP HANA VORA 1.2 on HortonWorks HDP 2.3.4 or Cloudera CDH 5.6.x (Spark 1.5.2) #####
# To enable Native Spark Modeling with SAP HANA Vora comment the
#SparkAssemblyLibFolder and IDBMJarFile property lines above
# and uncomment the lines below.
#Spark.SparkAssemblyLibFolder="../SparkConnector/jars/spark-1.5.2-bin-hadoop2.6/lib"
#Spark.IDBMJarFile="../SparkConnector/jars/idbm-spark-1_5.jar"
# local location of SAP HANA Vora data sources jar (spark-sap-datasources-1.2.33-assembly.jar), copied from the Vora cluster or a Vora service marketplace download
#Spark.VoraJarFile="../SparkConnector/jars/spark-sap-datasources-1.2.33-assembly.jar"
#
# ##### MEMORY TUNING #####
# increase the Spark Driver memory (in MB) if you encounter out of memory errors
Spark.DriverMemory=4096
#
# ##### RECOMMENDED TO KEEP TO DEFAULTS #####
# Spark JVM and classpath settings
Spark.JVMPath="/jre/lib/amd64/server"
Spark.HadoopConfigDir="/tmp"
Spark.HadoopHome="/SparkConnector/
# per connection configuration
Spark.SparkConnectionsFile="/SparkConnector/SparkConnections.ini"
Spark.AutoIDBMActivation=true
Spark.DefaultOutputSchema="/tmp"
# set permanent generation size (in MB) as spark assembly jar files contain
# a lot of classes
Spark.JVMPermGenSize=256
# location of the logging properties file
Spark.log4jConfigurationFile="/SparkConnector/log4j.properties"

Spark.cfg file when connecting to a SAP HANA Vora 1.2 data source

Sample Code

# Configuration file for SAP HANA Vora ODBC connectivity and Native Spark Modeling settings.
# See SparkConnections.ini for DSN specific settings.
# SAP HANA Vora ODBC Connectivity
#
# Set the "Behaviour" option for all Vora ODBC DSNs here
# example 1: for a specific DSN called MY_VORA_ODBC_DSN
#ODBCStoreSQLMapper.MY_VORA_ODBC_DSN.Behaviour=Vora
#ODBCStoreSQLMapper.MY_VORA_ODBC_DSN1.Behaviour=Vora
#ODBCStoreSQLMapper.MY_VORA_ODBC_DSN2.Behaviour=Vora
# example 2: to use Vora for all DSNs
#ODBCStoreSQLMapper.*.Behaviour=Vora
#
# Native Spark Modeling settings.
# Switch the properties depending on the data source type: either Hive or
# SAP HANA Vora 1.2
#
### HIVE on HortonWorks HDP 2.4.2 or Cloudera CDH 5.8.x (Spark 1.6.1) ####

#Spark.SparkAssemblyLibFolder="../SparkConnector/jars/"
#Spark.IDBMJarFile="../SparkConnector/jars/spark-1.6.1-bin-hadoop2.6/lib"
#
### SAP HANA VORA 1.2 on HortonWorks HDP 2.3.4 or Cloudera CDH 5.6.x (Spark 1.5.2) ####

# To enable Native Spark Modeling with SAP HANA Vora comment the
SparkAssemblyLibFolder and IDBMJarFile property lines above
# and uncomment the lines below.
Spark.SparkAssemblyLibFolder="../SparkConnector/jars/spark-1.5.2-bin-
hadoop2.6/lib"
Spark.IDBMJarFile="../SparkConnector/jars/idbm-spark-1_5.jar"
# local location of SAP HANA Vora data sources jar (spark-sap-
datasources=1.2.33-assembly.jar), copied from the Vora cluster or a Vora
# service marketplace download
Spark.VoraJarFile="../SparkConnector/jars/spark-sap-datasources-1.2.33-
assembly.jar"
#
### MEMORY TUNING ####
# increase the Spark Driver memory (in MB) if you encounter out of memory
errors
Spark.DriverMemory=4096
#
### RECOMMENDED TO KEEP TO DEFAULTS ####
# Spark JVM and classpath settings
Spark.JVMPath="../jre/lib/amd64/server"
Spark.HadoopConfigDir="../tmp"
Spark.HadoopHome="../SparkConnector/"
# per connection configuration
Spark.SparkConnectionsFile="../SparkConnector/SparkConnections.ini"
Spark.AutoIDBMActivation=true
Spark.DefaultOutputSchema="/tmp"
# set permanent generation size (in MB) as spark assembly jar files contain
a lot of classes
Spark.JVMPermGenSize=256
# location of the logging properties file
Spark.log4jConfigurationFile="../SparkConnector/log4j.properties"
16.1.3.2 Edit the SparkConnections.ini File

You want to configure a new Spark connection linked to a particular ODBC connection to enable Native Spark Modeling. You know the name of the ODBC connection name (DSN).

1. Change directory to the SparkConnector directory under the install root directory.
2. Edit the SparkConnections.ini file. All property names begin with SparkConnection.

### Note

The default SparkConnections.ini file does not contain any entries. Samples are provided for each supported cluster type (Cloudera and HortonWorks) and data source (Hive and SAP HANA Vora) in the SparkConnections_samples.txt. Use these as a template for creating your own configuration.

#### Default SparkConnections.ini content

```plaintext
# This file contains Spark configuration entries that are specific for each particular data source name (DSN).
#
# To enable Native Spark Modeling against a data source you need to define at least the minimum properties.
# Create a separate set of entries for each DSN. Start each entry with the text "SparkConnection" followed by the DSN name.
#
# There are 2 mandatory parameters that have to be set for each DSN -
# 1. hadoopUserName, a user name with privileges to run Spark on YARN and create Hive/Vora tables
# 2. hadoopConfigDir, the directory of the Hadoop client configuration files for the DSN (the directory with the core-site.xml, yarn-site.xml, hive-site.xml files at a minimum)
#
# It is highly recommended to upload the spark assembly jar to HDFS.
# e.g. for a DSN called MY_DSN and a Spark 1.6.1 assembly jar in the HDFS jars folder -
# SparkConnection.MY_DSN.native."spark.yarn.jar"="hdfs://hostname:8020/jars/spark-assembly-1.6.1-hadoop2.6.0.jar"
#
# It is possible to pass in native Spark configuration parameters using "native" in the property.
# e.g. to add the "spark.executor.instances=4" native Spark configuration to a DSN called MY_HIVE_DSN -
# SparkConnection.MY_HIVE_DSN-native."spark.executor.instances"="4"
#
# Specific settings for HortonWorks HDP clusters
#
# These 2 properties are also mandatory for HortonWorks clusters and need to match the HDP version exactly.
# (hint: get the correct value from the spark-defaults.conf Spark client configuration file or Ambari)
# Example for HortonWorks HDP 2.4.2 (for Hive)
# SparkConnection.MY_HDP_DSN-native."spark.yarn.am.extraJavaOptions"="-Dhdp.version=2.4.2.0-058"
# SparkConnection.MY_HDP_DSN-native."spark.driver.extraJavaOptions"="-Dhdp.version=2.4.2.0-058"
#
# Example for HortonWorks HDP 2.3.4 (for SAP HANA Vora 1.2)
# SparkConnection.MY_HDP_DSN-native."spark.yarn.am.extraJavaOptions"="-Dhdp.version=2.3.4.7-4"
# SparkConnection.MY_HDP_DSN-native."spark.driver.extraJavaOptions"="-Dhdp.version=2.3.4.7-4"
# 
```

Connecting to your Database Management System on UNIX

Customer In-database Modeling
# Specific settings for SAP HANA Vora views support

To support registering Vora tables included inside Vora views add 2 lines for the "auto-register" option
For example, for a DSN called MY_VORA_DSN and Discovery service on "hostname:8500"

```
SparkConnection.MY_VORA_DSN.native."spark.sap.autoregister"="com.sap.spark.vora"
SparkConnection.MY_VORA_DSN.native."spark.vora.discovery"="hostname:8500"
```

#, Referring to the SparkConnections_samples.txt file for sample content.

---

### Note

There are two properties specifically for SAP HANA Vora:
- `spark.sap.autoregister`
- `spark.vora.discovery`

These are required to support SAP HANA Vora views containing Vora tables as a data source.

---

### 16.1.3.2.1 Sample Content for a HortonWorks Cluster with a Hive DSN

Sample SparkConnections.ini additional content for a HortonWorks HDP 2.4.2 (Spark 1.6.1) Hive DSN named MY_HDP242_HIVE_DSN.

---

#### Sample Code

```
# Sample entries for a HortonWorks HDP 2.4.2 (Spark 1.6.1) Hive DSN with name
# MY_HDP242_HIVE_DSN
# upload the spark 1.6.1 assembly jar to HDFS and reference the HDFS location here
SparkConnection.MY_HDP242_HIVE_DSN.native."spark.yarn.jar"="hdfs://:8020/jars/spark-assembly-1.6.1-hadoop2.6.0.jar"
# hadoopConfigDir and hadoopUserName are mandatory
# hadoopConfigDir - use relative paths to the Hadoop client XML config files (yarn-site.xml, core-site.xml and hive-site.xml at a minimum)
SparkConnection.MY_HDP242_HIVE_DSN.hadoopConfigDir="../../../SparkConnector/hadoopConfig/MY_HDP242_HIVE_DSN"
SparkConnection.MY_HDP242_HIVE_DSN.hadoopUserName="hive"
# HORTONWORKS SPECIFIC: these 2 properties are also mandatory for HortonWorks clusters and need to match the HDP version exactly
# sample values for HDP 2.4.2
SparkConnection.MY_HDP242_HIVE_DSN.native."spark.yarn.am.extraJavaOptions"="-Dhdp.version=2.4.2.0-058"
SparkConnection.MY_HDP242_HIVE_DSN.native."spark.driver.extraJavaOptions"="-Dhdp.version=2.4.2.0-058"
# (optional) time out in seconds
SparkConnection.MY_HDP242_HIVE_DSN.connectionTimeOut=1000
# (optional) performance tuning
#SparkConnection.MY_HDP242_HIVE_DSN.native."spark.executor.instances"="4"
#SparkConnection.MY_HDP242_HIVE_DSN.native."spark.executor.cores"="2"
#SparkConnection.MY_HDP242_HIVE_DSN.native."spark.executor.memory"="4g"
```
16.1.3.2.2 Sample Content for a HortonWorks Cluster with SAP HANA Vora DSN

Sample SparkConnections.ini additional content for a HortonWorks HDP 2.3.4 (Spark 1.5.2) SAP HANA Vora connection with a (Simba) Spark SQL DSN named MY_HDP234_VORA_DSN.

Sample Code

```
# upload the spark 1.5.2 assembly jar to HDFS and reference the HDFS location here
SparkConnection.MY_HDP234_VORA_DSN.native."spark.yarn.jar"="hdfs://hostname:8020/jars/spark-assembly-1.5.2-hadoop2.6.0.jar"
# hadoopConfigDir and hadoopUserName are mandatory
# hadoopConfigDir - use relative paths to the Hadoop client XML config files (yarn-site.xml, core-site.xml and hive-site.xml as a minimum)
# NOTE: for Vora the hive-site.xml can be an empty file
SparkConnection.MY_HDP234_VORA_DSN.hadoopConfigDir="/../SparkConnector/hadoopConfig/MY_HDP234_VORA_DSN"
SparkConnection.MY_HDP234_VORA_DSN.hadoopUserName="spark"
# HORTONWORKS SPECIFIC: these 2 properties are also mandatory for HortonWorks clusters and need to match the HDP version exactly
# sample values for HDP 2.3.4
SparkConnection.MY_HDP234_VORA_DSN.native."spark.yarn.am.extraJavaOptions"="-Dhdp.version=2.3.4.7-4"
SparkConnection.MY_HDP234_VORA_DSN.native."spark.driver.extraJavaOptions"="-Dhdp.version=2.3.4.7-4"
# VORA SPECIFIC: add auto-register settings to support Vora views
SparkConnection.MY_HDP234_VORA_DSN.native."spark.sap.autoregister"="com.sap.spark.vora"
SparkConnection.MY_HDP234_VORA_DSN.native."spark.vora.discovery"="hostname:8500"
# (optional) time out in seconds
SparkConnection.MY_HDP234_VORA_DSN.connectionTimeOut=1000
# (optional) performance tuning
#SparkConnection.MY_HDP234_VORA_DSN.native."spark.executor.instances"="4"
#SparkConnection.MY_HDP234_VORA_DSN.native."spark.executor.cores"="2"
#SparkConnection.MY_HDP234_VORA_DSN.native."spark.executor.memory"="4g"
#SparkConnection.MY_HDP234_VORA_DSN.native."spark.driver.maxResultSize"="4g"
```

16.1.3.2.3 Sample Content for a Cloudera Cluster with Hive DSN

Sample SparkConnections.ini additional content for a Cloudera CDH 5.8.x (Spark 1.6.1) Hive DSN named MY_CDH58_HIVE_DSN.
**Sample Code**

```java
# Sample for a Cloudera CDH 5.8.x (Spark 1.6.1) Hive DSN with name
# MY_CDH58_HIVE_DSN
# upload the Spark 1.6.1 assembly jar to HDFS and reference the HDFS location
# here
SparkConnection.MY_CDH58_HIVE_DSN.native."spark.yarn.jar"="hdfs://hostname:8020/jars/spark-assembly-1.6.1-hadoop2.6.0.jar"
# hadoopConfigDir and hadoopUserName are mandatory
# hadoopConfigDir - use relative paths to the Hadoop client XML config files
# (yarn-site.xml, core-site.xml and hive-site.xml at a minimum)
SparkConnection.MY_CDH58_HIVE_DSN.hadoopConfigDir="../.../SparkConnector/hadoopConfig/MY_CDH58_HIVE_DSN"
SparkConnection.MY_CDH58_HIVE_DSN.hadoopUserName="hive"
# (optional) time out in seconds
SparkConnection.MY_CDH58_HIVE_DSN.connectionTimeOut=1000
# (optional) performance tuning
#SparkConnection.MY_CDH58_HIVE_DSN.native."spark.executor.instances"="4"
#SparkConnection.MY_CDH58_HIVE_DSN.native."spark.executor.cores"="2"
#SparkConnection.MY_CDH58_HIVE_DSN.native."spark.executor.memory"="4g"
#SparkConnection.MY_CDH58_HIVE_DSN.native."spark.driver.maxResultSize"="4g"
```

---

**16.1.3.2.4 Sample Content for a Cloudera Cluster with SAP HANA Vora DSN**

Sample SparkConnections.ini additional content for a Cloudera CDH 5.6.x (Spark 1.5.2) SAP HANA Vora connection with a (Simba) Spark SQL DSN named **MY_CDH56_VORA_DSN**.

**Sample Code**

```java
# Sample for SAP HANA Vora 1.2 on Cloudera CDH 5.6.x (Spark 1.5.2) with a
# (Simba) Spark SQL DSN with name **MY_CDH56_VORA_DSN**
# upload the spark 1.5.2 assembly jar to HDFS and reference the HDFS location
# here
SparkConnection.MY_CDH56_VORA_DSN.native."spark.yarn.jar"="hdfs://hostname:8020/jars/spark-assembly-1.5.2-hadoop2.6.0.jar"
# hadoopConfigDir and hadoopUserName are mandatory
# hadoopConfigDir - use relative paths to the Hadoop client XML config files
# (yarn-site.xml, core-site.xml and hive-site.xml at a minimum)
# NOTE: for Vora the hive-site.xml can be an empty file
SparkConnection.MY_CDH56_VORA_DSN.hadoopConfigDir="../.../SparkConnector/hadoopConfig/MY_CDH56_VORA_DSN"
SparkConnection.MY_CDH56_VORA_DSN.hadoopUserName="spark"
# VORA SPECIFIC: add auto-register settings to support Vora views
SparkConnection.MY_CDH56_VORA_DSN.native."spark.sap.autoregister"="com.sap.spark.vora"
SparkConnection.MY_CDH56_VORA_DSN.native."spark.vora.discovery"="hostname:8500"
# (optional) time out in seconds
SparkConnection.MY_CDH56_VORA_DSN.connectionTimeOut=1000
# (optional) performance tuning
#SparkConnection.MY_CDH56_VORA_DSN.native."spark.executor.instances"="4"
#SparkConnection.MY_CDH56_VORA_DSN.native."spark.executor.cores"="2"
#SparkConnection.MY_CDH56_VORA_DSN.native."spark.executor.memory"="4g"
#SparkConnection.MY_CDH56_VORA_DSN.native."spark.driver.maxResultSize"="4g"
```
16.1.3.3 Configure the YARN Port

You want to use specific ports for the Spark connection.

Native Spark Modeling uses the YARN cluster manager to communicate, deploy and execute the predictive modeling steps in Spark. The cluster manager distributes tasks throughout the compute nodes of the cluster, allocates the resources across the applications and monitors using consolidated logs and web pages. YARN is one of the main cluster managers used for Spark applications. Once connected to the cluster via YARN, Spark acquires executors on nodes in the cluster, which are processes that run the modeling steps. Next, it deploys and executes the modeling steps through a series of tasks, executed in parallel. Native Spark Modeling uses the YARN client mode to enable direct control of the driver program from Automated Analytics. It needs to communicate across the network to the Hadoop cluster to:

- Upload the SAP BusinessObjects Predictive Analytics application (a jar file).
- Initiate and monitor the Spark application remotely.

The communication across the network may be blocked by firewall rules. To enable the communication to flow through the firewall, configure the ports used by YARN:

1. Change directory to the SparkConnections directory from the install root directory.
2. Edit the SparkConnections.ini file to set specific ports for the YARN connection:

```ini
# Spark ODBC DSN = MY_HIVE
# security (see http://spark.apache.org/docs/latest/security.html)
# From Executor To Driver. Connect to application / Notify executor state changes. Akka-based. Set to "0" to choose a port randomly.
SparkConnection.MY_HIVE.native."spark.driver.port"="55300"
# From Driver To Executor. Schedule tasks. Akka-based. Set to "0" to choose a port randomly.
SparkConnection.MY_HIVE.native."spark.executor.port"="55301"
# From Executor To Driver. File server for files and jars. Jetty-based
SparkConnection.MY_HIVE.native."spark.fileserver.port"="55302"
# From Executor To Driver. HTTP Broadcast. Jetty-based. Not used by TorrentBroadcast, which sends data through the block manager instead.
SparkConnection.MY_HIVE.native."spark.broadcast.port"="55303"
# From Executor/Driver To Executor/Driver. Block Manager port. Raw socket via ServerSocketChannel
SparkConnection.MY_HIVE.native."spark.blockManager.port"="55304"
```

16.1.4 Restrictions

Thanks to Spark, the building of Automated Analytics models can run on Hadoop with better performance, higher scalability and no data transfer back to the SAP BusinessObjects Predictive Analytics server or desktop. For the models using Hadoop as a data source, the model training computations by default are delegated to the Spark engine on Hadoop whenever it is possible.

To run the training on Spark, you must fulfill the conditions described below:

- You have installed SAP BusinessObjects Predictive Analytics 2.5 or higher.
- You are connected to Hadoop data source using the Hive ODBC driver from Automated Analytics.
- You have installed the required versions of Hadoop distribution (see [Hadoop platform][page 53]), Hive (see [About Hive Database][page 28]) and Spark.
- You have installed the Apache Spark Binary and Spark Assembly Jar (see [Install the Apache Spark Jar Files][page 56]).
If the delegation to Spark is not possible or if the option is not selected, the Automated Analytics modeling engine does the training computations on a machine where SAP BusinessObjects Predictive Analytics is installed (for example, execution of the Automated Analytics algorithms).

⚠️ Restriction

As of SAP BusinessObjects Predictive Analytics 2.5 version, Native Spark Modeling is supported for the training of classification and regression models with single target. All other types of models are mainly handled by the Automated Analytics engine.

Refer to the SAP Note 2278743 for more details on restrictions.

To change the default behaviour of the system:

1. Select File Preferences or press F2.
2. On the Model Training Delegation panel, deselect the option.
3. Click OK to save your changes.

ℹ️ Note

When editing the preferences, you can restore the default settings by clicking the Reset button.

### 16.1.5 Performance Tuning

#### Split HDFS file

If the hive table is created as an external table on a HDFS directory, check if this directory contains multiple files. Each file represents a partition when this hive table is processed in Native Spark Modeling.

**Setting Spark Executor Core number - spark.executor.cores**

When running Native Spark Modeling on a cluster, do not assign all available CPU to Spark executors as this could prevent Hive from fetching the results due to a lack of resources. For instance, assign only 75% of CPU to Spark. If you have 12 cores on a node machine, set spark.executor.cores=9.

**Recommended Spark settings**

```
# Spark Executor settings
SparkConnection.MY_HIVE.native."spark.executor.instances"="4"
SparkConnection.MY_HIVE.native."spark.executor.cores"="16"
# if you use many threads => increase memory
SparkConnection.MY_HIVE.native."spark.executor.memory"="16g"
```
Install Automated Analytics server on a jump box on the cluster itself.

It is recommended to install Automated Analytics server on a jump box on the cluster itself. The memory size of the jump box must be sized to be big enough to hold the result sets generated from the Native Spark Modeling application.

**Note**
For more details, refer to the section Recommendation for Automated Analytics Server Location [page 54] in the related information below.

### 16.1.6 Troubleshooting

Troubleshootings for particular message codes can appear in the logs.

**Note**
The message codes starting with "KXEN_W" are warning messages and the processing will still continue. The message codes starting with "KXEN_I" are information messages and the processing will still continue.

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Remediation Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>KXEN_W_IDBM_UNSUPPORTED_FEATURESDetected</td>
<td>An unsupported feature was detected. The warning message will provide a more detailed description.</td>
</tr>
<tr>
<td>KXEN_W_IDBM_REGRESSION_MODELS_NOT_SUPPORTED</td>
<td>Native Spark Modeling will be &quot;off&quot; and the normal training process will continue.</td>
</tr>
<tr>
<td>KXEN_W_IDBM_INCOMPATIBLE_VARIABLE_USAGE</td>
<td>To use Native Spark Modeling for this model, either remove the affected variable(s) or avoid the feature if possible.</td>
</tr>
<tr>
<td>KXEN_W_IDBM_SPACE_ADV_PARAMS_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_INCOMPATIBLE_TRANSFORM_FEATURE_USAGE</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_USER_STRUCTURES_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_COMPOSITE_VARIABLES_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_CUSTOM_CUTTING_STRATEGY_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_CUTTING_STRATEGY_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_UNSUPPORTED_TRANSFORM</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_K2R_GAIN_CHARTS_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_RISK_MODE_K2R_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_DECISION_MODE_K2R_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_K2R_ORDER_NOT_SUPPORTED</td>
<td></td>
</tr>
<tr>
<td>KXEN_W_IDBM_MULTI_TARGETS_MODE_NOT_SUPPORTED</td>
<td></td>
</tr>
</tbody>
</table>
16.2 Modeling in SAP HANA

You can delegate the model training computations in SAP HANA if the following prerequisites are fulfilled:

- APL (version 2.4 or higher) must be installed on the SAP HANA database server.
- The minimum required version of SAP HANA is SPS 10 Database Revision 102.02 (SAP HANA 1.00.102.02).
- The APL version must be the same as the version of SAP BusinessObjects Predictive Analytics desktop or server.
- The SAP HANA user connecting to the ODBC data source must have permission to run APL.


⚠️ Restriction

Cases when model training is not delegated to APL:

- In the Recommendation and Social Analysis modules.
- When the model uses a custom cutting strategy.
- When the model uses the option to compute a decision tree.

To unselect the default behavior:

1. Select File ➔ Preferences or press F2.
2. On the Model Training Delegation panel, deselect the option.
3. Click **OK** to save your changes.
17 Special Case

17.1 Installing and Setting Up unixODBC on Suse 11

The unixODBC software package is an ODBC driver manager and is mandatory to be able to connect to a DBMS using the standard ODBC technology. Depending on the version of Linux currently used, the processes to install this package can vary. The following section describes the process to install unixODBC on Linux SUSE 11.

17.1.1 Checking the Setup of the Software Package Repository

If the operating system has been installed with the default setup, the Software Package repository is set up to use the installation DVD.

**Code Syntax**

```
sudo zypper repos
```

**Output Code**

```
<table>
<thead>
<tr>
<th>Root's password:</th>
<th>Alias</th>
<th>Enabled</th>
<th>Refresh--</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>SUSE-.....</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
```

Connecting to your Database Management System on UNIX
Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP’s gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with “you”, or a gender-neutral noun (such as “sales person” or “working days”) is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP’s gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).