



Apache Zeppelin for Denodo - User Manual

Revision 20201201

NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2020
Denodo Technologies Proprietary and Confidential

1 INTRODUCTION

Apache Zeppelin for Denodo is a web-based notebook. This customization of a standard distribution of [Apache Zeppelin](#) adds some new features that make it easier to use this tool with Denodo and offer a more integrated experience.

This application allows you to log in to a VDP server using the Denodo authentication and authorization infrastructure, and execute VQL sentences in your Zeppelin paragraphs.

This document contains a functional explanation of the Denodo-customized functionalities and screens of the application. For a complete reference on usage and all other features of Apache Zeppelin, please refer to the Apache Zeppelin website at <https://zeppelin.apache.org/>

Apache Zeppelin is developed by [The Apache Software Foundation](#).

2 INSTALLATION AND CONFIGURATION

2.1.1.1 Installing the notebook

This application can be downloaded from the [Denodo Support Site](#).

Note that this application is distributed in two different versions:

- *Apache Zeppelin for Denodo - Shared Server*
- *Apache Zeppelin for Denodo - Standalone*

In a server that is meant to be accessed by several users in your network (concurrently or not) you should use the **Shared Server version**. For security reasons this version only includes the Denodo and Markdown interpreters. If you want to use additional interpreters you should install the Standalone version, meant to be used on a local machine by a single user.

The Standalone version includes the interpreters for *angular, denodo, file, jdbc, md, python, sh* and *spark*. If you need to install more interpreters you can check the Appendix III of this manual.

Once the zip file is downloaded and unzipped, you can start the application by executing `bin/zeppelin.cmd` on Windows, or `bin/zeppelin.sh` on Linux. Please note it might take up to a minute to complete the start process.

If you are using linux you need to add execution permissions executing `chmod u+x bin/*.sh`

However, note that it is *highly recommended* to go through the rest of the “Installation and Configuration” section in this manual in order to adapt Zeppelin to your needs (e.g. connect to the right VDP server) before starting the notebook for the first time.

2.1.1.2 Configuring access to the Denodo server and database

In order to execute Denodo VQL paragraphs we have defined a customized JDBC Denodo interpreter, registered under the name `denodo`.

Important: please note that, when you start the Zeppelin application for the first time, all the configuration for all the interpreters is read from the `interpreter/{interpretername}` folder structure, and then merged together into a single `conf/interpreter.json` file (which contents can be afterwards modified by a Zeppelin admin user from the UI). This is standard Apache Zeppelin behaviour, and it makes it highly recommendable to configure your Denodo server access at the `interpreter/denodo/interpreter-setting.json` file before starting Zeppelin for the first time.

At the `interpreter/denodo/interpreter-setting.json` file there are two main configuration settings that you might need to change:

- `default.url` containing the JDBC URI to be used to connect to the desired VDP server. Default value is `jdbc:vdb://localhost:9999/?noAuth=true`
- `zeppelin.jdbc.denodo.database` containing the name of the default database to be used for paragraph execution (note that paragraphs can later specify a different database to be executed on). Please note that the VDP users used for accessing Zeppelin need to have connection privileges on the database configured here. Default value is `admin`.

After the first execution, the admin user will be responsible for making the configuration changes in the interpreter section of the application.

The screenshot shows the Zeppelin web interface for a Denodo interpreter configuration. The 'default.url' field is set to `jdbc:vdb://localhost:9999/?noAuth=true` and the 'zeppelin.jdbc.denodo.database' field is set to `admin`. A dropdown menu is open, showing options like 'Interpreter', 'Notebook Repos', 'Credential', 'Helium', 'Configuration', and 'Logout'.

name	value
common.max_count	1000
default.completer.schemaFilters	
default.completer.ttlInSeconds	120
default.driver	com.denodo.vdp.jdbc.Driver
default.password	
default.precode	
default.splitQueries	false
default.statementPrecode	
default.url	jdbc:vdb://localhost:9999/?noAuth=true
default.user	
zeppelin.jdbc.auth.type	
zeppelin.jdbc.concurrent.max_connection	10
zeppelin.jdbc.concurrent.use	true
zeppelin.jdbc.denodo.database	admin
zeppelin.jdbc.interpolation	false
zeppelin.jdbc.keytab.location	
zeppelin.jdbc.maxConnLifetime	-1

2.1.1.3 Configuring authentication

The authentication mechanisms of the Denodo-customized Zeppelin application are integrated with the VDP server being accessed from Denodo's main interpreter.

Authentication is always performed on a VDP server, and the application can use Denodo SSO, Kerberos-based or basic (user+password) authentication. By default only basic authentication is enabled. You can modify this behavior to configure the allowed authentication methods by changing the `zeppelin.denodo.login.types` property at the `conf/zeppelin-site.xml` file. Possible values for this property are BASIC, KERBEROS, DENODOSSO and ALL. If the property is not defined, all types of login are enabled. **DENODOSSO authentication is only available on Denodo 8 and later versions.**

When all authentication methods are enabled, Denodo SSO will have the priority followed by KERBEROS and BASIC. If the Denodo SSO configuration is valid zeppelin will authenticate through Denodo SSO and the configured identity provider. If Denodo SSO is not configured or the configuration is not valid, the next authentication method is Kerberos. So if your browser is configured so that it is able to obtain a valid Kerberos token from the Operating System infrastructure, the application will try to log you in using Kerberos. This will happen even if your Denodo installation is not configured for using Kerberos, so this might lead to inability to log in if you have KERBEROS or ALL as configured login type, your browser is integrated with your OSs Kerberos infrastructure, and your VDP is not.

If Zeppelin is configured to use ALL login types but your DenodoSSO application is not working the zeppelin application tries to use Kerberos and if your browser is not integrated with a Kerberos infrastructure, the application will default to basic authentication using a username and password.

To configure DenodoSSO in zeppelin you need to check the `zeppelin.denodo.login.ssoUrl` property at the `conf/zeppelin-site.xml` file. This is the URL of the Denodo SSO application, by default the value is <http://localhost:9090/sso> but maybe it is necessary to change the host and/or the port.

For the configuration of DenodoSSO it is important to configure the Denodo Security Token Server. This configuration is more detailed in the DenodoSSO documentation. The configuration files you need to check are the `SSOConfiguration.properties`, `SSOTokenConfiguration.properties` and the `tokenJavaKeyStore.jks`. It is also important to correctly configure the redirection URL's in the identity provider, make sure that the host of the sso URL configured in zeppelin is the same as in the redirection URL's configured in the identity provider. You must configure in vdp the privileges of the role of the Denodo Security Token Server. This role needs to have access to the databases that need to be accessed from Zeppelin.

Also note that the implementation of SPNEGO in Microsoft Web browsers negotiates between Kerberos and NTLM mechanisms. This means that if both browser and server support Kerberos, this will be used, but if for some reason Kerberos is not possible the browser will try NTLM. You therefore might be getting NTLM tokens sent to the Zeppelin server because the browser is unable to authenticate using Kerberos. This behaviour occurs with Chrome, Edge or Internet Explorer because it uses OS settings. In the case of Firefox, the browser has its own configuration and doesn't use NTLM when Kerberos is not configured. If the browser uses the NTLM token we will get an error from the zeppelin authentication mechanism. **This error happens with Chrome or Internet**

Explorer and the solution is to change the `zeppelin.denodo.login.types` property to **BASIC** or **KERBEROS** or use **Firefox**.

Another thing one has to bear in mind is the logout. In some browsers like firefox the basic authentication doesn't expire and if you try to log out the browser sends the authentication header again and logs in again automatically.

In the scope of this application, Zeppelin users can adopt two different roles: “admin user” and “data scientist user”. The “admin user” can modify specific configurations like create and modify interpreters. The “data scientist user” can execute notes and jobs. The admin user can also execute paragraphs and jobs. For a user to be admin, they must have the `assignprivileges` role in Denodo. This role will be assigned in the role management section of the Denodo administration tool.

2.1.1.4 Notebook storage

Apache Zeppelin has a pluggable notebook storage mechanism controlled by `zeppelin.notebook.storage` configuration option with multiple implementations.

By default, Zeppelin stores the information of the notebooks and paragraphs in json files inside the folder `/notebook`. We have decided that this information is significantly less exposed if it is stored in a derby database. For this we have added a new `NotebookRepo org.apache.zeppelin.notebook.repo.DerbyDBNotebookRepo` that will be configured in the file `/conf/zeppelin-site.xml` using the parameter `zeppelin.notebook.storage`.

```
<property>
  <name>zeppelin.notebook.storage</name>
  <value>org.apache.zeppelin.notebook.repo.DerbyDBNotebookRepo</value>
  <description>DerbyDB notebook persistence layer implementation</description>
</property>
```

All other notebook [storage implementations](#) can be used at the same time as the new `DerbyDBNotebookRepo` implementation. You only need to add them in the configuration parameter, separated by commas.

It is important to keep in mind that if other implementations are activated, the notebooks will be stored using both implementations. And therefore, the json files will continue to be exposed on the system where they are stored.

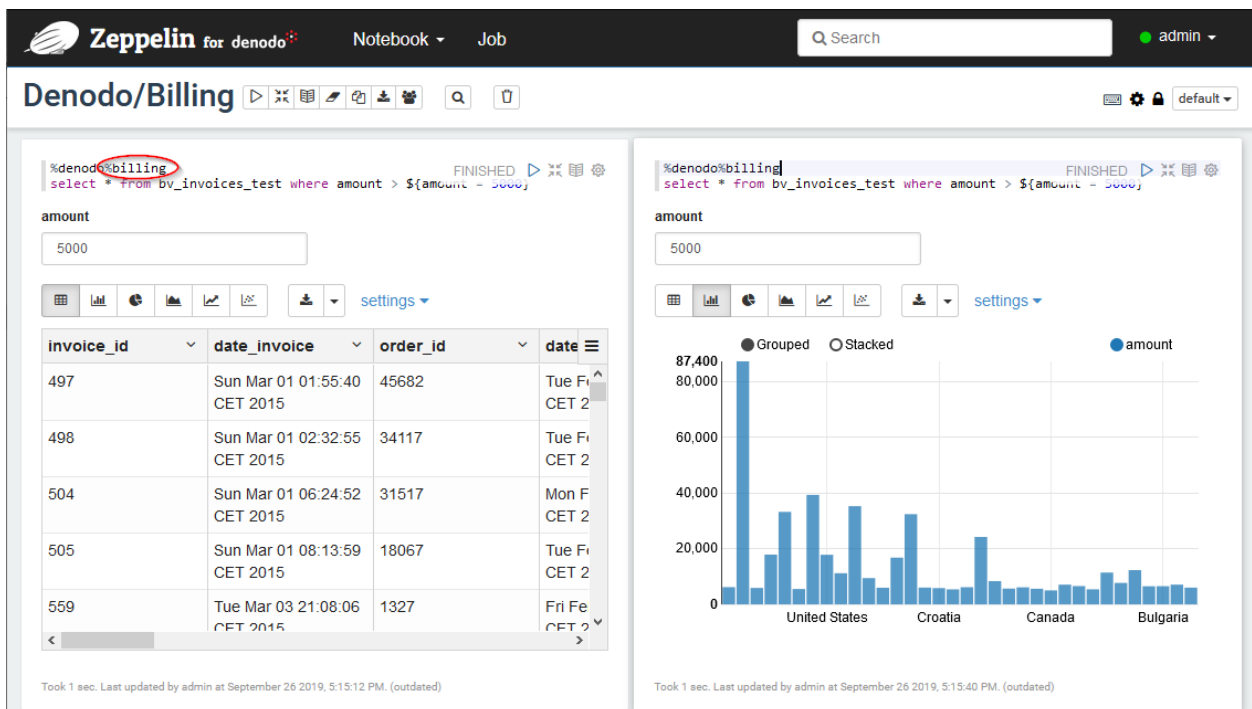
2.1.1.5 Accessing the notebook from a browser

Once started, the notebook can be accessed at <http://localhost:8080> (if needed, replace localhost with the name of the machine it has been installed on).

Modify the configuration in `conf/zeppelin-site.xml` if you need to change the port Zeppelin starts at, or want to configure Zeppelin to be accessed via HTTPS.

3 PARAGRAPH EXECUTION

We have made some changes in the paragraph execution for the Denodo Interpreter. In this interpreter we create a connection with Denodo in each paragraph and we allow to specify the database to connect. By default we will use the database defined in the `zeppelin.jdbc.denodo.database` property of the interpreter configuration but if we want to connect with a different database we can write the database name concatenated with the interpreter name. The format is `%denodo%DATABASE_NAME`.



The screenshot shows two paragraphs in a Zeppelin Notebook. Both paragraphs execute the same SQL query: `select * from bv_invoices_test where amount > ${amount = 5000}`. The left paragraph uses the interpreter `%denodo%billing` and displays a table of invoice data. The right paragraph uses the interpreter `%denodo%billing|` and displays a grouped bar chart of the same data.

invoice_id	date_invoice	order_id	date
497	Sun Mar 01 01:55:40 CET 2015	45682	Tue Fr CET 2
498	Sun Mar 01 02:32:55 CET 2015	34117	Tue Fr CET 2
504	Sun Mar 01 06:24:52 CET 2015	31517	Mon F CET 2
505	Sun Mar 01 08:13:59 CET 2015	18067	Tue Fr CET 2
559	Tue Mar 03 21:08:06 CET 2015	1327	Fri Fe CET 2

The bar chart on the right shows the distribution of invoice amounts by country. The y-axis represents the amount, ranging from 0 to 87,400. The x-axis lists countries: United States, Croatia, Canada, and Bulgaria. The United States has the highest number of invoices, followed by Croatia, Canada, and Bulgaria.

4 APPENDIX I: CONFIGURING BROWSERS FOR KERBEROS

4.1.1.1 Firefox

By default Firefox does not enable SPNEGO authentication, and consequently Kerberos. It has to be enabled manually therefore.

4.1.1.1.1 Enable Kerberos Authentication

1. Go to **about:config** in the address bar in Firefox
2. Click "*I'll be careful, I promise*" when warned about changing advanced settings
3. Enter *negotiate* in the Search box
4. Set value of the **network.negotiate-auth.trusted-uris** to your domain name

4.1.1.1.2 Enable Kerberos Delegation

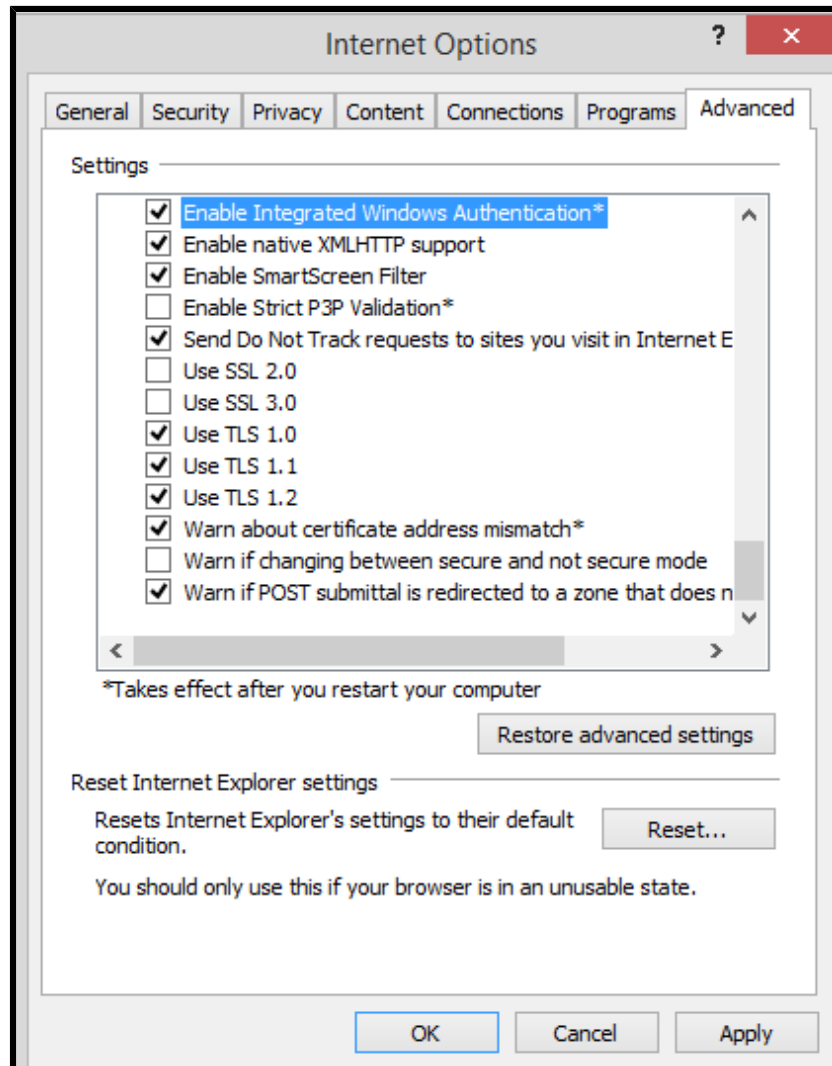
1. Go to **about:config** in the address bar in Firefox
2. Click "*I'll be careful, I promise*" when warned about changing advanced settings
3. Enter *negotiate* in the Search box
4. Set value of the **network.negotiate-auth.delegation-uris** to your domain name

Restart browser and check that everything works.

4.1.1.2 Internet Explorer

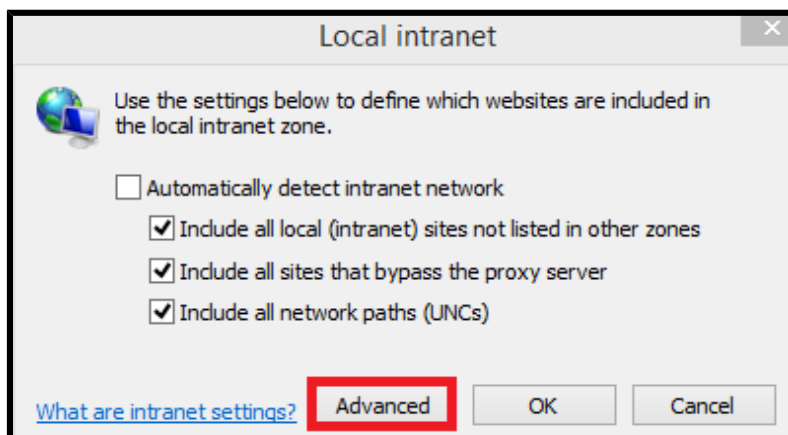
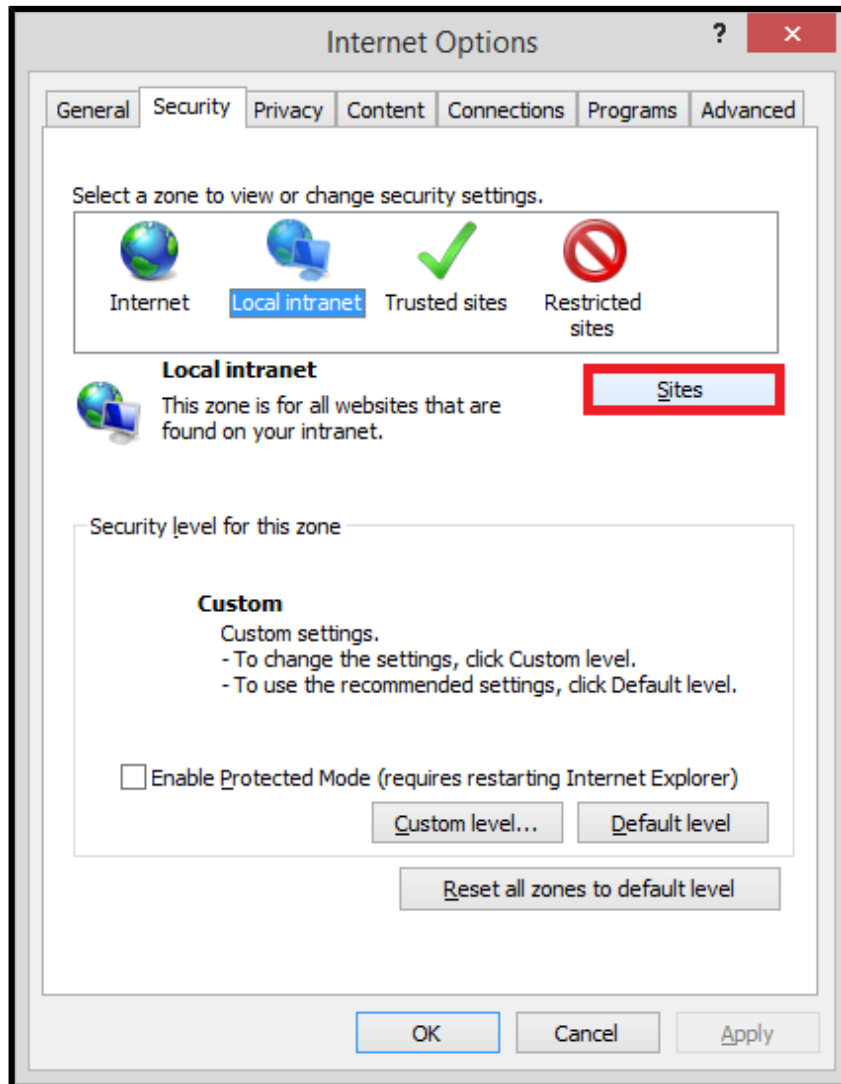
4.1.1.2.1 Enable Kerberos Authentication

1. Click **Tools -> Internet Options**
2. Advanced tab
3. Enable checkbox for Enable Integrated Windows Authentication

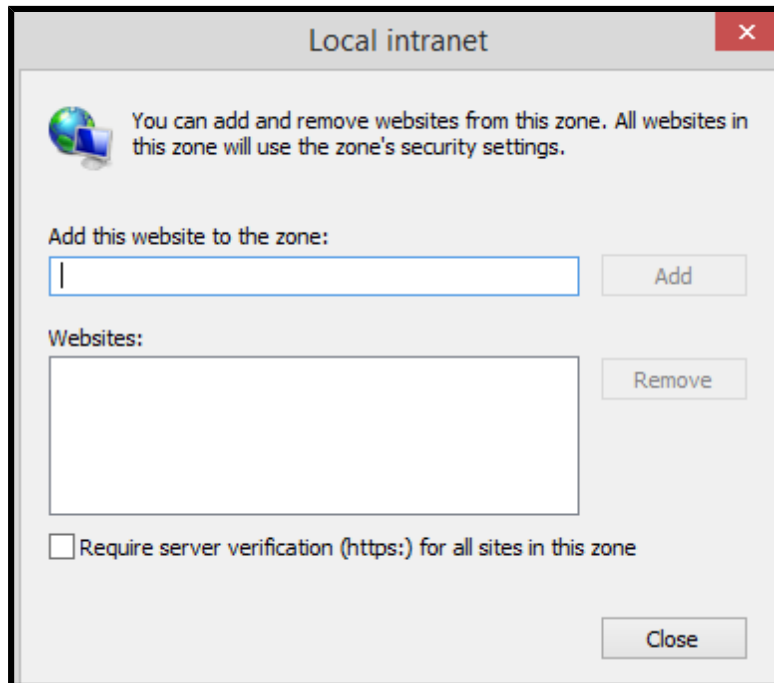


4.1.1.2.2 Enable Kerberos Delegation

1. Click **Tools -> Internet Options**
2. **Security tab -> Local intranet -> Sites**



3. Add the site in question.



Restart browser and see if everything works.

4.1.1.3 Chrome

Chrome in Windows will use the Internet Explorer settings, so configure them within Internet Explorer's **Tools** -> **Internet Options** dialog as explained in the previous section.

5 APPENDIX II: UPDATING THE DENODO JDBC DRIVER

The Denodo-customized Zeppelin distribution is already provided with the Denodo VDP JDBC Driver (`denodo-vdp-jdbcdriver.jar`) included in the `interpreter/denodo` folder. You may need to update this driver to match any update you have installed on the Denodo VDP server

To update the Denodo VDP JDBC Driver you must take the driver included in the `lib/extensions/jdbc-drivers/vdp-VERSION` directory of your Denodo platform and copy into the Zeppelin `interpreter/denodo` folder.

6 APPENDIX III: INSTALLING ADDITIONAL INTERPRETERS

The Apache Zeppelin for Denodo Standalone distribution is provided with some pre installed interpreters (angular, denodo, file, jdbc, md, python, sh, spark). The original distribution provides a Linux script to install interpreters and we add a new script to install interpreters in Windows.

The new script is `install-interpreter.cmd` and the execution is the same as [Linux](#).

6.1.1.1.1 Install all community managed interpreters

```
.\bin\install-interpreter.cmd --all
```

6.1.1.1.2 Install specific interpreters

```
.\bin\install-interpreter.cmd --name md,shell,jdbc,python
```

6.1.1.1.3 Get list of community managed interpreters

```
.\bin\install-interpreter.cmd --list
```

Please note that some interpreters have specific restrictions on which versions of JDK they can be used with, not respecting these restrictions may cause problems during the installation and performance of these interpreters.

7 APPENDIX IV: USING THE PYTHON INTERPRETER IN WINDOWS

The Python interpreter is already included in the Apache Zeppelin for Denodo - Standalone distribution. This interpreter is compatible up to Python version 3.7.

To use it in Windows, it will be necessary to have the directory "C:\tmp" and that the user who is running zeppelin has permissions on this directory.

8 APPENDIX V: CREATION OF MULTIPLE DENODO INTERPRETERS

You may wish to query more than one Denodo server. For this it will be necessary to create a new Denodo interpreter in the interpreter section of the application.

- The name of the denodo interpreters must always begin with "denodo".
- In the interpreter group option, "jdbc" must be selected.
- The "default.driver" property must be set to "com.denodo.vdp.jdbc.Driver".
- The "default.url" property must be set to "jdbc:vdb://localhost:9999/?noAuth=true".
- You must add the property "zeppelin.jdbc.denodo.database" and configure the default database.

After applying this configuration it will be possible to create notes with the new Denodo interpreter.

There will be some requirements that must be satisfied depending on the type of authentication.

- In case of using Basic authentication, the user with which you login in zeppelin must exist in all VDP servers to which you want to access to execute queries. The password must be the same in all Denodo servers.
- In case of using Kerberos authentication, it is important that all VDP servers are configured using the same SPN and the same KeyTab.
- In case of using the Denodo SSO authentication, it is important that all VDP servers are using the same Denodo SSO application. The `sso.url` property of the `conf\SSOConfiguration.properties` must point to the same url.

After the creation of new denodo interpreters, the authentication of the application will continue to be made in the VDP server configured for the default denodo interpreter.

All other interpreters will need to use the same authentication data when executing paragraphs. That is why it is important that the authentication configuration in the VDP servers is the same.

To use the new interpreters the procedure is the same as before. When creating a new note, the new Denodo interpreter is selected in Default Interpreter. In the paragraph execution by default we will use the database defined in the `zeppelin.jdbc.denodo.database` property of the interpreter configuration but if we want to connect with a different database we can write the database name concatenated with the interpreter name. The format is `%denodoXXX%DATABASE_NAME`.

In the particular case of using a new "denodo2" interpreter, when logging into the application, this will continue to be done with the VDP server configured in this "denodo" interpreter. When we enter a note configured with the interpreter "denodo2" and execute a paragraph, a new connection will be created using the data of the initial authentication.