You won't be able to pull or push project code via SSH until you add an SSH key (/profile/keys/new) to your profile

Add key (/profile/keys/new) | Don't show again (/profile?user%5Bhide_no_ssh_key%5D=true) | Later

Home (/bdre/documentation/wikis/home) | Pages (/bdre/documentation/wikis/pages) | Git Access (/bdre/documentation/wikis/git_access)

how to checkout projects

Page History (/bdre/documentation/wikis/how-to-checkout-projects/history)

If you can see this page then you are added to the BDRE project. This document will help you build BDRE from source. Audience for this document are developers and architects who want be part of BDRE framework development or may just want to test it by running the UI. Please read this document (https://gitlab.com/bdre/documentation/wikis/list-of-things-to-learn-for-BDRE-framework-development) and understand the skillset needed (or to be learned) for contributing to BDRE.

Setup

- Install Git, MVN and Oracle JDK 7 if you haven't already. In Windows be sure add git and other bash tools in the commandline path during installation. To get started download and install following free, safe and open source software.
  - Oracle JDK 7 Download for Windows (https://s3.amazonaws.com/bdre-public/software/jdk-7u79-windows-x64.exe)
  - Git Command line Client (For Windows - GitBash)Download for Windows (https://git-scm.com/download/win)
  - Maven Download for Windows (http://apache.mirrors.pair.com/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.zip)
  - IntelliJ Idea (download Community edition)Download for Windows (https://www.jetbrains.com/idea/download/)
  - MySQL 5.6 Download for Windows (https://s3.amazonaws.com/bdre-public/software/mysql-installer-community-5.6.25.0.msi)

Alternatively you can look into our shared folder \\D-113074428\bdre\Software if you are within Wipro backbone and download the software from there.

For complete software requirement read this (https://gitlab.com/bdre/documentation/wikis/software-requirement-for-bdre-development). You don't need to download all software to run the UI in the browser. Git, Maven, MySQL and Intellij should be enough.

Important:

After installing Git first set your full name (like John Doe) and email id in git config using following command.

```
git config --global user.name "Your Name"  git config --global user.email "your_email@wipro.com"
```

Replace *Your Name* and *your_email* with your real name and Wipro email.

- Add `jdk` location in the PATH env variable.
- Add `JAVA_HOME` env variable with your installed jdk location.
- Add `mvn` location in the PATH env variable.
- Add `M2_HOME` env variable with your installed maven location.
- To use git from Command line, add `git` in the PATH env variable.
- Add `mysql` location in the PATH env variable.

Proxy setup while working inside Wipro network

- If you are working from Wipro office and behind Wipro proxy then you need to setup proxy for command line operations. The easiest way to do that would be adding 3 environment variables before performing any git operations. If you want to set proxy permanently then read this (http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sysdm_advancd_environmnt_addchange_variable.mspx?mfr=true).

```
In Linux use this:
export http_proxy=http://<wipro_ad_id>:<ad_pwd>@proxy1.wipro.com:8080
export https_proxy=http://<wipro_ad_id>:<ad_pwd>@proxy1.wipro.com:8080
export no_proxy=localhost

Or you can use this in Windows:
set http_proxy=http://<wipro_ad_id>:<ad_pwd>@proxy1.wipro.com:8080
set https_proxy=http://<wipro_ad_id>:<ad_pwd>@proxy1.wipro.com:8080
set no_proxy=localhost
```

Test your git access:

Git Proxy Setting: When you are working behind a proxy, git needs proxy configuration to connect to repositories.

For e.g. to set proxy:

```
git config --global https.proxy https://WIPRO\username:password@proxy1.wipro.com:8080
git config --global http.proxy http://WIPRO\username:password@proxy1.wipro.com:8080
```

To reset:

```
git config --global --unset https.proxy  git config --global --unset http.proxy
```

You may skip this section if you are familiar with Git version control or in hurry :-).

Please clone the following **test repo** in Gitlab and create your branch , make and push your change and test anything you want. You need to create a ssh key to make changes. Contact @bdremishi, @bdrearijit, @bdreharsha, @bdrekapil if you face any issues.

Test this: `git clone http://wosggitlab.wipro.com/bdre/gitlabtest.git`

If you are not familiar with Git version control, please read the following Git tutorials.

BDRE quick git head start - bdre_getting_started.pptx (https://gitlab.com/bdre/documentation/uploads/af02e6fef5ef1137429561877703fcb4/bdre_getting_started.pptx)

Git Tutorials (https://www.atlassian.com/git/tutorials/) (note we are not using Atlassian git but git command line tutorial is same for all git providers)

Watch Git Video Tutorial (https://www.youtube.com/watch?v=7p0hrpNaJ14)

Once you are able to access git and became familiar, please checkout three BDRE repositories.

MD Repository (metadata_management)

`git clone http://wosggitlab.wipro.com/bdre/metadata_management.git`

```
You can proceed to 'Building' section if you just want to run the UI. You do not need to clone or build the IM, Application or Jack repo to run the UI.
```

IM Repository (im_framework)

`git clone http://wosggitlab.wipro.com/bdre/im_framework.git`

Jack Repository:

`git clone http://wosggitlab.wipro.com/bdre/jack.git`

```
**Note:**

If you are part of BDRE Contributors and have access to Gitlab, then use the following git urls in place of the previous ones:

* Gitlab Test: `git clone https://YOURID@gitlab.com/bdre/test.git`.
Where *YOURID* is your gitlab user id (like bdrejohn).

* MD Repository (metadata_management): `git clone https://<yourid>@gitlab.com/bdre/metadata_management.git`.

* IM Repository (im_framework): `git clone https://<yourid>@gitlab.com/bdre/im_framework.git`.

* Jack Repository: `git clone https://<yourid>@gitlab.com/bdre/jack.git`.
```

Important: Dealing with line endings

Linux and Windows differ in how they handle line endings.

The `git config core.autocrlf` command is used to change how Git handles line endings. It takes a single argument. So it is better to set the `autocrlf` in git accordingly to avoid further complications.

- In Windows use: `git config --global core.autocrlf true` .
- In Linux use: `git config --global core.autocrlf input` .

For further changes to convert CRLF to LF when you are using linux systems refer to this link (https://help.github.com/articles/dealing-with-line-endings/ (https://help.github.com/articles/dealing-with-line-endings/))

Building

- You need to create your own environment. To create your environments, edit
    - metadata_management/md-commons/src/main/resources/mybatis-config.xml

You can setup an environment called `local` and assuming you are running MySQL server with default port configuration `localhost:3306` . Later you'd set the credentials of the DB. If `local` already exists then edit it or create it by coping an existing entry.

mybatis-config.xml

```
<environment id="local">
            <transactionManager type="JDBC"/>
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.jdbc.Driver"/>
                <property name="url" value="jdbc:mysql://localhost:3306/platmd"/>
                <property name="username" value="root"/>
                <property name="password" value="root"/>
            </dataSource>
</environment>
```

ENVIRONMENT file

- Create a file called ENVIRONMENT in md_commons and im_commons resources directory. Sample ENVIRONMENT (https://gitlab.com/bdre/documentation/uploads/e9500dec3cb508d4839d6c9a7edcfea5/ENVIRONMENT) file.
- This file is in git ignore so does not get checked in. It must point to your environment code.

```
environment=local
```

- Open your settings.xml. It’s in **~/.m2/settings.xml**. If not present please create it. Alternatively you can right click any .pom in Idea and open/create **settings.xml**.
- Obtain the settings.xml (https://gitlab.com/bdre/jack/raw/develop/installer/guestfiles/maven/settings.xml) from Jack/installer/guestfiles/maven/ and replace your local **~/.m2/settings.xml**.
- NOTE : Add or change the jar versions in settings.xml according to your need but pasting settings.xml (https://gitlab.com/bdre/jack/raw/develop/installer/guestfiles/maven/settings.xml) AS IS should be fine. *If you have proxied internet access then configure proxy (http://maven.apache.org/guides/mini/guide-proxies.html)

In Wipro office, placing following in settings.xml should work.

```
<proxies>
        <proxy>
            <id>wipro-proxy</id>
            <active>true</active>
            <protocol>http</protocol>
            <username>yourUserName</username>
            <password>yourPassword</password>
            <host>proxy1.wipro.com</host>
            <port>8080</port>
            <nonProxyHosts>localhost</nonProxyHosts>
        </proxy>
</proxies>
```

- Perform a mvn clean install in MD(first) and IM projects only.

Running the BDRE Web UI

- Start MySQL 5.6 server and setup *platmd* DB and MySQL credentials

```
mysql -uroot -e "create database platmd;"
```

```
mysqladmin -u root password 'root'
```

- create the tables and procs using metadata_management/mysql/scripts
  - cd into metadata_management folder and execute following scripts

```
sh <YourPath>\metadata_management\mysql\scripts\create-tables.sh <database_user> <database_password> <database_name> <databse_server_hostname> <databa
```

```
sh <yourpath>\metadata_management\mysql\scripts\create-procs.sh <database_user> <database_password> <database_name> <databse_server_hostname> <databas
```

- After a successful build of both the MD and IM projects, cd into md-rest-api folder and start the Jetty server using metadata_management/md-rest-api/quick-run.bat (or .sh if you are using Unix)(Make sure the Batch Scripts Support for Windows or Bash Support for Unix plugin is installed)
- Use Chrome browser and open http://localhost:9999/mdui/pages/content.page (http://localhost:9999/mdui/pages/content.page)
- Login using admin/zaq1xsw2
- BDRE home page looks like this (https://gitlab.com/bdre/documentation/uploads/6d1699cdbdb15648ba0b849dd1923367/bdress.png)

Creating a test job

- Create a RDBMS data import job from *Job Setup From Template* > *New RDBMS Import Job*
- Change the JDBC URL/credentials to your localhost MySQL platmd DB that contains BDRE metadata.
- Click test connection
- Select 2 - 3 tables with columns like this (https://gitlab.com/bdre/documentation/uploads/d93c5de43dbf79502226d7227a7b46ae/rdbms_import.png)
- Create the jobs and see the pipeline (https://gitlab.com/bdre/documentation/uploads/3bce103069908d065beb8d04b15f7cc7/pipeline.png).
- Click *XML* , *Details* etc and checkout the generated workflow.

Open the projects in IntelliJ Idea

- We strongly recommend using IntelliJ Idea to develop BDRE. However BDRE should work fine with other Java IDEs
- Start Idea and go to **File > Open ...** and then browse to the three repo folders from three separate Idea windows.
- Idea will automatically detect the maven projects and prompt to *Import Changes*. Import changes. Here (https://gitlab.com/bdre/documentation/uploads/5578910e1812ce684dd70b73fe345f0c/BDRE_idea.png) is how the metadata_management project looks while opened in Idea.
- After completion of import, open the terminal window and perform usual git and mvn operations.
  - e.g. mvn clean install
- You *must* now create your own branch in all three projects when you are ready to contribute to BDRE. **Note:** Never modify code in develop branch.
- Please use git properly (https://gitlab.com/bdre/documentation/uploads/af02e6fef5ef1137429561877703fcb4/bdre_getting_started.pptx) and contribute to the development.

- For completed, committed and pushed changes, open *Merge Request* from gitlab.com.
- Good luck!

How to Deploy Process in your hadoop environment

After making entries into metadata related to your process, you have to deploy this process in your hadoop environment.

Steps to deploy your process

1. Click deploy button on process page corresponding to the process you want to deploy. (Deploy button will show status regarding deployment of process, when you hover over the button.)
2. You have to copy deploy-env.properties (https://gitlab.com/bdre/jack/tree/develop/installer/guestfiles/deploy/deploy-env.properties) file and process-deploy.sh (https://gitlab.com/bdre/applications/blob/develop/app-commons/scripts/process-deploy.sh) file from application repositories and put properties file in same directory from where you are running the .sh file. You have to provide executable permissions to every shell script.
3. You have to edit deploy-env.properties. Set all properties based on your environment.
4. Process Deploy main class calls deploy script based on type of container process. For Example for process type 1 it will call process-type-1.sh script.
5. You have to specify path of your process-type-n.sh script in administration -> settings -> mdconfig -> deploy.script-path as `process-type-` . You have to provide executable permissions to every shell script.
6. You have to setup a crontab in your environment. Crontab will run process-deploy.sh script. for example `*/5 * * * * /home/cloudera/process-deploy.sh` .
7. After the deployment is complete and in UI the status for the process is deployed (turns green), you have to execute it.
8. Download Workflow.py (https://gitlab.com/bdre/applications/blob/develop/app-commons/scripts/Workflow.py) and flume.sh (https://gitlab.com/bdre/applications/blob/develop/app-commons/scripts/flume.sh). Update Workflow.py and flume.sh according to your environment.
9. Update hostname variable in Workflow.py.
10. Update `pathForFlumeng` , `pathForFlumeconf` , `pathForFlumeconfFile` variables in flume.sh.
11. To store logs for execution, you have to create log directory as mentioned in administration -> settings -> mdconfig -> execute.log-path. create directory and give permission for every user to write into it. `chmod -R 777 <folderName>`
12. Also update path in administration -> settings -> mdconfig -> execute.oozie-script-path tags for Workflow.py. And under administration -> settings -> mdconfig -> execute.standalone-script-path for flume.sh.
13. You have to start a jetty server on your hadoop environment. You will need rest-exec module in metadata_management repository. cd into rest-exec and run executor.sh to start the server. Then you have to specify the url of this server to administration -> settings -> mdconfig -> execute.rest-exec-url.
14. By default server will start at 7777 port. You can change it, based on which port is open for you by updating executor.sh.

Last edited by AR288503 (/u/AR288503) 12 days ago