# Repository Test Plan:

This test plan focuses on validating the capabilities and features of a Pentaho repository.

All jobs in this repository use one or more of slave servers during execution.

CMI end to end focuses on the command line tools used to work with a repository. Import with and without rules, execution from a slave server with authorized and unauthorized users. This test also validates several jobs/transformation features that are prone to defects.

PDI and PUC focuses on use of the UI tool set to work with a repository. Import, execution from PUC and PDI both manually and through the schedule, use of explorer, and user permissions.

Permissions focuses on the permissions across PDI and PUC

Large Repository focuses on importing a large repository (this is more for performance)

# Repository Execution

1. Set up a 2 VMs with Remote Repository Test Environment Configuration Set Up steps
2. Under the Ubuntu configuration execute Repository-CMI-Integration Test and the large repository test.
3. Under Windows configuration execute PUC and Spoon Integration Test and Permissions Test. **Note that connection to the repository will use the ip and port of the Pentaho server VM not localhost.**
4. If a step with a referenced test case fails, but it doesn’t impact the objective of being able to import, connect to, execute jobs as admin or as suzy and the 2 negative tests case pass for import with rules and execution of a job in an admin folder by suzy.
   1. Fail the referenced test case, and write up a defect, reference the defect on the test case
   2. Continue testing the end to end test
   3. Pass the end to end and all passing referenced tests if all objectives are met.
5. If any test step fails and it impacts the objective of the end to end.
   1. Fail the referenced test case and reference the defect on the test case
   2. Block all other reference test cases
   3. Fail the end to end test case and reference the same defect
6. Under Window configuration execute Repository-PUC and Spoon-Integration
7. Follow the same rules as 3 & 4 for execution results
8. Under Window configuration execute Repository-PUC and Spoon-Permissions
9. Follow the same rules as 3 & 4 for execution results

NOTE: If the test needs to be updated for any reason, you will need to follow the update process for this repository.

# Repository Tests or Test Plan

1. Add or modify an existing test cases to repository test plan template.
2. Add and/or make any changes to .ktr/.kjb and save to the repository
3. Add the jobs/transformation within a job need to be execute on one or more of the slave server (CarteServer1 or 2)
4. Add any baseline and data files /home/devuser/9.0.0.0/(data, baselines)
5. Run the job and make sure it works
6. Exit out of the repository
7. In Spoon go to Tools🡪Repository 🡪Export Repository
8. Export the repository to /home/devuser/9.0.0.0/repositories or C:\9.0.0.0\repositories with the same name. Select yes to overwrite it.
9. Copy the repository back to the 172.16.10.71 /home/devuser/9.0.0.0/repositories

NOTE: make sure that there is not a later version of the repository uploaded

1. Copy any data or baseline files to /home/devuser/9.0.0.0/(data, baselines)
2. In the test case description add which configuration set up the test needs, and the repository it lives in and put the following assets can be found at:

<http://iwiki.pentaho.com/display/QA/Engineering+Test+Databases#EngineeringTestDatabases-TestAssets>