




Configuring the Pipeline Engine

 In XNAT version 1.9 and onwards, the Pipeline Engine Framework requires two components:


- Pipeline Engine
- XNAT Pipeline Engine Plugin

The XNAT Pipeline Engine is a Java-based framework that links sequential activities, human and computer, into a defined process flow and manages how data moves from step to step in that flow based on the results of each step. In most laboratories, some processes (or pipelines) are carried out automatically without any human intervention while others require a person to do manual steps, such as drawing a region of interest. Pipeline Engine facilitates both fully automated and semi-automated workflows. Pipelines can be executed up to a step, then notify a user(s) to perform the manual task, and finally restart the pipeline from the next step. The process flow is defined in an XML document called the *pipeline descriptor* and the executables are defined in a separate XML document called *resource descriptors*.

Pipeline engine is a standalone tool. However, when used with XNAT one can:

- set up project-based workflows with project specific and experiment specific parameters,
- track a pipeline and send email notifications
- capture provenance information as the pipeline executes

Download and Installation

-  The location of <XNAT_HOME> depends on your system configuration. In the default XNAT Vagrant machine, this is located in **/data/xnat/home**



When you install XNAT, pipeline engine is setup for you. By default, the pipeline engine is located in a directory next to XNAT_HOME: **/data/xnat/pipeline**, which we will call PIPELINE_HOME.

Pipeline engine can be located anywhere on the file system, as long as the PIPELINE_HOME folder has appropriate permissions for the user who launches tomcat and the non-default path to the PIPELINE_HOME should be set using Administer -> More Options -> Default Settings.

If you choose to setup pipeline engine outside XNAT, after downloading the engine, launch

SSH Terminal

TEXT

Copy

```
1 <PIPELINE_HOME>/setup.sh YOUR_ADMIN_EMAIL_ID YOUR_SMTP_SERVER
```

This step will generate the file <PIPELINE_HOME>/pipeline.config

<PIPELINE_HOME>/pipeline.config

CODE

Copy

```
1 PIPELINE_EMAIL_ID = admin@your-place.org
2 PIPELINE_SMTP_HOST = your.smtp.server
3 PIPELINE_CATALOG_ROOT_PATH = PATH_TO_CATALOG_FOLDER
4 ADMIN_EMAIL = admin@your-place.org
5 XNAT_SITE = YOUR_SITE_ID
```

Creating a pipeline

Please refer to the **XNAT Pipeline Development Schema** developer documentation.

Installing a pipeline

Only site admins can install pipelines in XNAT. Please refer to the admin documentation on **Installing Pipelines in XNAT**.

Once a pipeline has been installed, any project owner can add it to their project.

Running a pipeline

Pipelines can be executed in two modes viz. standalone mode using `<PIPELINE_HOME>/bin/PipelineRunner` or by updating XNAT as the pipeline progresses. In order to update XNAT, use `PIPELINE_HOME/bin/XnatPipelineLauncher` with appropriate parameters.

✔ User Docs: Running Pipelines In XNAT

Tutorial Video



Mohana Ramaratnam - Pipelines In Depth

NRG

2:19:44

Connecting To a Compute Cluster / GRID

Pipeline engine ships with support for Sun Grid Engine (SGE). This is done using the DRMAA API. A site can use its own job scheduling application and integrate it with **XNAT/Pipeline Engine** by "overloading" the script `PIPELINE_HOME/bin/schedule`. Out of the box, this script just invokes the command that is passed. `PIPELINE_HOME/lib/PipelineClient.jar` contains the class `org.nrg.pipeline.client.PipelineJobSubmitter` which supports scheduling jobs on the SGE.

`/Pipeline/resourceRequirements` can be used to set resource requirements like Architecture, free memory while submitting the job onto the grid.

Copyright © 2025 • Powered by Scroll Viewport & Atlassian Confluence