XNAT Documentation

XNAT Pipeline Development Schema

⚠ XNAT version 1.9 onwards, the Pipeline Engine Framework now requires two components:

- the Pipeline Engine
- the XNAT Pipeline Engine Plugin.

✅ **PIPELINE_HOME** is a shortcode referred to in this documentation. It stands for the location of your pipeline directory in the XNAT web app file system. This can be set in **Admin UI - File System Settings**. By default, this location is `/data/home/xnat/pipeline` . XNAT comes preinstalled with a series of example pipelines that can be found in this folder. Some of these example pipelines are mentioned in this documentation.

Creating a pipeline involves:

- Installing the package/executable(s) that will be executed by the pipeline
- Creating the pipeline descriptor
- Creating resource descriptors
- Optional - creating velocity template file, creating screen and action class
- Modifying/creating report page(s) for the results of a pipeline

We recommend that pipelines and resource descriptors be placed in a separate folder within PIPELINE_HOME/catalog

ℹ XPATH and Pipeline

As pipeline engine uses XML documents, one can use the power of XPATH expressions to navigate through the various elements and attributes. One can thus set parameters using XPATH statements. A string contained within the caret symbol (^) is treated as XPATH expression and the engine resolves all such expressions before executing the steps.

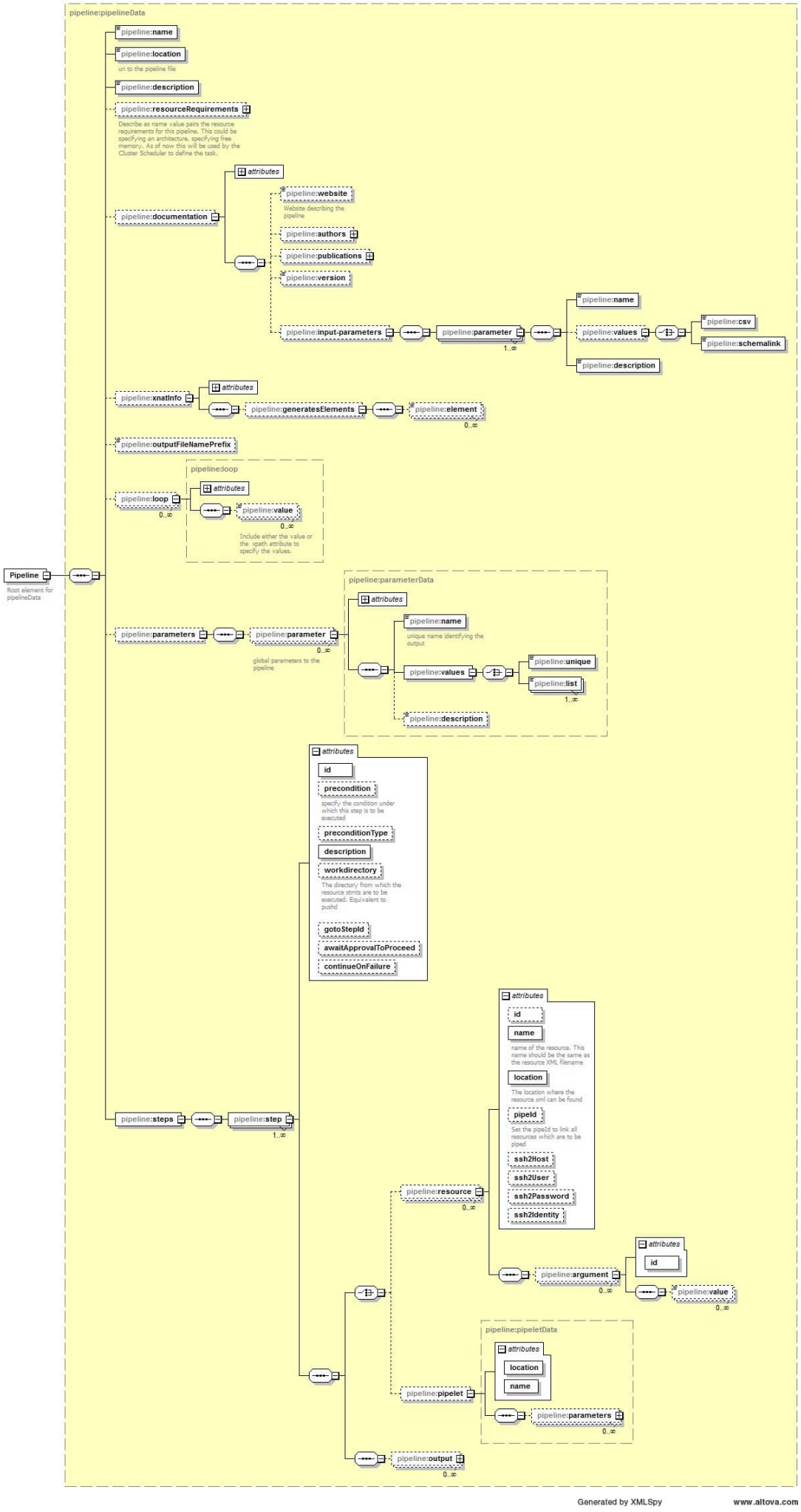
Pipeline Schema

Pipelines are defined using a **pipeline descriptor** and **resource descriptors**. A resource descriptor describes an executable. An executable is identified by its name, its location and the arguments that it takes.

Pipeline Descriptor

PIPELINE_HOME/sample_pipelines/SampleAutoRunPipeline.xml is an instance of a pipeline descriptor.

Schema Representation of Pipeline Element


pipeline:pipelineData
pipeline:name
pipeline:location
uri to the pipeline file
pipeline:description
pipeline:resourceRequirements
Describe as name value pairs the resource requirements for this pipeline. This could be specifying an architecture, specifying free memory. As of now this will be used by the Cluster Scheduler to define the task.
pipeline:documentation
attributes
pipeline:website
Website describing the pipeline
pipeline:authors
pipeline:publications
pipeline:version
pipeline:input-parameters
pipeline:parameter
1..∞
pipeline:name
pipeline:values
pipeline:csv
pipeline:schemalink
pipeline:description
pipeline:xnatInfo
attributes
pipeline:generatesElements
pipeline:element
0..∞
pipeline:outputFileNamePrefix
pipeline:loop
attributes
pipeline:value
0..∞
Include either the value or the xpath attribute to specify the values.
pipeline:loop
0..∞
Pipeline
Root element for pipelineData
pipeline:parameterData
attributes
pipeline:name
unique name identifying the output
pipeline:values
pipeline:unique
pipeline:list
1..∞
pipeline:description
pipeline:parameters
pipeline:parameter
0..∞
global parameters to the pipeline
attributes
id
precondition
specify the condition under which this step is to be executed
preconditionType
description
workdirectory
The directory from which the resource stmts are to be executed. Equivalent to pushd
gotoStepId
awaitApprovalToProceed
continueOnFailure
attributes
id
name
name of the resource. This name should be the same as the resource XML filename
location
The location where the resource xml can be found
pipeId
Set the pipeId to link all resources which are to be piped
ssh2Host
ssh2User
ssh2Password
ssh2Identity
pipeline:resource
0..∞
pipeline:argument
0..∞
attributes
id
pipeline:value
0..∞
pipeline:pipeletData
attributes
location
name
pipeline:pipelet
pipeline:parameters
0..∞
pipeline:output
0..∞
pipeline:steps
pipeline:step
1..∞
Generated by XMLSpy
www.altova.com

Element Name	Purpose
name	Name of the pipeline
description	A short description of the pipeline which is displayed to the user.

resourceRequirements	Name, Value pair of requirements for running the pipeline. This is used while sche grid. Eg:
	CODE
	`<resourceRequirements>`
	`  <property name="DRMAA_JobTemplate_JobResource">-l arch=lx24-amd64,mem_f`
	`</resourceRequirements>`

documentation	Use this element to inform the XNAT users' about the pipeline. Set the parameters needs using the input-parameters. The parameter can be specified as a XPATH us comma separated list using *csv*.

xnatInfo	Set the datatype that the pipeline is applicable to using the *appliesTo* attribute and that the pipeline will create using *generatesElements*. E.g.
	CODE
	`<xnatInfo appliesTo="xnat:mrSessionData">`
	`  <generatesElements>`
	`  <element>fs:aparcRegionAnalysis</element>`
	`  <element>fs:asegRegionAnalysis</element>`
	`  </generatesElements>`
	`</xnatInfo>`

outputFileNamePrefix	Pipeline engine captures the STDOUT and STDERROR when the pipeline executes specify the file path prefix. STDOUT will be created as .log and error as .err.
loop	A pipeline step can be executed for a list of values. Create such a list using loop e CODE `<loop id="mpragescans" xpath="^/Pipeline/parameters/parameter[name='mprs']` `mpragescans` identifies the list of values specified as the xpath statement. Note statement is contained within ^ symbol. Using `PIPELINE_LOOPON(mpragescans)` will mimic a for loop over the values of Using `PIPELINE_LOOPVALUE(mpragescans)` will result in all values of mpragesc For example: CODE

Element Name	Purpose
	<step id="0" description="Prepare Folder Structure" workdirectory="^/Pipeline/parameters/parameter[name='workdir']/values/unic <resource name="mkdir" location="commandlineTools"> <argument id="p"/> <argument id="dirname"> <value>^concat('RAW/',PIPELINE_LOOPON(mpragescans))^</value> </argument> </resource> </step>
parameters	Use this element to specify the parameters to the pipeline *inline*.
steps	Use this element to specify the ordered sequence of steps that the pipeline engin step results in call to possibly multiple executables.
step	Each step is identified by its ID attribute. **Attributes**: `precondition [CONDITION]` : The step is executed only if the precondition evalu `workdirectory [PATH]` : The directory within which the executables will be invo `gotoStepId [STEP ID]` : Like a GOTO statement, can be used to bypass the ord steps. `awaitApprovalToProceed [true|false]` : Setting this attribute to true will resu terminating execution. This is like a pause. The step at which the pipeline can be r using the ID attribute. `continueOnFailure [true|false]` : The pipeline engine stops executing when status of non-zero value. If you want to override this behavior, set `continueOnFa`
step/resource	This sequence of resources specifies the ordered collection of tasks to be done. A email, executing a script etc. Each task is defined in a resource descriptor. **Attributes:** `name` : the name of the resource descriptor `location` : path to the resource descriptor. A resource descriptor is identified usi NOTE: the location attribute does not refer to the location of the executable, it refe the XML which describes the executable. The location when not absolute is relati PIPELINE_CATALOG_ROOT_PATH property in pipeline.config file. `ssh2*` : One can execute a task remotely using the ssh2 credentials set using ssh ssh2User,ssh2Password and ssh2Identity. The data is not copied on the remote h here is that the folder in which the data is present is mounted on both hosts - rem which the pipeline engine is executing. `pipeId` : A sequence of executables can be chained within a step using the `pipe`
step/pipelet	A pipelet is a pipeline. One can string together pipelines to create a new pipeline. passed to the pipelet.

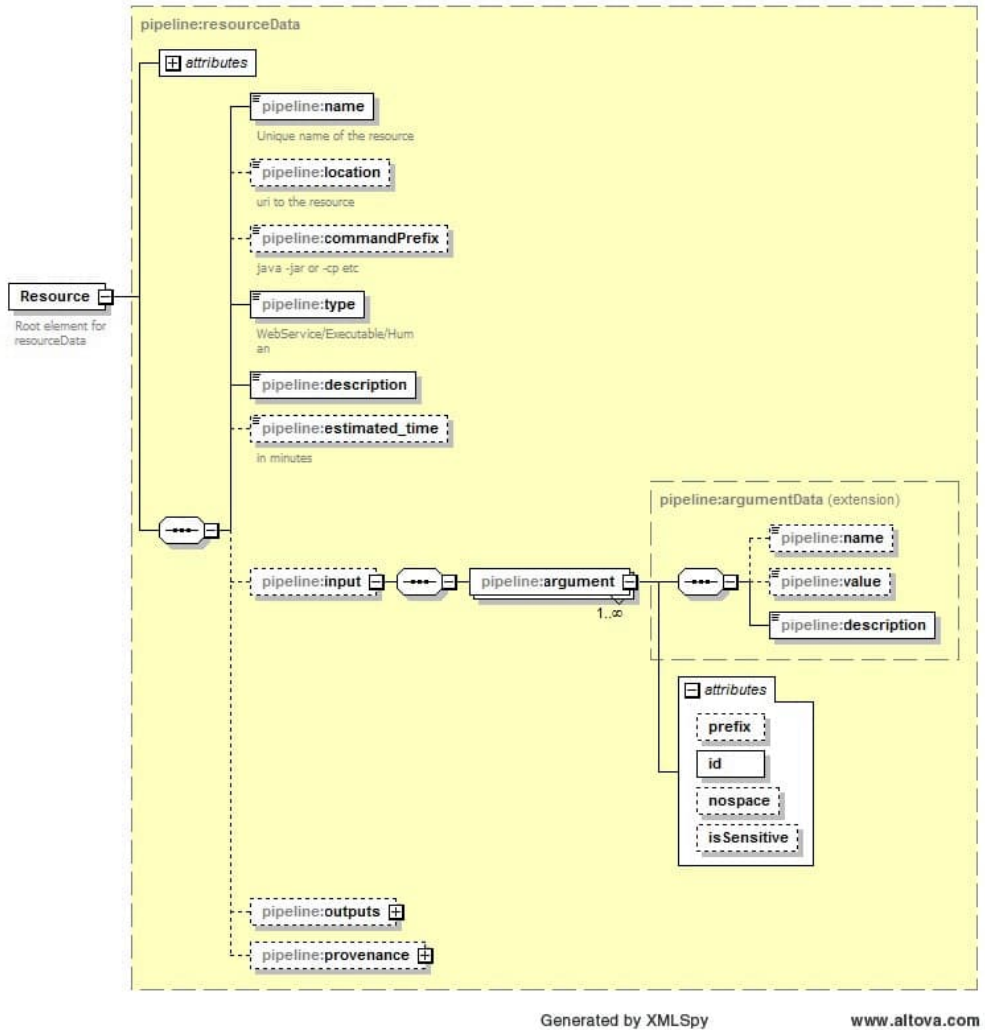
Element Name	Purpose
step/output	This element defines a collection of files which a step may create.

Resource Descriptor

An executable is invoked with appropriate arguments. A resource descriptor defines the executable - its location, its arguments, its output

PIPELINE_HOME/catalog/ant-tools/AntCopy.xml is an instance of a resource descriptor.

Schema Representation of Resource Element



Element Name	Purpose
name	Name of the executable
location	path to the executable
commandPrefix	prefix to be used before invoking executable at `location/name`.

Element Name	Purpose
input/argument	`name` - the argument name as used by the executable `value` - value of the argument **Attributes:** `id` - This is the arguments ID `prefix` - The prefix to be used. E.g. "-" or "--" or "/". The default value is "-" `nospace [true|false]` - This attribute specifies if a space character should be present between an argument and its value `isSensitive [true|false]` - This attribute when set to true, is masked in all log files

Specifying Parameters for a Pipeline (Parameter Descriptor)

Input parameters can be specified inline within the pipeline descriptor document or on the command prompt or as a parameter file. Specifying the parameters inline on a production pipeline is rare as parameters change with the project/experiment.

PIPELINE_HOME/sample_pipelines/Parameters.xml is an instance of a parameter descriptor document.