# CAS-PGE: Learn by Example

The following guide serves as a hands-on learning exercise for explaining the basics of how a CAS-PGE project can be set up and used. For this exercise, it will be necessary to download a sample project. Please obtain this sample project from the following link [#download].

## Example Overview

The example project detailed in this exercise is called FileConcatenatorPGE. This CAS-PGE project performs two functions. First, it collects two input files and concatenates them together into a second file. Second, it generates metadata from the generated product and ingests this metadata into a cas-filemanager instance.

## Requirements

* A deployed CAS-Workflow instance. See [#workflow] for instructions on how to set this component up
* A deployed CAS-Filemanager instance. See [#filemgr] for instructions on how to set this component up
* A deployed CAS-Crawler. See [#crawler] for instructions on how to set this component up
* Maven 2
* Environment variables
  + CRAWLER\_HOME
  + WORKFLOW\_HOME
  + WORKFLOW\_URL
  + FILEMGR\_URL
  + PGE\_ROOT = /usr/local/pge
    - The directory in which the PGE scripts and configuration files will reside.
* Ensure $WORKFLOW\_HOME/lib contains at least the following:
  + cas-crawler-<VERSION>.jar
  + cas-filemgr-<VERSION>.jar
  + cas-pge-<VERSION>.jar

### 1. Setting up CAS-PGE relevant directories.

There are a number of components associated with typical CAS-PGE deployments. These can include CAS-PGE configuration files, external scripts, input files, output files etc. The below steps will help guide you in setting up a configuration directory for the FileConcatenator PGE project as well as setting up a deployment directory for running your PGE. Note, the deployment directory could be located anywhere and it is assumed that for a production project, this directory could be shared among multiple-PGE services.

[1. Create CAS-PGE configuration directory]

> cd /usr/local

> mkdir –p $PGE\_ROOT/file\_concatenator/pge-configs

[2. Create CAS-PGE deployment directory]

> mkdir –p $PGE\_ROOT/file\_concatenator/output/jobs

[3. Create CAS-PGE input files directory]

> mkdir –p $PGE\_ROOT/file\_concatenator/files

[4. Create CAS-PGE extractors directory]

> mkdir –p $PGE\_ROOT/file\_concatenator/extractors

## 2. Download the FileConcatenatorPGE project

The FileConcatenatorPGE project is a Java project that uses the Maven build system for producing a run-time CAS-PGE library. Please follow the below instructions to download and extract the project.

[1. Download FileConcatenatorPGE]

[#downlod link]

[2. Extract project]

tar xf fileconcatenator-pge.tar –C /usr/local/src

## 3. Customize and deploy the CAS-PGE configuration file

The CAS-PGE configuration file for identifying the steps involved in executing the PGE are located in fileconcatenator-pge/src/main/resources/config/PGEConfig.xml.

The PGEConfig.xml file performs the following functions:

1. Describes how to run the PGE (ie. what external programs to call and in which order)
2. Defines custom metadata used within the execution of the CAS-PGE
3. Describes how to build metadata files generated as a result of the execution of the CAS-PGE and what to do with these files

Below is the sample PGEConfig.xml file used within the fileconcatenator-pge project:

<?xml version="1.0" encoding="UTF-8"?>

<pgeConfig>

<!-- How to run the PGE -->

<exe dir="[JobDir]" shell="/bin/bash">

<!-- cd to PGE root -->

<cmd>cd [PGE\_ROOT]/file\_concatenator</cmd>

<cmd>cp [InputFile1] [OutputFile]</cmd>

<cmd>cat [InputFile2] >> [OutputFile]</cmd>

</exe>

<!-- Files to ingest -->

<output>

<!-- one or more of these -->

<dir path="[JobDir]" createBeforeExe="false">

<!-- one or more of these \*\* regExp or name can be used-->

<files regExp=".\*\.txt" metFileWriterClass="org.apache.oodt.pge.examples.fileconcatenator.writers.ConcactenatingFilenameExtractorWriter" args="[PGE\_ROOT]/file\_concatenator/extractors/concatenatingfilename.extractor.config.xml"/>

<files regExp=".\*\.txt" metFileWriterClass="org.apache.oodt.cas.pge.writers.metlist.MetadataListPcsMetFileWriter" args="[PGE\_ROOT]/file\_concatenator/extractors/metlistwriter/metout.xml"/>

</dir>

</output>

<!-- Custom metadata to add to output files -->

<customMetadata>

<!-- helpful keys -->

<metadata key="LessThan" val="&#x3C;"/>

<metadata key="LessThanOrEqualTo" val="[LessThan]="/>

<metadata key="GreaterThan" val="&#x3E;"/>

<metadata key="GreaterThanOrEqualTo" val="[GreaterThan]="/>

<metadata key="Exclamation" val="&#33;"/>

<metadata key="Ampersand" val="&#38;"/>

<metadata key="NotEqualTo" val="[Ampersand]="/>

<metadata key="LogicalAnd" val="[Ampersand][Ampersand]"/>

<metadata key="CshPipeToStdOutAndError" val="[GreaterThan][Ampersand][Exclamation]"/>

<metadata key="ProductionDateTime" val="[DATE.UTC]"/>

<metadata key="JobDir" val="[PGE\_ROOT]/file\_concatenator/output/jobs/job-[ProductionDateTime]"/>

<metadata key="InputFile1" val="[PGE\_ROOT]/file\_concatenator/files/concatenatingInputFile1.txt"/>

<metadata key="InputFile2" val="[PGE\_ROOT]/file\_concatenator/files/concatenatingInputFile2.txt"/>

<metadata key="OutputFile" val="[JobDir]/concatenatedOutputFile-[ProductionDateTime].txt"/>

</customMetadata>

</pgeConfig>

## 4. Build and deploy FileConcatenatorPGE

Deploy the fileconcatenator-pge JAR pacakge

> cd /usr/local/src/fileconcatenator-pge

> mvn package

> mv target/fileconcatenator-pge-\*.jar $WORKFLOW\_HOME/lib

Deploy fileconcatenator-pge resources

1. PGEConfig.xml
   1. cp /usr/local/src/fileconcatenator-pge/src/main/resources/config/PGEConfig.xml $PGE\_ROOT/file\_concatenator/pge-configs
2. Sample files
   1. cp /usr/local/src/fileconcatenator-pge/src/main/resources/files/concatenatingInputFile\*.txt $PGE\_ROOT/file\_concatenator/files
3. Extractor configuration file
   1. cp /usr/local/src/fileconcatenator-pge/src/main/resources/extractors/concatenatingfilename.extractor.config.xml $PGE\_ROOT/file\_concatenator/extractors
4. Met-list writer configuration file
   1. cp /usr/local/src/fileconcatenator-pge/src/main/resources/extractors/metlistwriter/metout.xml $PGE\_ROOT/file\_concatenator/extractors/metlistwriter

## 5. Configure deployed CAS-Workflow for running FileConcatenatorPGE

1. Navigate to your deployed CAS-Workflow’s policy directory

> cd $WORKFLOW\_HOME/policy

1. Modify events.xml

Add the following entry to this file:

<event name="fileconcatenator-pge">

<workflow id="urn:oodt:FileConcatenatorWorkflow"/>

</event>

1. Create a new policy file titled: fileconcatenator-pge.workflow.xml.

Add the following entries to this file:

<cas:workflow xmlns:cas="http://oodt.jpl.nasa.gov/1.0/cas"

name="FileConcatenatorWorkflow"

id="urn:oodt:FileConcatenatorWorkflow">

<tasks>

<task id="urn:oodt:FileConcatenator"/>

</tasks>

</cas:workflow>

1. Modify tasks.xml

Add the following entries to this file:

<task id="urn:oodt:FileConcatenator" name="FileConcatenator"

class="org.apache.oodt.pge.examples.fileconcatenator.FileConcatenatorPGETask">

<conditions/>

<configuration>

<property name="PGETask\_Name" value="FileConcatenator"/>

<property name="PGETask\_ConfigFilePath" value="[PGE\_ROOT]/file\_concatenator/pge-configs/PGEConfig.xml" envReplace="true"/>

<property name="PGETask\_DumpMetadata" value="true"/>

<property name="PCS\_WorkflowManagerUrl" value="[WORKFLOW\_URL]" envReplace="true" />

<property name="PCS\_FileManagerUrl" value="[FILEMGR\_URL]" envReplace="true"/>

<property name="PCS\_MetFileExtension" value="met"/>

<property name="PCS\_ClientTransferServiceFactory" value="org.apache.oodt.cas.filemgr.datatransfer.LocalDataTransferFactory"/>

<property name="PCS\_ActionRepoFile" value="file:[CRAWLER\_HOME]/policy/crawler-config.xml" envReplace="true"/>

</configuration>

<requiredMetFields>

<metfield name="RunID"/>

</requiredMetFields>

</task>

1. Modify workflow-lifecycles.xml

Add the following entries to this file (if not already present):

<stage name="pge\_setup\_build\_config\_file">

<status>BUILDING CONFIG FILE</status>

</stage>

<stage name="pge\_staging\_input">

<status>STAGING INPUT</status>

</stage>

<stage name="pge\_exec">

<status>PGE EXEC</status>

</stage>

<stage name="pcs\_crawl">

<status>CRAWLING</status>

</stage>

1. Modify workflow-instance-met.xml

Add the following entry to this file:

<workflow id="urn:oodt:FileConcatenatorWorkflow">

<field name="RunID"/>

</workflow>

1. Restart CAS-Workflow

> cd $WORKFLOW\_HOME/bin

> ./wmgr restart

## 6. Run File Concatenator PGE

1. Navigate to CAS-Workflow home binary directory

> cd $WORKFLOW\_HOME/bin

1. Invoke the File Concatenator PGE by running the wmgr-client command-line

> ./wmgr-client --url http://localhost:9001 --operation --sendEvent --eventName fileconcatenator-pge --metaData --key RunID testNumber1

## 7. Verify output of PGE execution

After invoking the wmgr-client script as directed above, you should see an entry like the following:

*INFO: Successfully ingested product: [/usr/local/pge/file\_concatenator/output/jobs/job-2011-08-05T23:42:51.178Z/concatenatedOutputFile-2011-08-05T23:42:51.178Z.txt]: product id: a2d6d5ff-bfbc-11e0-8531-dff90856f73a*

Additionally, you should see a the below two files in the generated job directory:

* Generated product file: $PGE\_ROOT/file\_concatenator/output/jobs/job-2011-08-05T23\:42\:51.178Z/concatenatedOutputFile-2011-08-05.txt
* Generated met file: $PGE\_ROOT/file\_concatenator/output/jobs/job-2011-08-05T23\:42\:51.178Z/concatenatedOutputFile-2011-08-05.txt.met