Thales Communications & Security

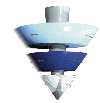
20–22 rue Grange Dame Rose CS 80518

78141 Vélizy Cedex

France

Tel.: +33 (0)1 73 32 00 00

[www.thalesgroup.com](http://www.thalesgroup.com)



ANIMATOR to SHAPE

How to import existing SCADAsoft schematics in web based HMI

FOLLOW OF TECHNICAL DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **LOG OF CHANGES** | | | |
| **Revision** | **Date** | **Author** | **Modification** |
| **AA** | 27/01/2015 | JC MENCHI | Creation |
| **AB** | 02/06/2015 | JC MENCHI | Update for SVG background (MWT3.1.0) |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **APPROVAL** | | | | |
|  | **Name** | **Role** | **Date** | **Signature** |
| **Written by** | JC Menchi | SWARC | 27/01/2015 |  |
| **Verified by** |  |  |  |  |
| **Approved by** | O BERNARD | SWEM |  |  |
| **Approved by the customer if necessary** |  |  |  |  |

CONTENTS

[1. Overview 4](#_Toc410750431)

[1.1 Basic Usage 4](#_Toc410750432)

[1.2 Installation 5](#_Toc410750433)

[1.3 How to use 5](#_Toc410750434)

[1.4 limitationS 6](#_Toc410750435)

[1.5 Dependency 6](#_Toc410750436)

[1.6 Usefull Tools 7](#_Toc410750437)

[2. Example 8](#_Toc410750438)

[3. File format 12](#_Toc410750439)

[3.1 Animation class mapping file 12](#_Toc410750440)

[3.2 INSTANCE mapping file 12](#_Toc410750441)

[3.3 Symbol import file 12](#_Toc410750442)

[3.4 Geoserver styling description 13](#_Toc410750443)

[ANNEXE A GEOSERVER Configuration 21](#_Toc410750444)

Figure

[Figure 1 : animator to MWT 4](#_Toc410750445)

[Figure 2 : Animator image example 8](#_Toc410750446)

Overview

The new web based HMI for SCADAsoft 6 is not anymore based on Animator animation toolkit like version 4 or 5. The new version is based on THALES Hypervisor MWT web animation package.

The MetaConfigurator support only the new MWT format and cannot read Animator project.

In order to import images from previous SCADAsoft version a conversion tool has been developed. This tool is provided as-is and work in a best effort mode to import as much as possible. It may be necessary to use other tools to edit converted data to build a correct image.

This documentation is not a reference manual for svg, geoserver or shape file format. It assumes basic knowledge of geoserver, especially on how to make an image from shape files.

Animator images can be converted to shape files to be used with geoserver or to SVG files to be used directly by MWT. SVG is the preferred format for version based on Hypervisor VS 9.4 or higher. Shape format is the only option for Hypervisor VS 9.3.

## Basic Usage

The conversion tool is python script that can read Animator project and convert images to a format that can be used by MetaConfigurator. From those data, MetaConfigurator can produce configuration files for Hypervisor MWT web based HMI.

For every animator images (.ani) the tool will produce several files:

* Static graphic information in svg or shape format,
* active backdrop information in svg or shape format,
* SVG file with active symbol position and RTDB connection info for MetaConfigurator image editor.



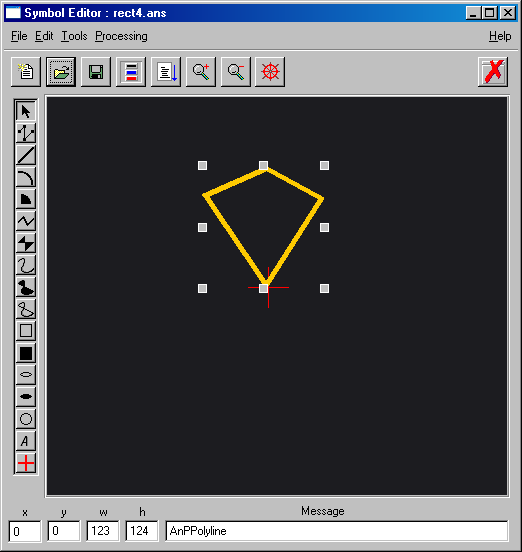
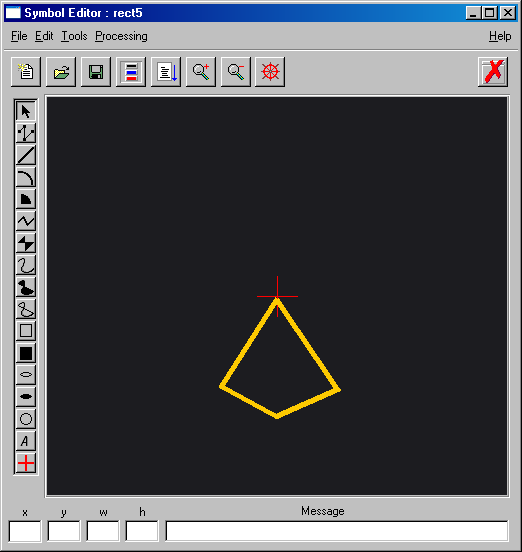
1. animator to MWT for GIS



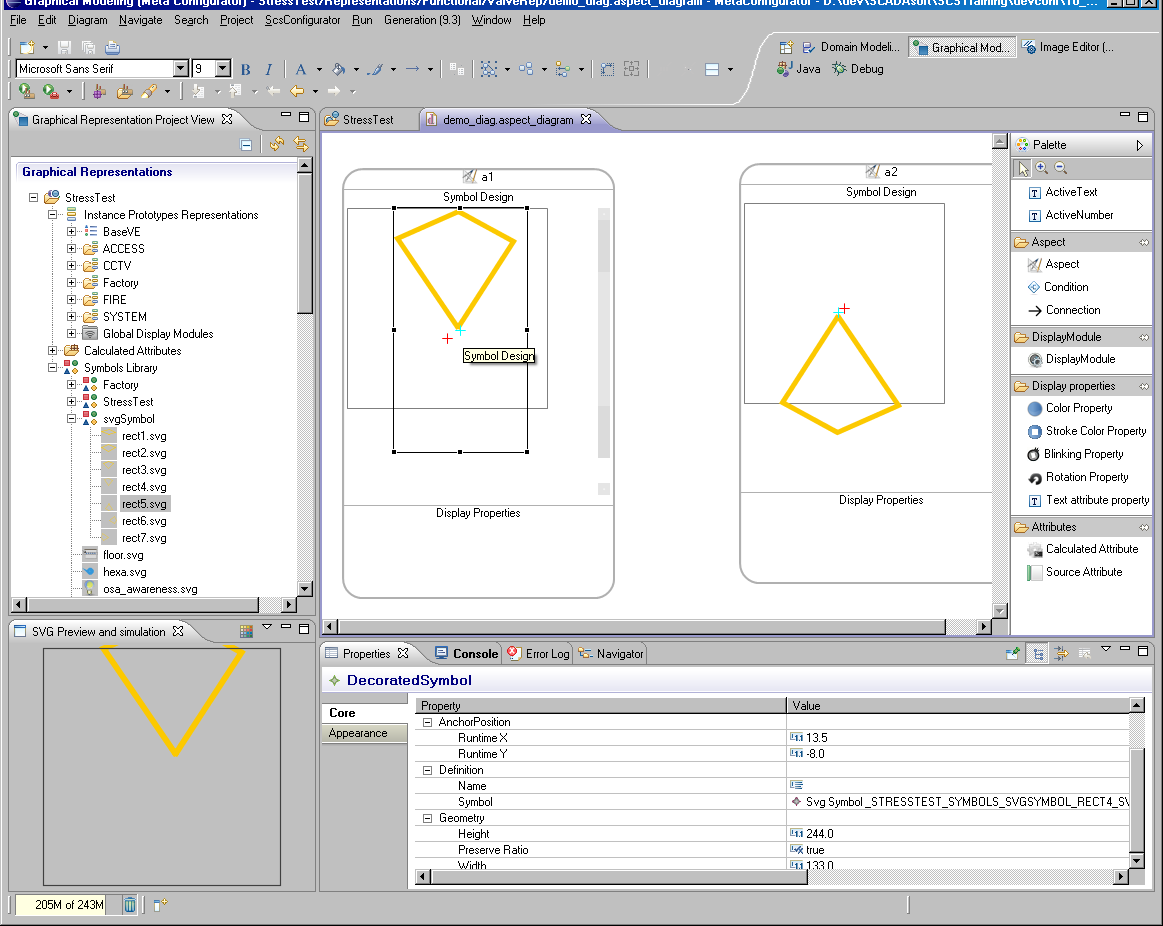
1. Animator to MWT for SVG

### Symbol management

For every animator symbol (.ans), the tool will produce a svg file that is centered on the animator symbol origin (red cross). This origin is not necessarily at the center of the geometry it is used to align the different symbol for animation. In the MetaConfigurator aspect editor this origin is shown as a blue cross (the red cross is the center of the aspect itself).

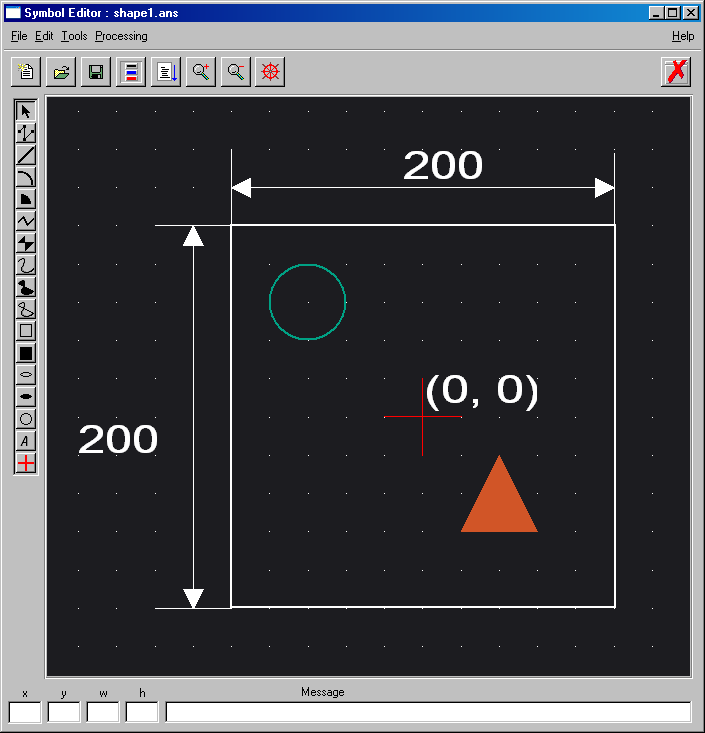
 

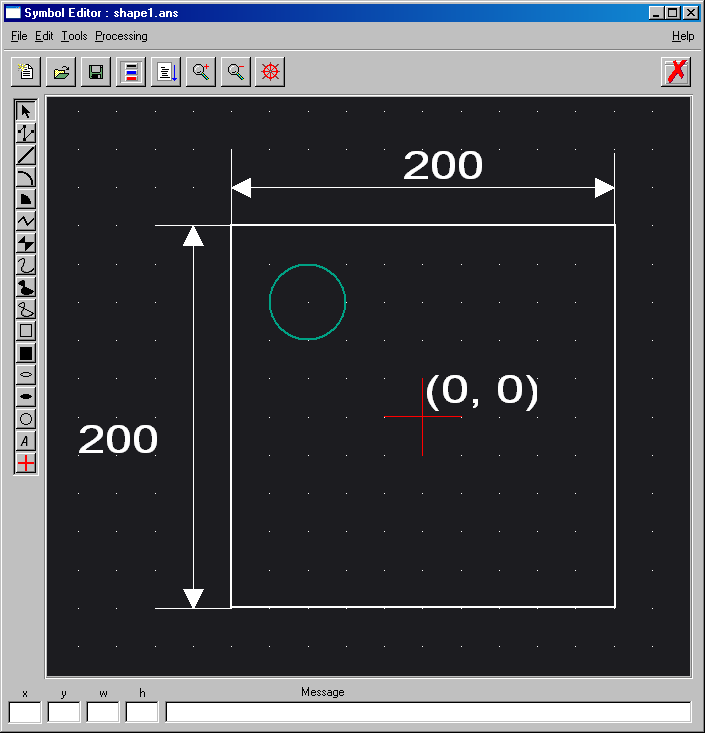
1. animator symbol editor



1. Same symbol in Aspect editor

In metaconfigurator the aspect editor is designed to use a drawing area of 200x200 pixels.

For a new project that is using scstudio and metaconf, it is interesting to take this drawing area into account when drawing a new symbol. 



## Installation

The converter is written in Python, using version 2.7 and 2 extra libraries:

* Python shape file library <https://github.com/GeospatialPython/pyshp> version 1.2.1
* Python parsing <https://pypi.python.org/pypi/pyparsing/2.0.3> version 2.0.3.

Install any Python 2.7 distribution then set you PATH and PYTHONHOME according to your installation.

The converter installation contains the 2 libraries; it is made of 3 files: ani2img.py, pyparsing.py, shapefile.py.

The converter will use the SCADAsoft tool an2svg.exe to convert animator data to svg, so a SCADAsoft 5.3.1P2 installation is necessary.

To run the converter executes the following command: python ani2img.py.

Use the correct options depending on your configuration.

usage: ani2img.py [-h] [-d ROOTDIR] [-i IMGDIR] [-c CLASSMAPPINGFILE] [-m INSTANCEMAPPINGFILE] [-e EXPORTLIST] [-W WIDTH] [-H HEIGHT] [-f] [-l LOGLEVEL] [-F FONT]

optional arguments:

-h, --help show this help message and exit

-d ROOTDIR, --datadir ROOTDIR

root folder of Animator workspace (default .)

-i IMGDIR, --imagedir IMGDIR

Animator subimage folder to convert (default '')

-c CLASSMAPPINGFILE, --classMappingFile CLASSMAPPINGFILE

Mapping file between animator class and Hypervisor generic symbol (default no mapping)

-m INSTANCEMAPPINGFILE, --instanceMappingFile INSTANCEMAPPINGFILE

Mapping file between animator dbpath and Hypervisor equipment ID (default no mapping)

-e EXPORTLIST, --export EXPORTLIST

Type of file to export: svg, shape, symbol, instance (default svg)

Application log level [error,warn,info,debug] (default 'info')

-F FONT, --font FONT Font to use for text (default use animator specified font)

-W WIDTH, --width WIDTH

Width of exported image (default source width)

-H HEIGHT, --height HEIGHT

Height of exported image (default source height)

-f, --full Export ActiveClass geometry too

-l LOGLEVEL, --logLevel LOGLEVEL

## How to use

Animator images contain not only graphic information, but SCADAsoft RTDB connection information and animation class. Similar information exists in MetaConfigurator but with different naming. So it is necessary to provide mapping information to the converter, so that it can generate files that can be interpreted by the MetaConfigurator.

Two mapping are necessary:

* SCADAsoft RTDB address to instance ID of Hypervisor equipment
* Animator ActiveClass name to MetaConfigurator Graphic Representation

The mapping are simple CSV files that are specified using –c and –m option, it is possible to use several –c or –m option on the command line, so you can divide your mappings in several files.

For instance mapping it is possible to use the same scs2hv.xml file used by the SCADAsoft to Hypervisor connector. The convertor understands CSV, or xml file for instance mapping.

To ignore something you can map the Animator name to an empty string.

If no mapping exists the converter will reuse the animator name.

At the end of conversion, the converter will show a list of all the unknown names. So it is possible to run the tool once without mapping to have a list of name to map.

You may have to use the –F option to specify the font to be used by geoserver, to have a correct rendering of non ASCII characters.

A typical usage of the convertor is:

python ani2img.py –e shape –e instance –F SimSun -c gzmmapClass.csv -m gzm1.csv –m scs2hv.xml -d D:\SCADAsoft\gzml5-gws\_G2PPAT130810\ScsVisu\Animator -i L5\OPS > import.log

The –e option is used to choose what to convert:

**-e symbol** convert all animator symbol (.ans) to svg

**-e svg** convert static graphic and active backdrop to svg

**-e shape** convert static graphic and active backdrop to shape files for geoserver

**-e instance** create pseudo svg file for MetaConfigurator instance import.

You may use as many –e options as needed.

The -i option allow you to convert only some of your images (the one in the image folder L5/OPS).

Check the log file for error or missing mapping.

The converter creates a folder named ***images***, containing for each animator image a set of shape and svg files. The generated files use the animator image name as a prefix.

If –e symbol is used, the converter creates a folder named ***svgSymbol*** with one svg file per animator symbol. This folder can be imported directly in MetaConfigurator Graphic Modeling as symbols to be used in aspect.

For example if the image is named Site1.ani, the generated files will be:

* Static background for geoserver : Site1pll.shp, Site1pll.shx, Site1pll.dbf, Site1plg.shp, Site1plg.shx, Site1plg.dbf
* Shape layer for geoserver : Site1pllobj.shp, Site1pllobj.shx, Site1pllobj.dbf, Site1plgobj.shp, Site1plgobj.shx, Site1plgobj.dbf
* Static background svg : Site1.svg
* SVG for shape animation (active backdrop): Site1\_shapes.svg
* Instances file (ActiveSymbol) : Site1\_inst.svg

Shape files ending with pll contain polyline graphics and the one ending with plg contain polygons. A shape file contains only one type of entities; this is why it is necessary to generate different files for polyline and polygon. Depending on the geometry existing in the image, pll or plg files will be generated or not.

Shape files ending with obj contain geometry for ActiveBackdrop.

All the shape files have to be imported in geoserver.

The svg file contains information necessary to add equipment instances in the MetaConfigurator Image Editor.

## limitationS

The converter can only export image and basic symbol information, it cannot convert active symbol to MetaConfigurator GraphicRepresentation.

## Dependency

The converter generates shape files as defined by ESRI specification:

<http://en.wikipedia.org/wiki/Shapefile>

[http://www.**esri**.com/library/whitepapers/pdfs/**shapefile**.pdf](http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf)

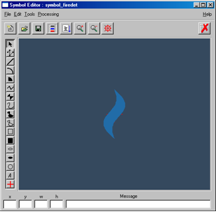
## Usefull Tools

Shape editor: OpenJUMP, QGIS,

DBF editor: libreoffice

SVG editor: Inkscape

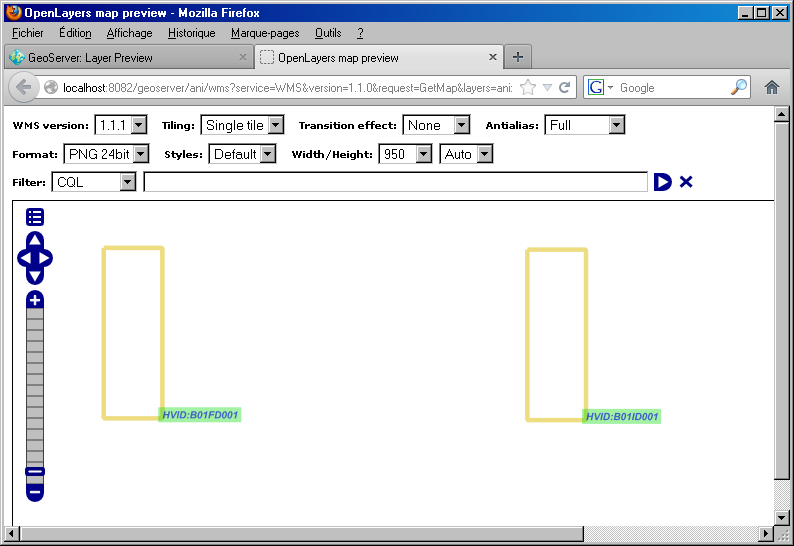
# Example



1. Animator image example

In this







File format

## Animation class mapping file

AnClassType, MC\_GraphicRep

ActiveSymbolInstance:valve,

ActiveBackdropInstance:smokeAB, FireZoneActiveBackdrop

ActiveBackdropInstance:intrusionAB, IntrusionActiveBackdrop

MobileSymbolInstance:camera, CameraRepVS9

In this example, all active symbol named valve are ignored.

## INSTANCE mapping file

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

<scsbamapping scsenv='SRV1' hvcreatorid='SCADA1'>

<classdesc hv='com.thalesgroup.myproject.data.config.equipment.BasicEquipmentStatusesType' scs='IPDeviceType'>

<instance hv='B01RT01A' scs=':SITE1:B001:F000:SYSTEM:RT01A'/>

<instance hv='B01RT02B' scs=':SITE1:B001:F000:SYSTEM:RT01B'/>

</classdesc>

</scsbamapping>

AnDBPath, HV\_ID

:SITE1:B001:F002:ACCESS:DO011, B01DO011

:SITE1:B001:F002:ACCESS:DO012, B01DO012

:SITE1:B002:F001:ACCESS:DO002,B02DO002

## Symbol import file

See MetaConfigurator documentation**: Meta Configurator IRS 62 939 840 – 506 – Rev F**

## Geoserver styling description

Using the following styling for shape made from animator is really important for performance and good text rendering.

This file has to be imported in geoserver with the name **ani\_line.sld** .

<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" version="1.0.0">

<sld:NamedLayer>

<sld:Name>Default Styler</sld:Name>

<sld:UserStyle>

<sld:Name>Default Styler</sld:Name>

<sld:Title>Animator Line</sld:Title>

<sld:Abstract>line coming from animator2shape transformation</sld:Abstract>

<sld:FeatureTypeStyle>

<sld:Name>name</sld:Name>

<sld:Rule>

<ogc:Filter>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>FILLED</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:Filter>

<sld:PolygonSymbolizer>

<sld:Fill>

<sld:CssParameter name="fill">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

</sld:Fill>

</sld:PolygonSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>solid</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>dot</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-dasharray">1 2</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>doubledot</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-dasharray">1 1 1 3</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>dashdot</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-dasharray">5 2 1 2</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>dash</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-dasharray">5 2</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

<sld:Rule>

<ogc:Filter>

<ogc:And>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>LINE</ogc:Literal>

</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_DASH</ogc:PropertyName>

<ogc:Literal>longdash</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:And>

</ogc:Filter>

<sld:LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Stroke>

<sld:CssParameter name="stroke">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-width">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="stroke-dasharray">10 5</sld:CssParameter>

</sld:Stroke>

</sld:LineSymbolizer>

</sld:Rule>

</sld:FeatureTypeStyle>

<sld:FeatureTypeStyle>

<sld:Name>text0</sld:Name>

<sld:Rule>

<ogc:Filter>

<ogc:PropertyIsEqualTo>

<ogc:PropertyName>SCS\_STYLE</ogc:PropertyName>

<ogc:Literal>TEXT</ogc:Literal>

</ogc:PropertyIsEqualTo>

</ogc:Filter>

<sld:TextSymbolizer uom="http://www.opengeospatial.org/se/units/metre">

<sld:Label>

<ogc:PropertyName>SCS\_TEXT</ogc:PropertyName>

</sld:Label>

<sld:Font>

<sld:CssParameter name="font-family">

<ogc:PropertyName>SCS\_FONT</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="font-size">

<ogc:PropertyName>SCS\_WIDTH</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="font-style">

<ogc:PropertyName>SCS\_FTSTYL</ogc:PropertyName>

</sld:CssParameter>

<sld:CssParameter name="font-weight">

<ogc:PropertyName>SCS\_FTWGT</ogc:PropertyName>

</sld:CssParameter>

</sld:Font>

<sld:LabelPlacement>

<sld:PointPlacement>

<sld:AnchorPoint>

<sld:AnchorPointX>0</sld:AnchorPointX>

<sld:AnchorPointY>0</sld:AnchorPointY>

</sld:AnchorPoint>

</sld:PointPlacement>

</sld:LabelPlacement>

<sld:Halo>

<sld:Radius>0</sld:Radius>

<sld:Fill>

<sld:CssParameter name="fill">#DDDDDD</sld:CssParameter>

</sld:Fill>

</sld:Halo>

<sld:Fill>

<sld:CssParameter name="fill">

<ogc:PropertyName>SCS\_COLOR</ogc:PropertyName>

</sld:CssParameter>

</sld:Fill>

<sld:VendorOption name="conflictResolution">false</sld:VendorOption>

<sld:VendorOption name="goodnessOfFit">0.1</sld:VendorOption>

<sld:VendorOption name="spaceAround">-1</sld:VendorOption>

</sld:TextSymbolizer>

</sld:Rule>

</sld:FeatureTypeStyle>

</sld:UserStyle>

</sld:NamedLayer>

</sld:StyledLayerDescriptor>

1. GEOSERVER Configuration

In order to have a good rendering of data coming from animator, it is important to do some change in the geoserver default setup.

**First step: add a gridset for the geoserver tiling**

Open the file <GEOSERVER\_DATA\_DIR>/data\_dir/gwc/geowebcache.xml and add inside the “<gridSets>” element the following block:

(creating manually a gridset is really time consuming)

<gridSet>

<name>EPSG:23030</name>

<srs>

<number>23030</number>

</srs>

<extent>

<coords>

<double>-10.0</double>

<double>-10.0</double>

<double>10010.0</double>

<double>10010.0</double>

</coords>

</extent>

<alignTopLeft>false</alignTopLeft>

<resolutions>

<double>40.0</double>

<double>30.0</double>

<double>20.0</double>

<double>10.0</double>

<double>8.0</double>

<double>6.0</double>

<double>4.0</double>

<double>3.5</double>

<double>3.25</double>

<double>3.0</double>

<double>2.75</double>

<double>2.5</double>

<double>2.25</double>

<double>2.0</double>

<double>1.5</double>

<double>1.0</double>

<double>0.8</double>

<double>0.6</double>

<double>0.4</double>

<double>0.2</double>

<double>0.1</double>

<double>0.05</double>

<double>0.025</double>

<double>0.0125</double>

</resolutions>

<metersPerUnit>1.0</metersPerUnit>

<pixelSize>2.8E-4</pixelSize>

<scaleNames>

<string>EPSG:23030:0</string>

<string>EPSG:23030:1</string>

<string>EPSG:23030:2</string>

<string>EPSG:23030:3</string>

<string>EPSG:23030:4</string>

<string>EPSG:23030:5</string>

<string>EPSG:23030:6</string>

<string>EPSG:23030:7</string>

<string>EPSG:23030:8</string>

<string>EPSG:23030:9</string>

<string>EPSG:23030:10</string>

<string>EPSG:23030:11</string>

<string>EPSG:23030:12</string>

<string>EPSG:23030:13</string>

<string>EPSG:23030:14</string>

<string>EPSG:23030:15</string>

<string>EPSG:23030:16</string>

<string>EPSG:23030:17</string>

<string>EPSG:23030:18</string>

<string>EPSG:23030:19</string>

<string>EPSG:23030:20</string>

<string>EPSG:23030:21</string>

<string>EPSG:23030:22</string>

<string>EPSG:23030:23</string>

</scaleNames>

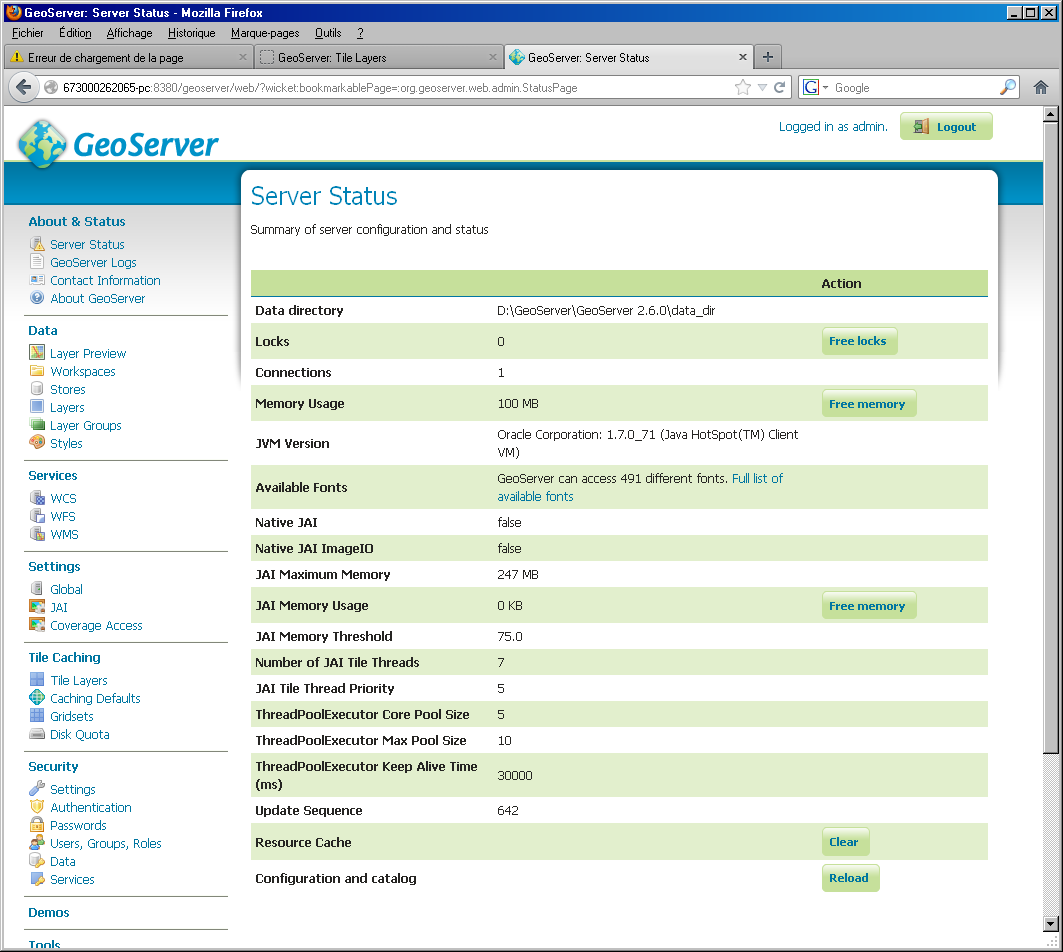
<tileHeight>256</tileHeight>

<tileWidth>256</tileWidth>

<yCoordinateFirst>false</yCoordinateFirst>

</gridSet>

After saving geowebcache.xml, you may start geoserver or ask it to reload its configuration (go to the “Server Status” page and click on the “Reload” button)



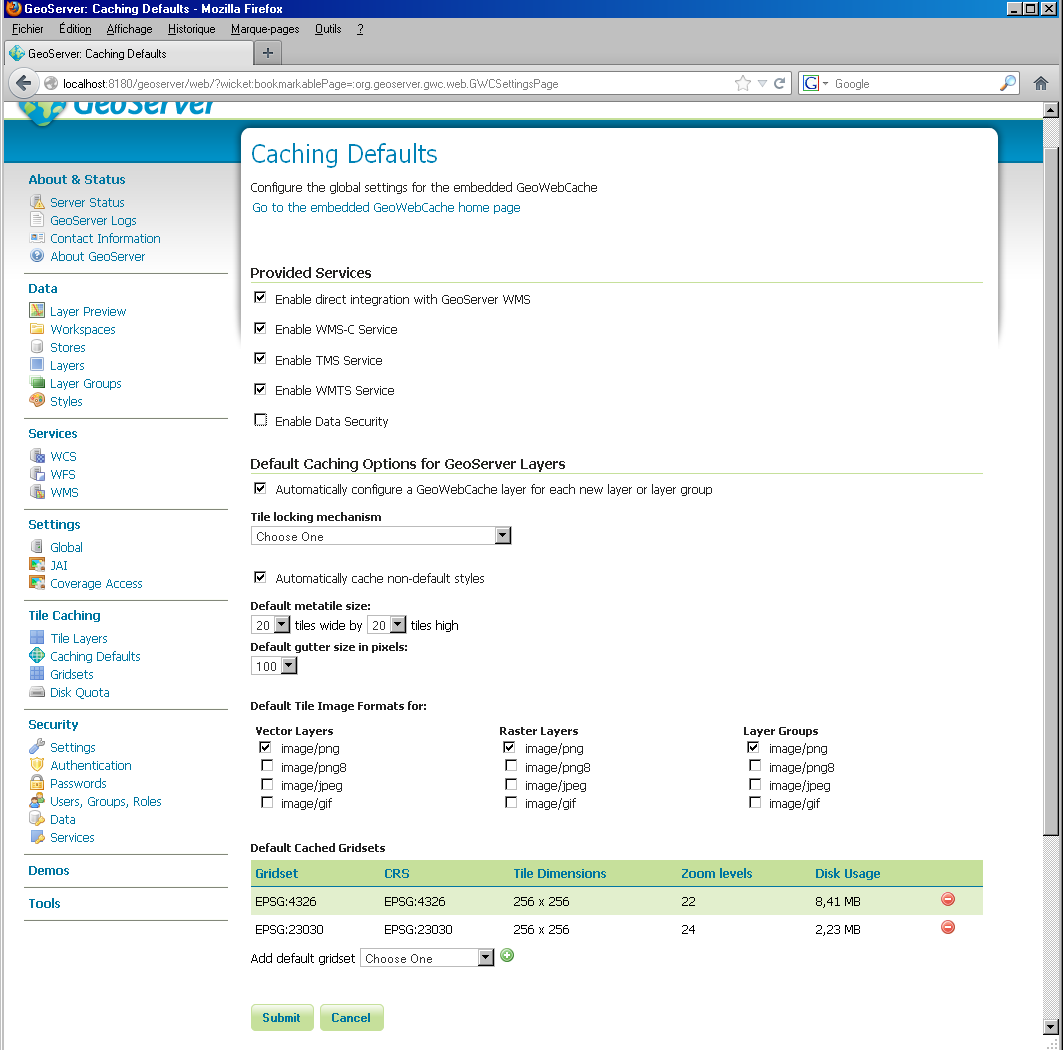
**Second Step: Define default parameter for tiling**

Go to the “Caching Default” page and check “Enable direct integration with GeoServer WMS”

Select a Default metatile size of 20x20

Set the Default gutter to 100

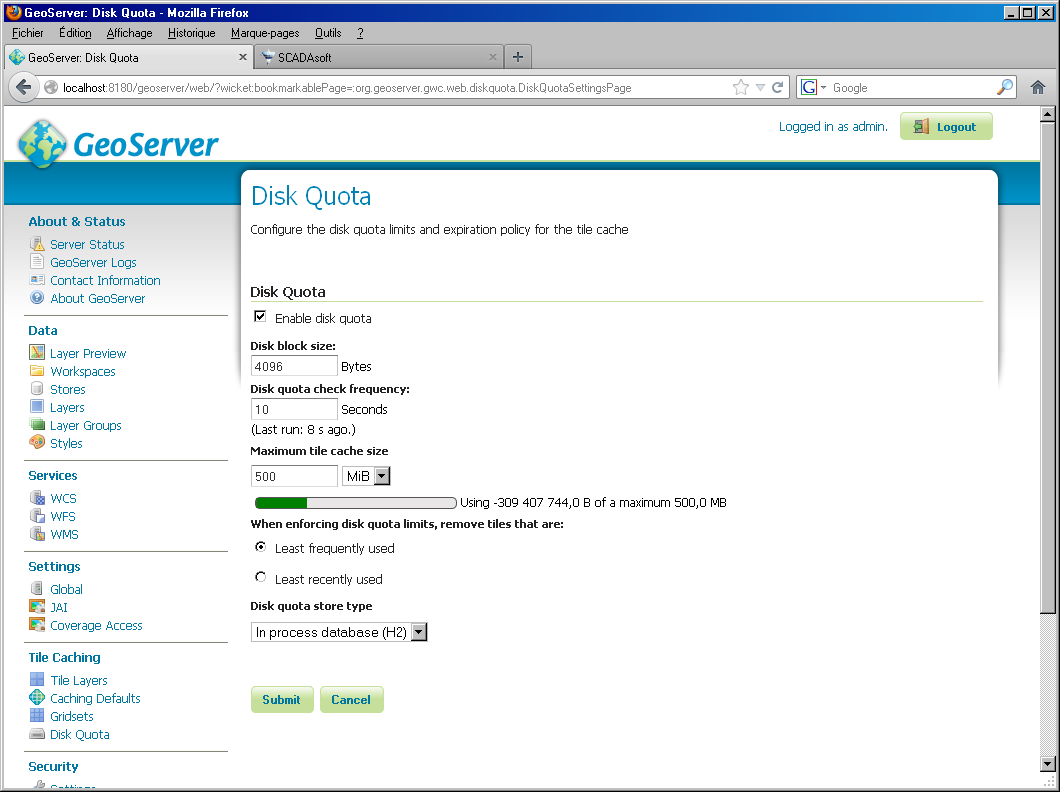
Add the “EPSG:23030” gridset to the list of “Default Gridsets”



**Third Step: activate Disk Quota**

Depending on the size of disk, where geoserver is running it is important to activate the “Disk Quota” for tiling.

Go on the “Disk Quota” page and check “Enable disk quota”, choose a size consistent with your empty space and click on the “Submit” button.

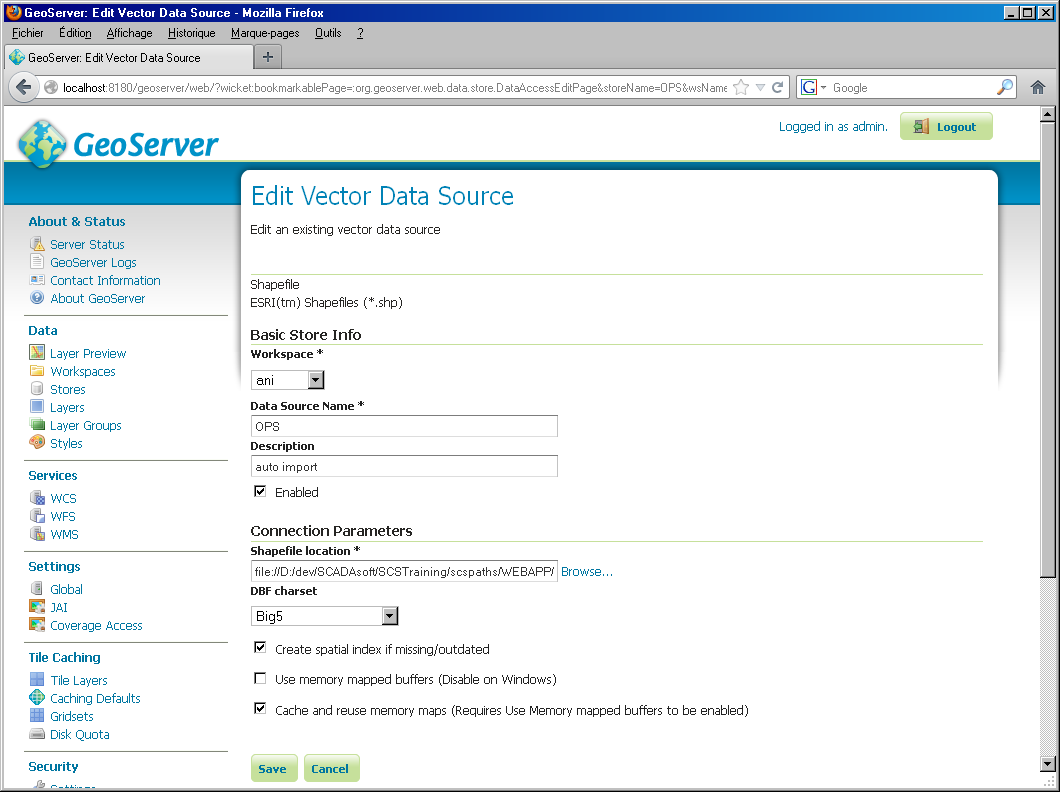


**Other issues:**

When important text data that are not plain ASCII, it is important to check that the DBF charset is properly set for your datastore.

On the “Stores” page, select your datastore and choose your charset, for example for Chinese you may use Big5.

Do not forget to click on “Save”.

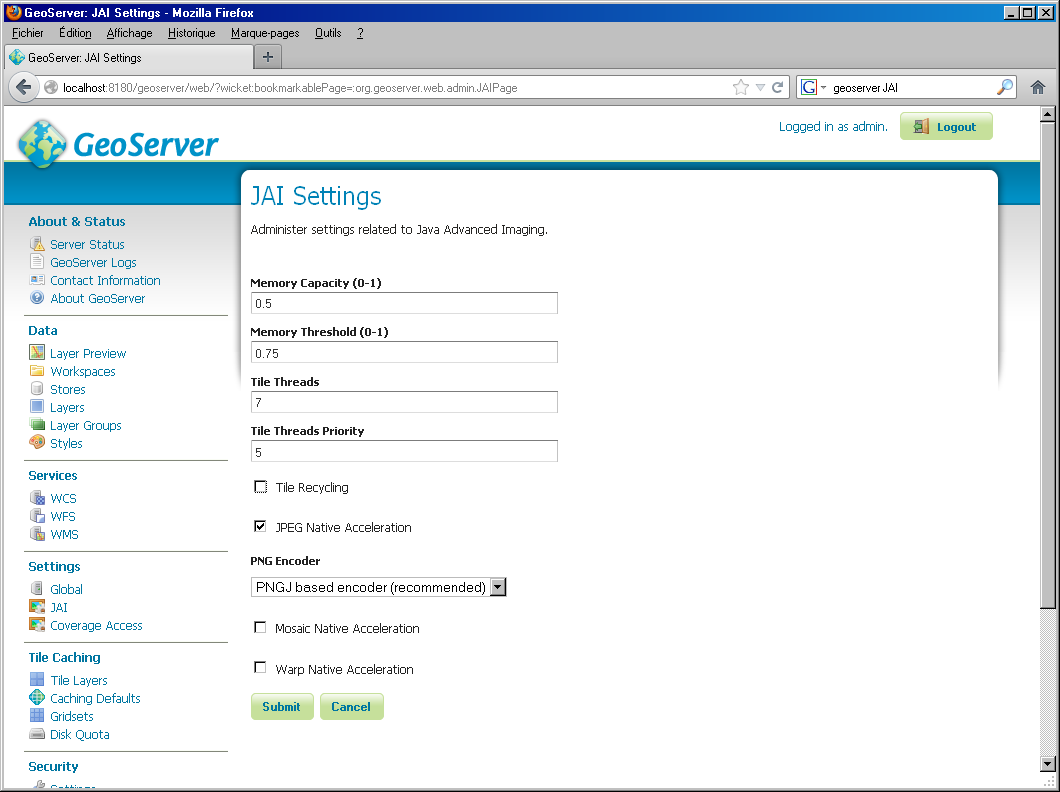


For better performance it is important to activate the JAI, see geoserver documentation for more information.

<http://docs.geoserver.org/stable/en/user/webadmin/server/JAI.html>

It is more important to have JAI, than a 64bits jvm.

Do not forget to check “JPEG Native Acceleration” in the “JAI” admin page.



When the tiling engine is properly setup, it is possible to have a faster access to tile by using a WMS url starting with

<http://geohost:8082/geoserver/gwc/service/wms>

instead of

http://localhost:8082/geoserver/wms