

Building an efficient and inexpensive PACS system

OsiriX - dcm4chee - JPEG2000

The latest version of OsiriX greatly improves compressed DICOM support, specifically JPEG2000¹ support. In this paper, we demonstrate how to build such system.

An efficient radiology workflow requires fast exchange of DICOM images from the server to the workstations, and fast reading of the images stored on each workstation. In real world, there are three bottlenecks to achieve these performances:

- 1) **Network bandwidth between workstations and PACS server.** The network is often based on 100 Mb² ethernet routers. This kind of network allows software to have a maximum speed of data transfer at about 11 MB/s. You can test your network bandwidth with the **Activity Monitor** application, located in your Utilities folder. A 11 MB/s connection allows you to transfer about 20 CT images per second, or about 50 CT images per second, if compressed in JPEG2000 LossLess³ algorithm (DICOM files are about 2.5x smaller).
- 2) **Disk speed.** Current disk speeds are quite good on workstations (but future SSD⁴ drives will be incredibly faster). Disk speed on the server side can be more problematic, if you selected an expensive RAID system, based on low speed disk (5400 RPM or less), or even worse a NAS⁵ system. A server needs to quickly load and transfer large amount of data, if there are multiple workstations requesting large exams at the same time. Again JPEG2000 LossLess compression will reduce the amount of data to load by about 3. The advantage of compression is double:
 - 1) You can store more files on your storage system.
 - 2) Files are smaller, hence the reading performance is better. You can read more files per second.Advice: don't choose a NAS for storing your DICOM image: NAS offers often slow reading/writing performances, and you don't need to have multiple computers sharing the NAS content, which is the primary feature of a NAS. Only the server will have to load and send these images, through DICOM protocol. If you use a RAID system, prefer RAID 0 (for faster speed access) or RAID 5 (for speed and back-up security). For a professional use, we recommend a hardware based RAID 5 system. A 4TB system, with JPEG2000 LossLess compression can store about 18 millions of CT images (about 9'000 studies of 2'000 images each).
- 3) **Processor speed.** JPEG2000 compression/decompression allows you to increase network transfer by 2.5x, and to reduce storage space by 2.5x, but it requires processing power for compression and decompression. With last version of OsiriX, decompression has been greatly improved, allowing to use all CPU cores at the same time. By implementing a JPEG2000 PACS system, be sure to have at least 2 cores on the server side: the server will have to compress images for storage, it will then send the compressed images to the OsiriX workstations. The CPU power is hence only necessary when adding new images to the server, not when sending images to the OsiriX clients. OsiriX clients will require CPU

¹ JPEG2000 is an open compression algorithm for images

² Transfer speed are described in Mb, Mega-Bit (Mb), in opposition to file size, described in Mega-Byte (MB). 1 Byte = 8 Bits

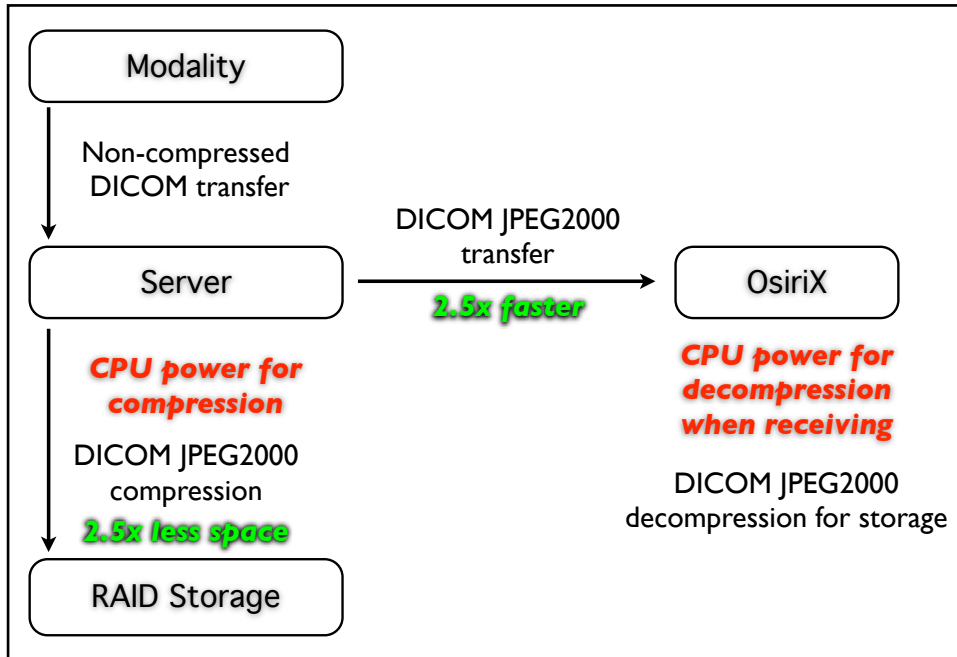
³ LossLess algorithm doesn't degrade image quality. Lossy algorithm degrades image quality but offer smaller file size. LossLess is about 3x smaller, Lossy algorithm are from 5x to 20x smaller.

⁴ SSD stands for Solid State Drive, based on RAM chips.

⁵ NAS stands for Network-attached storage. It uses ethernet TCP-IP, instead of USB2 or FireWire.

power when receiving or displaying the images on the workstations: all current Mac offers at least two cores. Choose a fast dual-core system (3 Ghz) or 4 or more cores (Mac Pro).

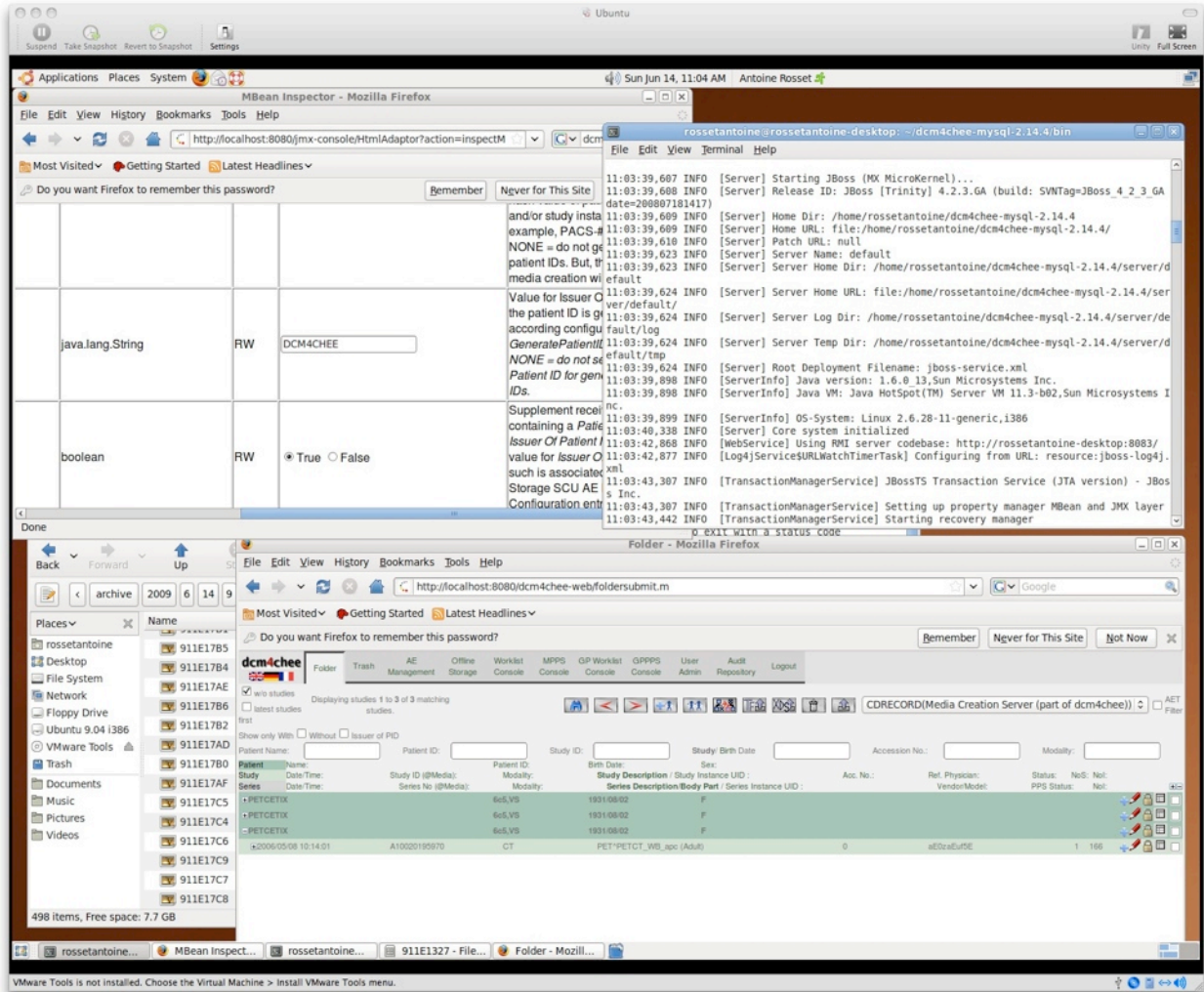
Set-Up



Two scenarios are possible, based on current open-source components:

- A. dcm4chee as PACS server, OsiriX as PACS workstation
- B. OsiriX as PACS server, OsiriX as PACS workstation

A. **dcm4chee as PACS server, OsiriX as PACS workstation.** dcm4chee is a powerful and robust server software to store and distribute DICOM files. It supports JPEG2000 compression. Sadly JPEG2000 compression is not supported on Mac OS, because the Java implementation for JPEG2000 is not available on Mac OS (jai imageio). We then recommend to install dcm4chee on a Linux based system, like Ubuntu or CentOS.



Software Configuration:

- 1) dcm4chee: Activate JPEG2000 compression for storing files on the server. Edit the jmx-console and activate the JPEG2000 compression: Change **NONE** to **J2KR**.

13. Optional: Configure image compression:

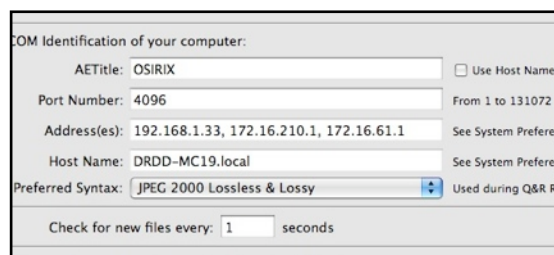
As noted above, this is not applicable for Mac OSX or 64 bit Windows.

At default configuration, received images are stored as received - in particular, no compression is performed. Lossless compression of received uncompressed images can be activated by attribute "CompressionRules" in the configuration page for the Storage SCP Service (service=StoreScp). E.g. set it to "JLSL", to compress all type of images received from any Storage SCU using JPEG-LS Lossless codec.



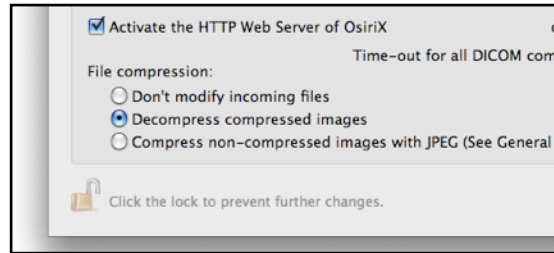
CompressionRules	java.lang.String	RW	J2KR	<p>List of image compression rules. Note that the value of the MaxConcurrentCodec attribute in the Compression Service also affects the on-the-fly compression and decompression during the store of images by the StoreSCP service.</p> <p>Format: <code>[condition]=value]compression</code> where condition—when to apply the compression rule. Separate multiple condition=value statements with a comma. The possible condition values are: calling - called AE title. Apply the rule to the called AE titles listed as values. called - called AE title. Apply the rule to the calling AE titles listed as values. pmi - photometric interpretation. cuid - SOP Class UID. Apply the rule for the SOP Class UIDs listed as values. imgtype - type of image. Apply the rule to the image types listed as values, such as RGB. value—the value of the condition. Each condition can have multiple values, separated by (pipe). For example, <code>{calling scu1 scu2}</code> compression—the type of compression to apply to each condition=value pair, separated by a space. Possible values are: NONE - do not apply compression. JPLL - JPEG Lossless JLSL - JPEG-LS Lossless J2KR - JPEG 2000 Lossless</p>
------------------	------------------	----	------	---

- 2) OsiriX: Activate JPEG2000 syntax on each OsiriX workstation. Select JPEG2000 Lossless & Lossy. See DICOM Listener Preferences.



- 3) OsiriX: Activate automatic decompression when receiving the images. By activating this feature, OsiriX will only decompress once the images, in background, allowing the best performance

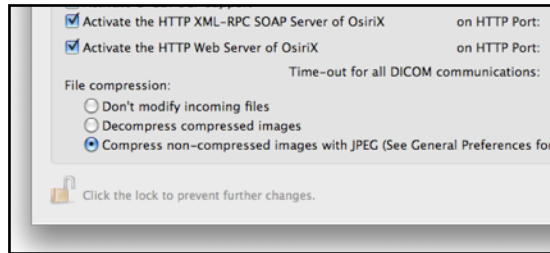
when having to display the images. See DICOM Listener Preferences.



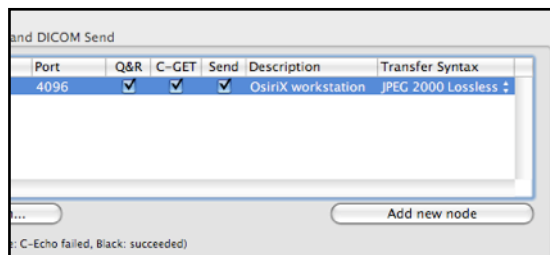
B. **OsiriX as PACS server, OsiriX as PACS workstation.** OsiriX can store files and communicate with other OsiriX workstations for exchanging files, through C-FIND, C-MOVE and C-GET SCU/SCP protocol. If you want to set-up a 'small' PACS, you can use OsiriX as a PACS server. The performances of OsiriX are not optimized when multiple PACS workstation try to load multiple studies at the same time, limiting the usage of OsiriX as a PACS server to 'small' deployment (less than 4 workstations).

Software Configuration:

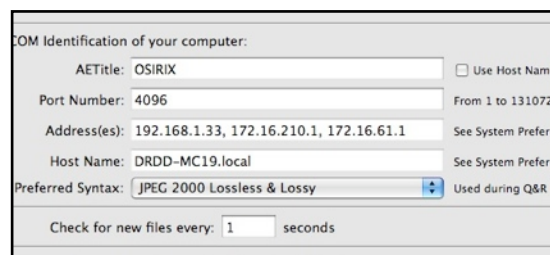
1) OsiriX Server. Activate JPEG2000 compression when receiving the image. See DICOM Listener Preferences.



2) OsiriX Server. Activate JPEG2000 syntax for each node. See Locations Preferences.

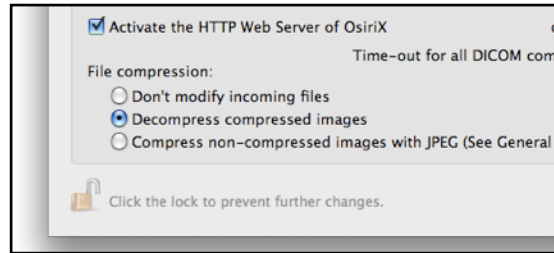


3) OsiriX Workstation: Activate JPEG2000 syntax on each OsiriX workstation. Select JPEG2000 Lossless & Lossy. See DICOM Listener Preferences.



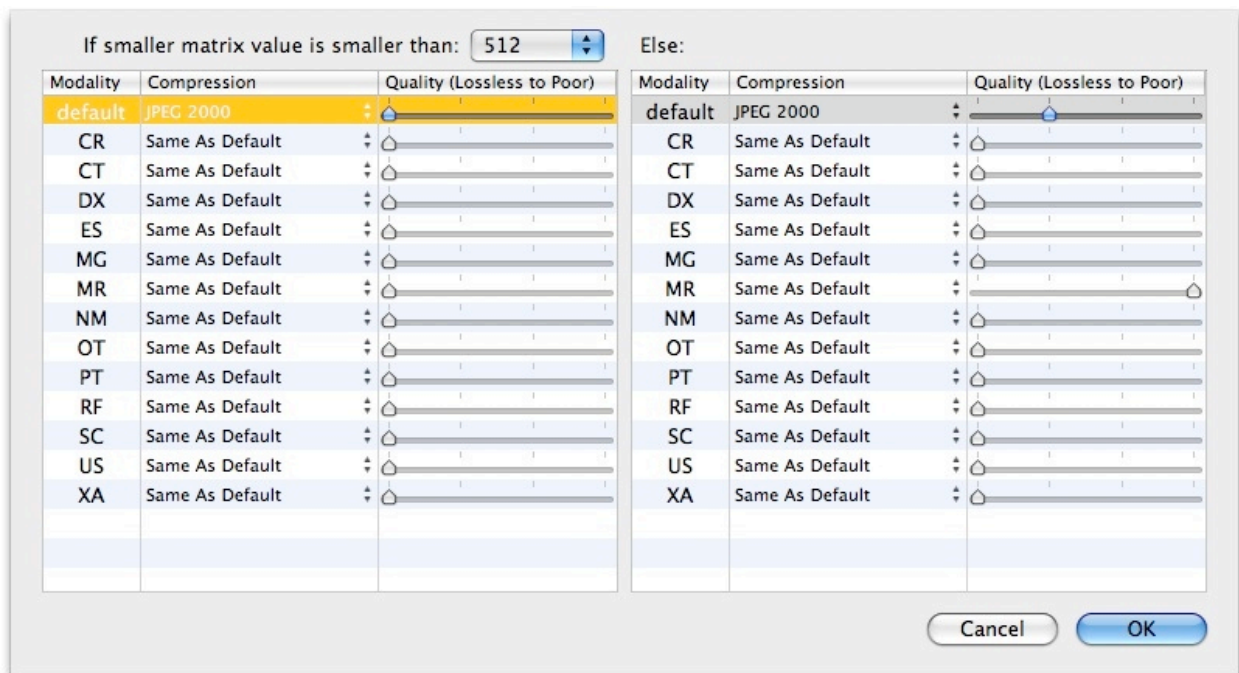
4) OsiriX Workstation: Activate automatic decompression when receiving the images. By activating this feature, OsiriX will only decompress once the images, in background, allowing the best

performance when having to display the images. See DICOM Listener Preferences.

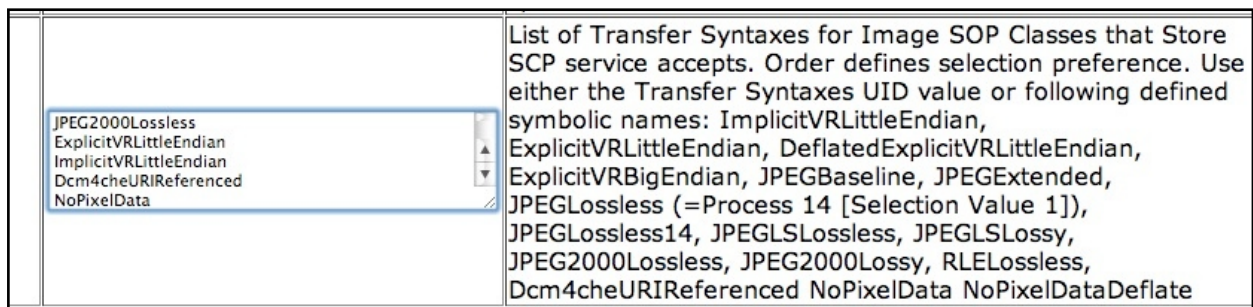


Comments

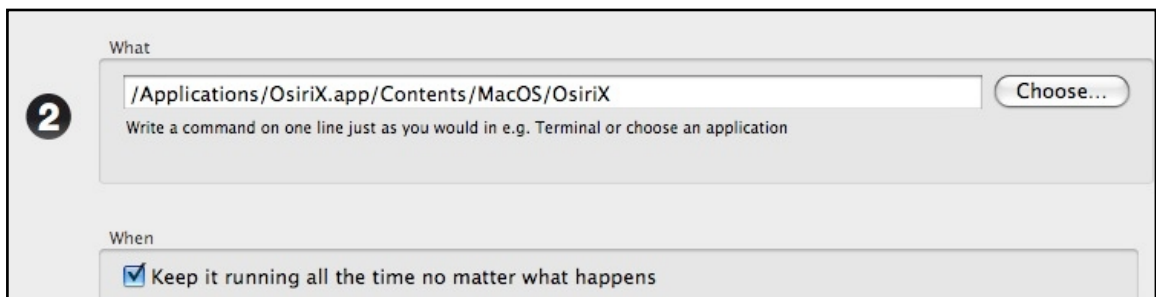
- This solution doesn't use 'on-demand compression/decompression', like compressing the images by the server during sending, or like decompressing the images on OsiriX, during viewing the images. Why? because 'on-demand compression/decompression' is slower than transfer speed: hence you lose the speed advantage of the compressed image transfer with these 'on-demand compression/decompression' methods. On the OsiriX workstation, 'on-demand decompression' when displaying the images can be quite slow on workstations with only 4 or less CPU cores, degrading the user experience.
- Don't use compression settings that are not **lossless** for small matrix images, like PET or small MR matrices. The quality degradation is too high on such images. You can configure JPEG2000 compression in General Preferences. The first position is **lossless** (about 2.5x smaller), the second position is **lossy** 5x smaller, the third position is **lossy** 10x smaller, the fourth position is **lossy** 15x smaller.



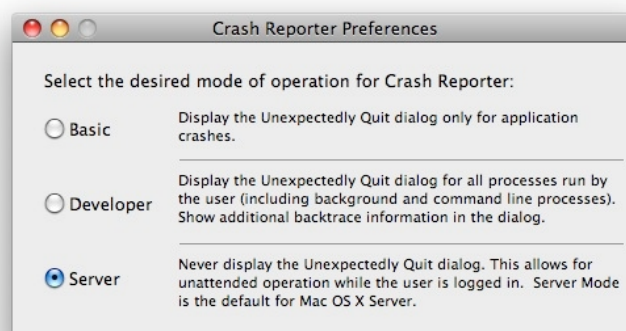
- Be sure that your network is 100% JPEG2000 configured, or you will lose all the advantages. For example, if you store your images in JPEG2000 format on your PACS server, but don't configure DICOM network protocol for JPEG2000, the server PACS will have to decompress the images, before sending them: slow compression for storage, slow decompression for preparing the images for the transfer and slow network transfer, because of file size.
- Don't mix multiple compression settings in your PACS network, for example, using at the same time some studies stored in JPEG2000 and other studies in JPEG. This can produce problems when DICOM nodes try to find a shared transfer syntax to exchange these images. dcm4chee will not, for example, decompress a JPEG2000 image and recompress it in JPEG, on-the-fly. Hence the communication will fail. To avoid this situation, we recommend to remove the JPEG support in dcm4chee, keep only the JPEG2000 Lossless support. We also recommend to remove the JPEG2000 Lossy compression in dcm4chee, because dcm4chee will not be able to convert between JPEG2000 Lossless <-> Lossy compression. This can create 'no presentation contexts error' blocking file exchanges between OsiriX and dcm4chee, if OsiriX is waiting for JPEG2000 Lossless images, and dcm4chee has only access to JPEG2000 Lossy images. Edit the store-scp service in the jmx console :



- If you use **dcm4chee**: optimize dcm4chee, for better robustness & performance:
 - Start dcm4chee as a startup item (copy and configure the dcm4chee_init_redhat.sh file, in the /etc/init.d folder). Use chkconfig (/sbin/chkconfig) to add the file to the startup system (this is related to the OS).
 - Reduce Console messages (remove `<appender-ref ref="CONSOLE"/>` and priority to ERROR instead of INFO). See the jboss-log4j.xml file, located in the dcm4chee folder.
 - Increase number of concurrent threads for compression to the number of available cores/processors. (jmx-console : service=CompressionService : MaxConcurrentCodec)
- If you use **OsiriX as a PACS server**:
 - Use Lingon (<http://http://tuppis.com/lingon/>) to create a 'always-up' application: 'Keep it running all the time' :



- Hide the errors messages from MacOS (CrashReporterPrefs application, from Developer tools) :



- Hide the errors messages from OsiriX - See Listener Preferences:

<input type="checkbox"/>	Accept only one association at a time (single process mode, poor performance)		
<input type="checkbox"/>	Publish my DICOM node over the network through Bonjour protocol		
<input type="checkbox"/>	Activate the DICOM Listener only if this user session is active (user switching).		
<input checked="" type="checkbox"/>	Don't display errors messages (Server Mode)		
<input type="checkbox"/>	Activate C-GET SCP support		
<input type="checkbox"/>	Activate the HTTP XML-RPC SOAP Server of OsiriX	on HTTP Port:	<input type="text" value="8080"/>
<input type="checkbox"/>	Activate the HTTP Web Server of OsiriX	on HTTP Port:	<input type="text" value="3333"/>

- Turn off messages about updates (OsiriX and plugins), see General Preferences.
- If you think, you don't have 'normal' performances. Look at the Log files on your OsiriX workstation and on your PACS server. If you have messages about compression/decompression, it means the transfer syntaxes are not correct. Normally, the server and the OsiriX should communicate without needs to compress/decompress data.
- In conclusion, with this set-up :
 - You can store 2.5x more files on your server
 - Your server can read them 2.5x faster (because 2.5x smaller), allowing more clients to access the server at the same time
 - Your server can transfer studies 2.5x faster to your PACS workstations
 - Welcome to the free & open-source world ! Enjoy !