# HTJ2K Transfer Syntax Support

## SUBMITTED BY Chris Hafey, Bill Wallace

## On Behalf of Working Group WG-04

# Introduction/Scope

The demand for remote access to DICOM images from a wide variety of devices is growing. Among the current transfer syntaxes, JPEG 2000 has several unique features which benefit remote viewing use cases such as resolution scalability, quality scalability and parallelism. A major drawback of JPEG2000 is its computational complexity which results in slower decode speeds and higher energy usage compared to other codecs. To address these issues, the JPEG standards committee adopted High Throughput JPEG 2000 (HTJ2K) in August of 2020. HTJ2K is a simple extension to the existing JPEG2000 standard that replaces block coder resulting in an order of magnitude speedup with a ~6-10% reduction in coding efficiency. Since HTJ2K builds on top of JPEG2000, it is straightforward to add support to existing libraries and transcode between the two block coders. Today there are several implementations available including an open source one that is ~50% faster than CharLS, an open-source implementation of JPEG-LS.

## Limitations of Current Standard

JPEG 2000 is the only transfer syntax which supports resolution scalability but is computationally complex. JPEG2000 also has limited parallelism and cannot take full advantage of higher core counts found in modern CPUs and GPUs.

Given these limitations, JPEG 2000 is not typically used with remote viewing use cases. Instead, viewers must use other transfer syntaxes that do not feature resolution scalability or parallelism or use a nonstandard codec that does. To address this gap, a new transfer syntax is needed for a codec that features resolution scalability, increased parallelism and fast decode times.

## Description of Proposal

This proposal is to add a HTJ2K lossy and lossless transfer syntax, and support for HTJ2K in the DICOMweb standard. In the DICOMweb area, the HTJ2K format will be added as a return type for pixel data, allowing for scalable resolution image access using HTTP byte range requests.

## Parts of Standard Affected

This work item will affect Part 5 and Part 18 of the standard.

## Resources & Time Line

About 8 people are active in Working Group 4. Bill Wallace and Chris Hafey have volunteered to work on writing this supplement. It is estimated that the work will take about 12 months. A first draft will be available by June 2022.

Members of WG-04 anticipate that four hours of Working Group Six meeting time will be required on each of four occasions during 2022 and 2023 to review and approve an early draft as well as public comment, letter ballot, and final text versions of the supplement.

***References***

* White Paper: High Throughput JPEG 2000 (HTJ2K) and the JPH format: a primer
  + <https://ds.jpeg.org/whitepapers/jpeg-htj2k-whitepaper.pdf>
* White Paper: High Throughput JPEG 2000 (HTJ2K): Algorithm, Performance and Potential
  + <https://www.htj2k.com/wp-content/uploads/white-paper.pdf>
* White Paper: Progressive Resolution Access in Medical Imaging
  + <https://docs.google.com/document/d/1hMv75h8g5a8EvIopucdhElKdu7QGF1PgpAK4J1Ahvp4/edit?usp=sharing>
* White Paper: Comparison of lossless decoding performance between JPEG-XL and HTJ2K (including estimates for speed with further optimizations)
  + <https://docs.google.com/document/d/1gS3cc7p00M3_Ub-0AC7shm16H_FCC2IyxqE2TCjpzUI/edit?usp=sharing>
* White Paper: Decoding High Throughput JPEG2000 (HTJ2K) on a GPU
  + <https://kakadusoftware.com/wp-content/uploads/ICIP2019_GPU.pdf>
* Comparison of WASM decode speeds for several DICOM codec implementations including CharLS (JPEG-LS), OpenJPEG (JPEG2000), libjxl (JPEG-XL) and OpenJPH (HTJ2K):
  + <https://docs.google.com/spreadsheets/d/1hCnRla2ZT6CVK6FuZC22On8QSqJGKsajn_TCcfkoI9k/edit?usp=sharing>
* 13 minute lightning talk on HTJ2K
  + <https://www.youtube.com/watch?v=Jvb3mUCyHH0>
* 5 minute video of using HTJ2K for video streaming
  + <https://www.w3.org/2021/03/media-production-workshop/talks/pierre-anthony-lemieux-media-production.html>
* Live Demo of HTJ2K decoding of WG-04 sample DICOM images in a web browser via WASM
  + <https://chafey.github.io/openjpegjs/test/browser/index.html>
* Open-source Implementations
  + OpenJPH: <https://github.com/aous72/OpenJPH>
  + OpenJPEG: <https://github.com/uclouvain/openjpeg>
  + GROK: <https://github.com/GrokImageCompression/grok>
  + OpenHTJ2K: <https://github.com/osamu620/OpenHTJ2K>
  + HTJ2K Reference software: <https://gitlab.com/wg1/htj2k-rs>
  + OpenHTS: <https://github.com/sandflow/openhtjs>
  + MatHTJ2k: <https://github.com/osamu620/MatHTJ2K>
* Commercial Implementations:
  + Kakadu: <https://kakadusoftware.com>
* SMPTE adoption of HTJ2K for Hollywood post production images
  + <https://github.com/SMPTE/st2067-21>

TODO: Add references to white papers, timing, implementations, use in entertainment, evidence of adoption, etc