User Manual for Kakadu's "kdu_show" Tool (Windows and Mac OSX)

— last updated for version v6.2

David Taubman, UNSW

1 Introduction

This brief document is designed to accompany the two demonstration viewers/browsers which come with Kakadu. Internally, these viewers are named "kdu_winshow" and "kdu_macshow", but the executables are both known as "kdu_show" on their respective operating systems. All source code to each of these tools is provided with licensed copies of Kakadu and may be used to aid in the construction of even more elaborate applications. However, both viewers already possess a very full list of features.

As of Kakadu version 6.2, the two viewers offer almost identical functionality and pretty much identical menu options and accelerators – except that the CMD key is used in place of CTRL for many of the Mac accelerators, in keeping with convention. The source code for both viewers is also very close, with most files and objects having identical names for "kdu_winshow" and "kdu_macshow" except that the former are prefixed by "*kdws_*" while the latter are prefixed by "*kdms_*". These viewers are implemented on top of powerful platform-independent Kakadu API's, such as '*kdu_region_compositor*', '*jpx_source*', '*mj2_source*' and the like.

Briefly, these viewers include the following features:

- Highly efficient viewing of arbitrarily large images, image compositions, and videos on the local file system, with zoom, pan, rotate, etc.
- Highly efficient remote viewing of large images and image compositions via the JPIP interactive communication protocol – remote browsing of video is not yet enabled.
- Ability to review and describe all codestream parameter attributes associated with compressed content.
- Ability to open multiple windows at once, to the same or different image sources, whether locally or remotely located – a convenient window duplication function makes it very simple to open a new window into a

current region of interest. When used with JPIP remote browsing, the data received from the server is shared by all relevant windows and advanced multiple JPIP channel features are exploited wherever available, for maximum efficiency.

- Ability to display the metadata file structure associated with any JP2-family source (JP2, JPX, MJ2).
- Metadata catalog side-bar for JP2/JPX files allows efficient navigation through richly annotated images, using both imagery and metadata information to drive the navigation – works beautifully with JPIP remote image browsing as well.
- Metadata editor, capable of providing rich annotation information to be saved in the JPX file format.

2 Basic Operations

2.1 Opening Files

The File menu may be used to open and close files. Alternatively, you may find the "ctrl-o" (Mac: "cmd-o") accelerator key useful for opening files quickly. Opening a new file automatically closes any existing open file. You can supply the name of a file to "*kdu_show*" on the command line, in which case that file will be opened on start up and its dimensions will guide the initial size of the view window.

The "*kdu_show*" tool can open JP2 files, JPX files, unwrapped JPEG2000 codestreams and even Motion JPEG2000 (MJ2) video files.

2.2 Panning, Zooming and Re-orienting

The "*kdu_show*" tool started out as a sophisticated demonstration application to showcase some of Kakadu's capabilities for incremental decompression of large images, based on spatial regions of interest (view ports), as well as Kakadu's ability to decompress images in a geometrically transformed domain. In fact, Kakadu's codestream management architecture has always allowed compressed JPEG2000 images to be accessed as though they had been compressed as smaller image, rotated and/or cropped versions of themselves. As a result, the core "*kdu_show*" features have always been those of panning, zooming, rotating and flipping a viewport into the compressed image.

Panning may be accomplished via the scroll-bars, or using the arrow keys. The image may also be dragged around with the mouse while the left mouse button is depressed and the shift key held down.

Zooming may be accomplished via the View menu or with the "Z" (zoom in) and "ctrl-Z" (zoom out) accelerators. By default, zooming is centred about the middle of the current viewport, but you may define an alternate focus box using the method described below. Zooming out is accomplished using the multi-resolution attributes of the discrete wavelet transform (DWT). As such, the degree to which you can zoom out is limited by the number of DWT levels created when the image was compressed. Although this limit could readily be circumvented by rescaling the decompressed image, it provides useful feedback to content creators, making them aware of the limitations associated with using too few DWT levels.

Rotation and flipping may be accomplished via the View menu or with the "[", "]", "_" and "|" accelerators. Note, however, that any rotation which involves image transposition (not just horizontal and vertical flipping) may slightly alter the displayed imagery if it happens to have been compressed using JPEG2000's reversible processing path (Kakadu's "Creversible=yes" attribute). The reason for this is that horizontal and vertical aspects of the wavelet transform do not completely commute when the reversible transform is used, due to LSB rounding effects. Kakadu accomplishes rotation by decompressing the imagery in a reverse order and performing the wavelet transform in an alternate order, rather than just decompressing the image and then subsequently reorienting it.

The application automatically resizes its window to match the image size, unless the image is too large to be viewed in the current window, in which case a restricted viewport is displayed. You may adjust the window size via the usual resize handles. Alternatively, and often more conveniently, you may enlarge or shrink the view window quickly using the "W" and "S" accelerator keys.

Special note on zooming: JPX files can contain compositions of codestream imagery, where each composed piece has arbitrary scale factors. The "*kdu_show*" tool handles such imagery correctly, regardless of the implied scaling factors, but there can be no one global scaling factor which results in optimal rendering of the entire composition. Also, the policy in "*kdu_show*" is to keep the global scaling factor constant as you navigate to different compositing layers and/or composited frames. As a result, you may find that the scaling factor applied to a particular codestream image component of interest is not ideal. Ideal scaling factors are integer expansion factors and power of 2 decimation factors. Whenever you use the "z" (zoom in) and "ctrl-z" (zoom out) function, the zoom factor is automatically adjusted to one which is ideal for the top-most or dominant imagery found within the view window or focus box (see below). However, you may also find the "alt-z" (optimize zoom) function very useful. It makes the smallest possible change to the current global scaling factor so as to optimally render the content which is dominant within the view window or the focus box (see below). Keep this function in mind when browsing complex JPX content.

Special note on zooming vs. display scaling: The "*kdu_show*" tool also provides you with a special "*Scale X2*" feature, which may be accessed via the View menu. While this may seem very similar to zooming, there is an important distinction. The "*Scale X2*" feature (accelerator key "X") causes each rendered image pixel to be displayed using a 2x2 block of display pixels, regardless of the current zoom setting. For example, zooming out from the image (so that it is being rendered at 50% resolution) causes the highest frequency subbands and the last stage of DWT synthesis to be dropped from the compressed imagery. With the "*Scale X2*" you can visualize the resulting imagery more clearly, especially on displays with very small pixels. When zooming in beyond 100% resolution, the "*Scale X2*" essentially just gives you another factor of 2 in zoom, but it allows the operating system to use GPU resources to do the scaling, which can result in faster rendering of zoomed imagery to the display.

2.3 Focus Boxes

Focus boxes have at least 3 uses in "*kdu_show*": 1) they define a centre for zooming; 2) they define the region of interest during remote image browsing with JPIP (see Section 4); and 3) they define spatial regions to be associated with metadata added using the methods described in Section 5.

To define a focus box, hold the left mouse button down while dragging the mouse to define the region of interest. If you change your mind after depressing the mouse button, you may hit the escape key to cancel the focus box being defined, without affecting any pre-existing focus box.

To remove a focus box, you may: 1) use the Focus menu; or 2) hit the "F" accelerator key.

By default, the focus box will be displayed using both a highlighting technique and a dashed outline. The highlighting technique makes the rest of the image slightly darker, and the focus box slightly lighter. You can turn off the highlighting by using the Focus menu, or hitting the "H" accelerator key.

You can widen or shrink an existing focus box using the Focus menu, or the "shift-W" and "shift-S" accelerators.

You can pan the focus box using the arrow keys, while holding down the shift key. Basically, the accelerators which affect the focus box are identical to those used to manipulate the view window, except that the shift key is held down.

2.4 Quality Layers

JPEG2000 codestreams may contain multiple embedded image qualities, which form the foundation of quality progressive remote image browsing. By default, local files are opened at maximum quality, but it can be useful to see the visual quality associated with each compressed quality layer. To reduce the number of quality layers you may use the View menu or the "<" accelerator. You may increase the quality again using the View menu or the ">" accelerator. These manipulations automatically change the status bar to one which displays the number of quality layers which are in use.

When playing video content, you may well find that displaying all of the available quality layers is too slow. Reducing the quality typically increases rendering speed dramatically, especially when the original content was losslessly compressed. Of course, this is only possible if multiple quality layers are available in the source content.

2.5 Navigating Between Image Elements

The "kdu_show" tool has 3 fundamental modes of image display, as follows:

Single Component Mode: In this mode, only a single image component is displayed as a greyscale image. Image components typically correspond to colour channels (e.g., red, green, blue, or luminance, and chrominance channels), but they may correspond to alpha blending channels, palette indices, or anything else.

To enter single component mode, use the View/Navigate menu or hit either of the "1" or "+" accelerator keys. While in this mode, you may navigate forward and backward within the image components of a given codestream by using the View menu or the "+" and "-" accelerators.

While in single component mode, you may navigate amongst the codestreams in the file (JPX and MJ2 files may have any number of codestreams) using the RETURN and BACKSPACE keys. Alternatively, you may use the View menu.

Single Compositing Layer Mode: In this mode, only one compositing layer is displayed, usually as a colour image, or alpha blended against a solid white background. A JP2 file contains exactly one compositing layer, which is the rendered colour image. A JPX file may contain any number of compositing layers. A raw codestream is treated as having exactly one compositing layer, corresponding to our best guess regarding the intended display image.

A Motion JPEG2000 (MJ2) file may contain one or more video tracks, each of which is treated as a compositing layer. This is sensible since MJ2 tracks can be composed onto a richer surface in the same way that JPX compositing layers can be composed. The fundamental difference between MJ2 video tracks as compositing layers and JPX compositing layers is that the former typically involves multiple images (video frames), all with the same dimensions, whereas the latter involves only a single image.

To enter single compositing layer mode, use the View/Navigate menu or hit the "L" accelerator key. While in this mode, you may navigate forward and backward amongst JPX compositing layers or amongst frames of a current MJ2 video track, by using the RETURN and BACKSPACE keys or the View menu. To move to a different MJ2 video track, use the View menu.

Composited Image Mode: In this mode, the application displays what it believes to be the intended image. For JP2 files and unwrapped raw codestreams, this is identical to the result displayed in single compositing layer mode. If a JPX file contains a composition box, however, the composition instructions are used to build the composited image which was intended by the content creator. For MJ2 files, this mode is currently equivalent to single compositing layer mode, so you can only view one video track at a time. However, this is a limitation only of the "*kdu_show*" demo application; the underlying '*kdu_region_compositor*' object, which does all the rendering work, fully supports composed video.

When a new file is opened, "*kdu_show*" starts out in the composited image mode, but if you have changed the mode to single component or single compositing layer mode, you may get back to composited image mode by using the View menu or hitting the "C" accelerator.

While in composited image mode, you may navigate forward and backward between composited frames (animations contain multiple frames, each of which is a composited image) by using the RETURN and BACKSPACE keys, or via the View/Navigate menu.

3 Status Information

Some summary information is displayed at the bottom of the application window on its status bar. The status bar is divided into fields. The information displayed in these fields also depends upon the status mode, which may be toggled using the Status menu, or via the "ctrl-T" (Mac: "cmd-T") accelerator key. In particular, you can currently toggle between the following status modes:

1. Left field displays current zoom factor (percent); centre field displays current image element; right field displays number of quality layers. The interpretations of these quantities are as given above for the Windows viewer.
2. Left and centre fields as above; right field displays the full size of the image at the current resolution, measured in pixels – i.e., exactly the same as the Windows viewer.
3. Left and centre fields as above (current zoom factor and image element); right field displays the amount of working memory used by the codestream management machinery (for all codestreams currently loaded into memory). As described for the Windows viewer, this value is generally very small if the codestreams are enabled for random access (precincts, PLT marker segments and a layer-minor progression order).
4. (Available during remote image browsing via JPIP) Left field displays the JPIP client status (e.g., "connecting", "receiving ...", "disconnected"); centre field as above; right field displays communication information, such as current data rate and total received data.

4 Remote Image Browsing with JPIP

By and large, images may be accessed remotely from a JPIP server (such as that implemented by the "*kdu_server*" utility) exactly as if they were local files.

Rather than experiencing significant delays while the image downloads, you will instead observe incrementally improving quality as image data is incrementally transferred by the server. JPIP servers focus the transmitted data around the region defined by the focus window so the quality in that region increases more rapidly. If no focus window is defined, it is taken to be the current view window. Once sufficient data has been received to reconstruct the image within the focus window without loss, at the current display resolution, the server enters the idle mode and will not transmit any further data until the image region, resolution (zoom factor), codestream, image components, compositing layer or animation frame are changed.

It is important to realize that the viewing tool is only loosely (asynchronously) coupled to the client component and, ultimately, to the server utility. You are always free to pan, zoom and otherwise navigate the display to any part of the image, regardless of whether any data has been received from the server to render that region. After some time, the server will adjust its transmission pattern to serve the new image region/resolution which is currently of interest. Of course, if the server has disconnected, no amount of navigation will bring in new data, so you will generally be left with non-uniform image quality which reflects the amount of time spent browsing in different image regions.

It is also important to realize that the JPIP server is allowed to shrink the region of the image for which it will serve data, if the region is too large to be served in a quality progressive fashion without consuming excessive server memory resources. If this happens, the changed region will be reflected on the view window through a new focus box. You may move this modified focus box around, to change the region to which the server's transmission is customized, but you may not enlarge the focus box without having the server force it back down again to its size limit.

4.1 Regular Image Browsing

To open a remote image, use the "File->Open JPIP URL" menu item ("ctrl-U" or "cmd-U" accelerators), or start "*kdu_show*" with an appropriate URL on the command line. The URL dialog box provides several fields for you to fill out. For regular image browsing, enter the name of the file you want to access in the "*Resource or request string*" field and enter the name or IP address of the server machine in the "*Server*" field. You may append an optional colon-separated port number to the server name or address, if your server is not listening on the default port 80.

One of three closely related JPIP transfer protocols may be selected via the "*Transport protocol*" drop-down list. For the most efficient communication, the "*http-tcp*" option is recommended; however, this requires the establishment of both an HTTP and a regular TCP channel, which might not be permitted by some institutional firewalls. In this case, select "*http*" as the next best option. The "*none*" option is primarily for experimental purposes. It is much less efficient, especially for the server.

If your organization requires all external traffic to flow through an HTTP proxy, select the "*http*" option and put the name or address of your HTTP proxy server in the "*Proxy server*" field – you may need to consult your system administrator to find out the name of your HTTP proxy. Proxies reduce the communication efficiency and responsiveness, so use them only if necessary.

You may elect to have all your image browsing results saved in a local cache directory, by entering the name of this directory into the "*Cache directory*" field. The advantage of this is that when you open the same image again on the same server, or a compatible server (one which assigns the same unique target identifier to the image resource), the previous browsing results will be reused and that data will not be sent over again by the server.

An alternate way to open an image on a remote JPIP server is to supply the URL on the command line. The URL must commence with either "*http://*" or "*jpip://*" as the protocol identifier prefix, followed by the server host name (or IP address), a single "/" character, and then the resource string. The syntax for the resource string and the server name are identical to those used by the URL dialog box. The alternate "*jpip://*" protocol identifier is provided so that the "*kdu_show*" utility can be fired up automatically from a web browser whenever a URL with this protocol identifier is encountered. In fact "*kdu_show*" registers itself with the Windows Internet Explorer (Safari on the Mac) as the target for such URL's, the first time it is ever used. You should be aware that when URLs are supplied on the command line, "*kdu_show*" obtains the remaining information (proxy server, cache directory, protocol variant) from persistent preference information, based on the last values supplied via the URL dialog. For this reason, it is advisable to use this dialog to configure the application first, before opening URLs from the command line or by clicking on JPIP URLs on a web page.

You may explicitly disconnect from a JPIP server, without actually closing the image, by using the "File->Disconnect JPIP request queue" menu item.

4.2 One-Shot Image Browsing

To implement the functionality of an image browser, "*kdu_show*" uses Kakadu's '*kdu_client*' object to generate a sequence of much more complex HTTP requests than that supplied via a typical command line URL. These much richer requests are generated using either HTTP POST or HTTP GET with query parameters appended to the URL, separated by the usual "?" query separator.

You may supply any of these more complicated URLs directly by including the "?" query separator and the query fields of interest on the command line or in the URL dialog. When you do so, ongoing communications will either be interactive or non-interactive, depending on whether you check the "Force interactive" box in the URL dialog. If the box is left unchecked, only the supplied request will be delivered to the server and no amount of interactive navigation within the image will cause any further requests to be sent to the server. The server will continue to transmit data relevant to the original request until all relevant information has been sent or the server times out or is disconnected. As a result, interactive navigation and image data delivery are entirely disconnected processes; you may be receiving data relevant to a small image region at high resolution, but be viewing the image at low resolution (zoom factor) or vice-versa. You may even be viewing a completely different codestream within the data source and so not see the effects of newly arrived image data.

The main purpose of this mode of behaviour is for server debugging.

To use single shot URLs (or to provide any explicit query options in the "Resource or request string" field of the URL dialog), you will need to be familiar with the JPIP syntax. It is not the purpose of this document to describe this syntax, but the JPIP committee draft may be downloaded from "http://www.jpeg.org/CDs15444.html." You may also examine the JPIP requests delivered to the Kakadu server during normal interactive image browsing by supplying the '`-record`' command line option to "*kdu_server*". Note that the request fields you enter into a URL dialog box may be hex-hex encoded for robust transport to the server, meaning that some characters may be converted to strings of the form "%DD", where D is a hexadecimal digit.

5 Properties, Metadata, Overlays and Editing

You may use "*kdu_show*" to view the coding parameter attributes and other properties of the codestream (or codestreams). You can also display, add and edit metadata in a JP2 or JPX file, resaving edited metadata in the JPX format – only a limited subset of metadata can be saved in the more restrictive JP2 format. From Kakadu version 6.2, an extremely useful metadata catalog appears as a sidebar within the main window associated with any JP2/JPX image which contains text labels.

5.1 Viewing Codestream Properties

Use the "File->Properties" menu item or the "ctrl-P" accelerator (Mac: "cmd-P") to view the coding attributes and other properties of the current codestream. Where more than one codestream is involved in the current image display, the application chooses one of them to display its attributes. Codestream properties include comment marker segments, coding parameter marker segments and tiling attributes.

You may double-click (single click on the MAC) on any attribute to read a detailed description of that attribute's interpretation, as provided by the Kakadu core system. In the MAC viewer, this detailed description is conveniently formatted together with the current values of the parameter attribute in question.

5.2 Viewing the File Structure with Metashow

Additional attributes are provided by the file format in which codestreams are embedded. To examine the file structure itself, use the Metadata menu or the "ctrl-M" accelerator (Mac: "cmd-M") to activate the "*metashow*" tool. You may click on various nodes within the metashow tool's tree view (browser view on the MAC) to obtain additional details or to have the main window navigate to corresponding image elements. For example, clicking (double-clicking in the MAC version) on the second compositing layer header box will cause the main window to enter single compositing layer mode and display the second compositing layer. Similarly, clicking on a contiguous codestream box or codestream header box will cause the main application to enter single component mode, while clicking on the composition box, if any, will cause the application to enter the composited image mode.

Clicking on some leaf nodes in the metadata tree will cause their contents to be expanded as text, while for other boxes a generic hex dump is provided.

The Mac version of "*metashow*" tool provides you with a number of additional features. If you select a metadata item which corresponds to an editable JP2/JPX metadata node, the "*Open in Metadata Editor*" button will be activated, allowing you to edit the node, add descendants, etc. Although edited metadata does not become part of the file structure until the file is saved and reloaded, both Mac and Windows versions of "*metashow*" use strike-through and text colouring conventions to highlight elements in the file structure which have been deleted or changed by the editor.

5.3 Adding New Metadata

To add metadata use the Metadata menu or the "ctrl-A" (Mac: "cmd-A") accelerator. This opens up the metadata editing dialog, which allows you to enter text labels or to import externally edited label or XML box contents. If no focus box has been selected, the metadata will be associated by default with the top-most current image entity (compositing layer or codestream), depending on the current image display mode (see Section 2.5).

If a focus box was defined at the time when you hit the "ctrl-A" (resp. "cmd-A") accelerator, the metadata will be associated by default with the top-most codestream which overlaps with the focus region; it will also be associated explicitly with that region. These associations are achieved by embedding appropriate number list and ROI description boxes in the JPX file (when you save the edited result).

You may alter any of the above associations within the metadata editing dialog. The editor explicitly excludes options which are illegal; for example, metadata associated with a focus region must also be associated with at least one codestream, not just with compositing layers. You will see that you can add multiple associations, but it is up to you to ensure that this is meaningful.

You can add descendant nodes to newly added metadata, via the "*Add child*" button, and navigate amongst the descendants you add in this way. For new metadata which is associated with a spatial region of interest, a label text node will automatically be added as a descendant, containing the initial string "*<new label>*". This is to remind you that regions of interest should probably have associated labels, and for backward compatibility with the much more limited editor provided with early versions of "*kdu_show*". However, you can always delete this descendant if you like.

5.4 Editing Existing Metadata

When the metadata editor is opened, using any of the above methods, it is initialized with a restricted set of metadata nodes which are considered relevant for editing. You can modify, delete or add descendants to any of these nodes. If the editor was opened by right/ctrl-clicking a region of interest, displayed in the flashing or static overlay mode, only the ROI node and its descendants are included in the editing set. If the editor was opened by right/ctrl-clicking on any other part of the image view, the editing set consists of all metadata nodes which are associated with the image entities (codestreams or compositing layers) under the mouse cursor, plus all global metadata. You can navigate amongst these metadata nodes via the "*Previous sibling*" and "*Next sibling*" buttons.

The metadata editor works closely with the catalog side-bar. The latter displays only labels and links (cross-references), but automatically updates (or pops up) in response to editing operations. At the top of the catalog side-bar is a "paste-bar" which can be used for copy/cut and paste operations. You can paste metadata directly into the catalog side-bar or within the metadata editor. If you paste something which has been cut (using "ctrl-X" resp. "cmd-X") it will be moved to the new location without altering any of the original contents or descendants; the original items will not be touched until they are actually pasted. Items can also be copied to the paste-bar as *links.* Links show up in the paste-bar in blue text with underlining; when pasted, they create a link to the original item within the metadata structure, rather than a copy of the item. This is extremely useful for building up semantic associations within the metadata. The "*kdu_show*" tool exploits links very extensively for navigation and discovering associations which can be used to determine which region-of-interest overlays to display when the "Metadata->Restrict overlays" option ("ctrl-R" resp. "cmd-R") is used.

The metadata editor also provides a button for synchronizing the "*metashow*" window's contents with the metadata node currently being edited, so you can examine how the node is represented within the overall file structure. Of course, this is only possible for metadata nodes which are already present in the file structure, since the editor does not change the file structure until the file is actually saved. Similarly, for label metadata, there is also a button for synchronizing with the *catalog side-bar view.*

The metadata editor has a lot of additional capabilities, which are either self explanatory or explained through tooltips which pop up when the mouse pointer hovers over interesting buttons.

5.5 Metadata Overlays

Once you have some spatially associated metadata, you will notice that "*kdu_show*" highlights its presence via partially transparent overlays. You may use the Metadata menu to control the appearance of such overlays. Most usefully, the "ctrl-alt-T" (Mac: "cmd-alt-T") accelerator may be used to toggle the overlay mode between *flashing*, *static*, and *no overlays.* You may also control the lightness/heaviness of overlays via the Metadata menu. To declutter the overlay display, you can set the minimum size of elements which will appear on overlays (set the threshold via the Metadata menu) or you can use the "Restrict overlays to catalog selection" option in the "Metadata" menu (or via "ctrl-R" resp. "cmd-R").

In the flashing or static overlay mode, whenever you left-click on an overlay region which is associated with a text label, the relevant entry will automatically be selected for you in the catalog side-bar. This is actually a more powerful feature than you might first imagine; where a region of interest is associated with multiple text labels, an algorithm attempts to select the most appropriate one in the catalog side-bar, based on recent navigation history, together with semantic associations formed by parent-child relationships and links. A nice way to construct multilingual metadata labels is to associate every region of interest with a "Grouping link" to a top-level index label which holds the name of the language (preferably written in the language of interest). Under each such Grouping link, add child labels to describe the metadata in the relevant language. If you do this, the automatic selection algorithm mentioned above, will automatically find the description of the clicked region of interest which uses the same language as the most recently selected item in the catalog side-bar.

Right clicking on an overlay region opens up the metadata editor, as does left clicking with the ctrl key held down. You can do the same thing using the "ctrl-E" (Mac: "cmd-E") accelerator.

It is possible to create a link directly to a region of interest (as opposed to a label which might describe the region of interest). To do this, use the "ctrl-L" (Mac: "cmd-L") accelerator (this generally copies metadata to the catalog paste-bar as links) while the mouse pointer is over the region of interest overlay within the image view. You need to be sure that the image view has keyboard focus, as opposed to the catalog side-bar – you can do this by clicking anywhere in the image view first. If successful, your region of interest coordinates will be displayed in the catalog paste-bar using link-text conventions. You can then paste a link to the region of interest into the catalog side-bar (use "ctrl-V" or "cmd-V") or from within the metadata editor. These links show up within the catalog side-bar, enabling a user to navigate directly to the region of interest.

5.6 The Catalog Side-bar

When you open a JPX image with associated metadata in the form of text labels (JPX label boxes), an auxiliary catalog panel is automatically opened on the right hand side of the image view. The same thing happens when you first add such metadata to an image which previously had none. You can always hide the catalog panel via the Metadata menu (or "ctrl-shift-C" resp. "cmd-shift-C" accelerator), but it is very convenient to keep around. The catalog contains a hierarchically organized record of all textual metadata in the source, under three broad categories:

Index Labels: These correspond to text labels which are not descended (via a JPX "asoc" box hierarchy) from any "nlst" box (a *number-list* box identifies image entities, meaning codestreams, compositing layers, or the "rendered result" of a composition), nor found within any codestream or compositing layer header box. As such, index labels are not directly associated with any image entity, nor can they be associated with any region of interest, since regions of interest inherently belong to image entities.

Despite their lack of any associations, index labels can be very useful for image navigation. This is because, you can always add links from any item in the index, where the target of the link can be another metadata item which does have image entity or region-of-interest associations. For example, you could build a geographical place name index, organized by counties, cities, suburbs, streets, etc., and then add links from entries in the index to region-of-interest associated metadata or just directly to the regions of interest. Clicking on the link within the metadata catalog, navigates the catalog to any associated label text while navigating the image view to the relevant region of interest, within the relevant image, compositing layer or composited result.

Image-Entity Labels These correspond to text labels which are descended from an "nlst" box or found within a codestream header box or compositing layer header box, but not descended from an "roid" (region-of-interest description) box. These text labels are interpreted as being directly associated with whole image entities, as opposed to spatial regions within an image entity. Again, you can add links from such labels to other, region-specific metadata, if you like. However, this probably makes less sense that the use of cross-references within a top-level catalog, as described above.

Region-Specific Labels These correspond to text labels which are descended from an "roid" (region-of-interest description) box. As such, they are directly associated with a specific spatial region of interest. As noted in the previous section, regions of interest must themselves be defined with respect to a codestream, but they may also be associated with compositing layers or a composited result, which contains the reference codestream (or codestreams). These associations are taken together, when interpreting the associated metadata and using it for navigating within the image view.

5.6.1 Hierarchical Associations within the Catalog

The catalog side-bar maintains the hierarchical association between its labels, as identified by the embedding of metadata within JPX "asoc" boxes. Extra levels in the hierarchy are inserted, if required, if the number of sibling labels encountered at any point grows too large to conveniently display within the catalog panel. This is done by introducing automatically creating alphabetical groupings. For example, the three sibling labels, "ants", "apples" and "Amazing Apes" might all be grouped under the heading "<A..." if is deemed to be

beneficial for user navigation within the catalog panel. These extra levels in the catalog hierarchy do not correspond to actual metadata – they are introduced only for navigational convenience.

It is worth mentioning that the catalog side-bar machinery updates itself automatically whenever metadata is added, deleted or modified via the metadata editor. The catalog side-bar also updates itself dynamically as more data arrives from a remote server, during remote image browsing via JPIP. All aspects of the catalog side-bar are dynamically adjusted, including the hierarchical association structure and the automatically generated summary groupings, such as "A..." for "ants", "apples" and "Amazing Apes". You can test this out yourself by adding a large number of metadata labels via the editor and watching what happens ... now try deleting some of the metadata again. The catalog side-bar does its best to keep the location of any selected items in the panel at the same location visually, regardless of how the structure changes around them. This is very important during JPIP image browsing, since the dynamic arrival of data from the server may frequently reconfigure the catalog – but this is not annoying in practice, so long as the selected item always stays where it was.

5.6.2 Navigating with the Catalog

Undoubtedly, the most useful feature of the metadata catalog is that it can be used to quickly navigate to associated imagery. All you have to do is double click any label in the catalog and the view window will adjust itself, if required, to show the associated codestream, compositing layer or animation frame, as appropriate. Moreover, for labels which are associated with spatial regions of interest, a focus box is automatically placed around the relevant spatial region's bounding box, and the view is panned and zoomed, as required, to make the region clearly visible.

Once you have selected a label in the metadata catalog panel, you can use the Metadata menu or the "ctrl-E" (Mac: "cmd-E") accelerator to open that label within the metadata editor. You can also right-click or ctrl-left-click the item in the catalog as an even more convenient way to open the item in the metadata editor. For region-associated labels, the editor opens at the region-of-interest node ("roid" box) in which the label is embedded, since this is the place where you should modify the region's shape or image entity associations, if you wish to. You can always navigate to the label text via the editor's "*Descendants*" button.

Navigation works in the reverse direction as well, taking you from a region-of-interest in the image view to associated metadata within the catalog, simply by clicking within the region's overlay in the image view.

5.6.3 Links and the Catalog

As mentioned in the introduction to this section, it is useful to construct links from unnassociated (index) or less associated (image entity) metadata entries to region-specific metadata. It is also useful to create links from the highly associated metadata back to the less associated index data. Kakadu uses the terms *"Alternate-child link"* and *"Alternate-parent link"* to refer to these different type of links. While the associations are not strictly important, the semantic difference between these two types of links is the following: if a metadata item could meaningfully be included as a descendant the item which links to it, an alternate-child link should be used – effectively, alternate-child links address the fact that metadata can often be meaningfully interpreted as descended from different parent items. Alternate-parent links provide a mechanism for the multiply descended child item to point back to its "virtual" parents.

Building bi-directional links is particularly useful for remote image browsing via JPIP, since interactively following links sends the relevant metadata request commands to the JPIP server to fetch data which might not yet be available. For example, when accessing a small region of the image view, you may receive region-of-interest metadata from the server, which shows up as overlays on the image view. Clicking on the region-of-interest overlays, may reveal other associated metadata in the catalog side-bar (this should typically be the case in a well-constructed set of associated metadata). This, in turn, may contain links to other metadata (e.g., alternate parent associations within the metadata ontology) and following these links may take the interactive user to other categories of data from which new region-of-interest associations may be discovered. Double clicking region-of-interest associated metadata in the catalog adjusts the image view and causes new window-of-interest requests to be sent to the JPIP server; and so forth.

Kakadu identifies a third type of semantic link relationship, which we term a "*Grouping link.*" Unlike Alternate-child and Alternate-parent links, grouping links are simultaneously links to other metadata in the catalog and parents (or branches) within the metadata hierarchy – i.e., they can have descendants. Semantically, all the descendants of a grouping link can be interpreted as members of the same group as the descendants of any other grouping link which has the same link target. You can use grouping links in lots of ways, but one important example is language groups. To construct language groups, first create a label in the index category (i.e., an unassociated label) for each language you are interested in (e.g., "English", "French", "Japanese", etc.). Then whenever you want to create a multi-lingual set of associated metadata for some entity (e.g., a region-of-interest), add a grouping link to each language in question under the entity you want to describe and then add the language-specific descriptive metadata under each such grouping link. This works beautifully in "*kdu_show*", for all kinds of navigational purposes. You will find that clicking the entity in question (e.g., a region-of-interest) takes you to the most appropriate description in the catalog, depending on the language most recently selected by the user.

To help you work with these different types of link semantics, the catalog side-bar colour-codes them differently and even adds suggestive arrows to the link text (the arrows won't show up on Windows XP unfortunately). To add links to the existing metadata, you first copy the link target to the catalog paste-bar as a link – use the Metadata catalog or the "ctrl-L" (Mac: "cmd-L") accelerator – and then paste the link to the location of interest. When you do so, a dialog box will appear to ask you what type of link you want to add; this dialog box contains its own explanation of the different types of links, so you don't have to remember everything you read here.

6 Saving Files

You may save images back out of "*kdu_show*" via the File menu options. This is of interest particularly if you have edited an image's metadata, or if the image has been obtained from a remote JPIP server – in that case, the saved file will contain whatever information has been received from the server up to the point when it is saved.

You may edit and resave a file over the top of the file you already have open. In this case, however, the file is actually saved under a different name, formed by appending the "~" character to the name of the currently open file, rather than overwriting it. When you close the open file, any such saved copy is automatically renamed so as to replace the one you had open. This happens even if you close the application (e.g., by hitting the "ctrl-Q" resp. "cmd-Q" accelerator). The File menu provides a convenient "Save and Reload" option, which combines the above operations of saving over the current file and then re-opening it.

You can save files as raw codestreams (choose a suffix of ".j2c" or ".j2k"), simple JP2 files (choose a suffix of ".jp2") or JPX files (choose a suffix of ".jpx" or ".jpf"). If you are currently viewing a motion JPEG2000 file (MJ2), you can save current frame as a JP2 file or a raw codestream, but you cannot currently save the entire movie.

The File menu provides two additional variations on the "Save As" option, which enable you to force the use of external links to the codestreams or force the embedding of codestreams in the saved file. In most cases, all relevant codestreams are already embedded in the images you open with "*kdu_show*" and the "Save As" option retains this embedding. However, JPX files can contain links to codestreams which may be found in other files; the "Save As" option preserves such links rather than embedding the codestreams. "Save As Linked" forces the use of links, meaning that codestreams which are currently embedded in the file you have open will be recorded as links into that file. Linked codestreams are very useful while you are editing metadata for a very large image or collection of images, since each time you save your results, only the metadata structure need actually be written to disk. Of course, with linked codestreams you have to be careful not to delete or overwrite the file which contains the linked codestreams.

"*kdu_show*" does its best to keep track of links into any of the files it has open and prevent you from saving over any of those files, but it can't be aware of files it does not have open.