Extending the SIAP with Logical Names

R. Williams, 28 April 2004

In the SIAP protocol, there are two parts: a metadata service that returns a VOTable of images and their metadata, and each row of that table contains a URL to a data service that will fetch an actual image in FITS or JPEG or some other format.

The SIAP protocol defines the nature of the columns of the table that is returned by the metadata service. What we considering in the following is to specify a new, optional column, and the associated semantics.

1. The Proposal

The proposal is to allow a new column called “Logical Name” (LN) which is string valued.

(1.1) The string should be built from the same character set that is used in identifier components (is this right Ray?)

alpha | digits | “-“ | “_” | “.” | “~” | “$” | “&” | “=” | “+”

Specifically, this choice of characters will enable the LN to be used as part of a Posix file name if desired by the consumer. This is useful for bulk use of SIAP services where lots of files are stored; and it may not be possible to deconstruct the URL to produce a suitable file name.

(1.2) If two rows of the table have the same LN, then we can assume they are in some sense “the same” image. Comparison is case-sensitive. Interpretation of this is not strictly defined, but some guidelines follow in the next section.

2. Guidelines

SIAP providers and consumers may ignore these guidelines and still be legal.

Recalling the metadata that MUST be present in each row, let us abbreviate:

- “Scale” is the scale pair from VOX:Image_Scale
- “Naxes” is the scale pair from “VOX:Image_Naxes”
- “Format” is the format from VOX:Image_Format
- “Bandpass” is the bundle of bandpass metadata
- “URL” is from the VOX:Image_AccessReference

The guidelines assume an SIAP table that has two rows that have a common LN. We compare the rest of the metadata in the rows to reach “reasonable” conclusions as specified below.

(2.1) If all the metadata in the two rows is the same except for the URLs, a consumer may assume that exactly the same data (same byte stream) can be fetched from the URLs.

There may be “normal” client-side processing to effect this, for example those URLs ending in .gz would imply client-side decompression is needed before the streams would be the same.

The URLs may have different hostnames—thus providing support for replica services.

The URLs may have different protocols, such as http:// and srb://, thus providing support for alternate delivery methods.

(2.2) If all the metadata is the same except for the URLs and the Formats, then a consumer may assume that the same data is present but in different image formats. So there could be, for example, a FITS and a JPEG version of the same image. The JPEG version may have a lower information content, but is assumed to be exactly co-registered with the FITS version.

(2.3) If all the metadata is the same except for the URLs and the Scales and Naxes, then a consumer may assume that the same information is present but at a different scale. However, if the relevant ratios of the scales and the Naxes are not right, then one image might be a cutout of another, or there might be a mistake.

(2.4) Guidelines (2.2) and (2.3) can be combined so that we have a way to find, for example, a quicklook Jpeg version of a Fits image.

(2.5) If all the metadata is the same except for the URLs and the Bandpass packages, then a consumer may assume that these images are exactly coregistered (stackable) but in different bandpass. A set of three monochrome images could, for example, be identified and combined into a color image.