WWW.IGIPZ.PAN.PL

Anatomy of AIMS communication

Aviso Internet Map Server whitepaper, part 1

Wojciech Pomianowski

2001-12-15

Submitting Request to AIMS

Aviso Internet Map Server communicates with external world via HTTP (Hypertext Transfer Protocol) and provides content readable in a web browser. Users wanting to get data from AIMS (called clients) must send a command either by typing it in web browser address bar or clicking on a link on a page. Typical command has the following form:

```
http://maps.igipz.pan.pl/aims/aims.dll?REQUEST=GetMap&MAP=Scotland_Land_Use
```

First part of command is server address:

```
http://maps.igipz.pan.pl/aims/aims.dll
```

and remaining string after ? sign is AIMS-specific request:

```
REQUEST=GetMap&MAP=Scotland_Land_Use
```

which consists of request type (in this case GetMap) and a subcomand MAP=Scotland_Land_Use. AIMS command syntax conforms to URI (Uniform Resource Identifier) standard (see RFC 3986). Request and response terms used below as well as all other communication issues conform to HTTP (Hypertext Transfer Protocol) 1.1 standard (see RFC 2616). Both standards are supported by all major web browsers. Readers are encouraged to follow the specifications.

GetMap Request

The request shown above is of "GetMap" type and has a subcommand specifying "Scotland_Land_Use" map name. This name is not a filename but an alias which may be solved by AIMS only. Actual map files may be buried deep in storage (or sit on another machine) and their names are hidden from the outside view.

In response to GetMap request AIMS sends back an image of the map in standard MIME image/png format. Web browser displays this data just like ordinary image file. The key point to understand is that GetMap requests behave exactly the same like images, so while editing HTML page you can insert a map just like a photograph by using command in place of a filename. The most basic request does not control any aspect of map display and AIMS would use default values stored in map. This can be changed by supplying additional subcommands, for example a geographical position of map center:

```
REQUEST=GetMap&MAP=Poland_Hipso&POS=19.99,53.2
```

Other map parameters may be controlled the same way but there is clear division of responsibilities between AIMS and AVISO. AIMS commands control mainly navigational parameters because this allows for panning and zooming via web browser. Other properties like graphical appearance, layers composition is left to map author working in AVISO. Let's see how this division works for the SCALE parameter. Each AVISO map is assigned a scale which is used when map opens. AVISO author can change scale by numeric value or by navigation control like slider or mouse wheel. AIMS client can also request a different scale. However, AVISO author can define minimum and maximum values, which AIMS client cannot exceed or change. AVISO author may additionaly define a set of preferred scales (called *scale pool*) and these may be available to AIMS client, again as a unchangeable option.

GetMap Request Subcommands

subcommand	*meaning*
MAP=<map name>	Specifies map name.
SCALE=<number>	Sets map scale. Additional minimum and maximum constraints saved in map also apply.

WIDTH=<pixels>
HEIGHT=<pixels>
	Sets map image size in pixels. Additional systemwide constraint also applies.

POS=<x-coord>,<y-coord>	Sets map center in geographic coordinates (decimal degrees). Geographic range constraints saved in map also apply.
CENTER=<x-coord>,<y-coord>	Sets map center in internal, coded coordinates. Geographic range constraints saved in map also apply.
GROUP=<group1>, <group2> …	Limits display to specific groups of layers defined within a map.

These subcommands work with static single map display. Sometimes we may want to handle user clicking on a map or dragging the map. The problem is how do we calculate geographic position of the click or drag offset when everything happening in user's browser is expressed in pixel coordinates? AIMS solves this without additional plug-ins or any other code at client side. The idea is that these calculations should be done at server side. Interaction parameters can be submitted to AIMS with special subcommands and effect subsequent map generation.

subcommand	*meaning*
PXOFFS=<x-pix>,<y-pix>	Moves map center given in CENTER by pixel values.
SCALEOFFS=<integer>	Modifies value given in SCALE to match next/previous value from scale pool. Useful for mouse wheel zooming.
SOLVEMAP=…	

SOLVEMAP subcommand is nedded in more complicated case - for instance, user may be presented with an index or introductory map and then transferred to another map or the same map zoomed in. SOLVEMAP parameters carry pixel-based click coordinates along with an old map parameters (other subcommands in the same request specify details of a new map). AIMS calculates position on the old map and replies with the new map centered on this position.

Legend Request

Legend request makes AIMS return an image of the legend for a given map, in the same image format as map itself. The syntax is as follows:

```
REQUEST=Legend&MAP=Scotland_Land_Use&SCALE=500000
```

Though the legend could be possibly provided as an inherent part of map image, AIMS takes more flexible route. Firstly, a legend may be too big to fit on the map. Then, a separate legend is easier to compose on the page with map and may be even displayed on dedicated page.
Required subcommand is MAP=<map name> and optional subcommand is SCALE=<number>. The latter is necessary for AVISO maps with layers switched on/off depending on scale. When map is re-scaled, legend content follows map content.

GetPage request

GetMap and Legend requests alone can bring the essential images to your page, but to get more sophisticated HTML pages with additional map-related elements AIMS must have more control over HTML pages itself. For example, we might need to display numeric value of current map scale. In

situation when user is allowed to zoom on map, this information cannot be put on page as fixed number. It must be generated by AIMS based on zoom request, so both map and page must be processed side by side. This is where GetPage is necessary.
GetPage is a request for HTML page hosting the map. AIMS responds with MIME image/html data forming the page. It is enough here to say that the page will have enhanced content. To fully understand what options are available and how to make them work together with GetPage request you should read "Crafting Map Website with AIMS" chapter.

A Chain of Requests

GetPage does not return map image. It is intended to be the initiator of transfer which, due to HTTP nature, must come in pieces. In usual circumstances it is the first of a sequence of three requests. After the page is loaded into client's browser, it will contain a reference to the map and sometimes to the legend. So the browser will send GetMap and Legend requests soon after. When the other two pieces of data arrive from AIMS, web browser will coalesce complete page with map and legend. You may use Fiddler Web Debugging Proxy to view flow of requests and responses between your web browser and IGiPZ Map Server site.

GetPage Request Subcommands

GetPage has only one subcommand: PAGE=<page name>, which specifies the HTML file name. However, it accepts all subcomands of GetMap described earlier, so actual commands resemble GetMap commands, for instance:

```
REQUEST=GetPage&PAGE=sg&MAP=Scotland_Land_Use&SCALE=6000000&POS=19.53,51.53
```

All supplied map parameters are used only to generate extra page elements. Of course, these elements must be in sync with map image, otherwise user would read wrong map title or zoom to wrong scale . It is important to make sure that GetPage parameters match GetMap parameters.

Other Requests

Except for three essential requests, AIMS responds to diagnostics request and steering request, which support remote statistics and remote control of AIMS. These requests are of administrator’s interest only and are submitted as numeric codes unknown to the public.