

The Woolz Internet Image Protocol (IIP) Server (Version 1.1.4)

Zsolt Husz ^{*} Bill Hill [†]

March 14, 2016

Abstract

The Woolz Internet Imaging Protocol (IIP) server (WlzIIPsrv) extends the IIP to allow views of 3D Woolz objects with multiple components. This manual describes these extensions for both users and developers of the WlzIIPsrv.

^{*}Supported by NIH under grant #1R01MH070370-01A2.

[†]bill.hill@igmm.ed.ac.uk

Contents

Title Page	i
Abstract	i
Contents	ii
1 Introduction	1
2 Extension overview	1
2.1 Woolz object specification	4
2.2 Setting sectioning and query parameters	5
2.3 Extended object reference	11
2.4 Image Processing Expressions	19
3 HTML query examples	22
4 WlzIIPsrv coding	23
4.1 Architecture	23
4.2 IIPsrv and WlzIIPsrv beside the IIP specification	25
5 Woolz IIP Proxy	25
5.1 WlzIIPProxy options	27
6 WlzIIPsrv installation	28
6.1 Install FastCGI	28
6.2 Source code	28
6.3 Compiling	28
6.4 Customisable parameters	29
7 WlzIIPProxy installation	29
8 Acknowledgement	30
A fcgi configuration	30

1 Introduction

The Internet Imaging Protocol IIP provides quick access to large image databases allowing large images to be viewed remotely. It makes use of tile-based delivery so that panning and zooming within large images is possible without needing to download the entire image. The computational requirements of the client machine for viewing are low, allowing viewers to be implemented within web browsers. However, it is limited to 2D images. In biomedical imaging (and many other fields) 3D images are routinely used. A set of extensions to the IIP server was developed within the Edinburgh Mouse Atlas Project [?] to allow the remote visualization of sections through large 3D objects. These extensions make use of many of the features of Woolz, an image processing system with it's own efficient image representation [?].

This document assumes understanding of the IIP [?].

For clarity, in the rest of this document IIPsrv will refer to the original IIP server, with code version 0.9.7 [?], WlzIIPsrv will refer to the extended server, while IIP will point to the protocol standard of [?].

The coordinate system convention used in this document is shown in Figure 1. The object coordinate is defined by the object and the section coordinates result after sectioning. Then, the translated section coordinates with the section's minimum at (0,0) are the display coordinates. Further, the section is divided into non-overlapping tiles covering the whole section. Tiles are numbered with 0, 1, 2, etc., with the 0th coordinates matching the display coordinates. The 1st, 2nd, etc. tiles continue from left to right and top to bottom. A client can visualise a section with reduced viewing area. This has the view coordinates and is client dependent.

2 Extension overview

In addition to the IIPsrv commands, a WlzIIPsrv command specifies an object, sets the viewing section parameters and requests image or meta data of the object. Tile request are similar to the TIFF requests, however for a Woolz object the resolution number used in **JTL/TIL** is ignored.

The extended command list that sets query parameters are summarised in Table 1, while Table 2 shows the new objects that may be queried with the **OBJ** command.

Command	Purpose	Syntax
CVT	Request an image to be returned as a composed image. CVT accepts now also PNG and WLZ format requests.	CVT= <i>format</i>
DST	Specify the distance of the sectioning plane	DST= <i>dis</i>
FXP	Specify the fixed point of the viewing section rotation	FXP= <i>X,Y,Z</i>
FXT	Specify the second fixed point of the viewing section rotation	FXT= <i>X,Y,Z</i>
MAP	Define a colour or grey value mapping	MAP= <i>mspec[,mspec[,mspec[,mspec]]]</i> where <i>mspec=t,il,iu,ol,ou,p0,p1</i>
MOD	Specify the projection mode	MOD= <i>mode</i>
PAB	Specify the 3D query point absolute in the object coordinate	PAB= <i>X,Y,Z</i>
PIT	Specify the pitch angle of the sectioning rotation	PIT= <i>angle</i>
PRL	Specify the 2D query point relative in tile or display or tile coordinate	PRL= <i>T,X,Y</i>
PTL	Retrieve a tile as a PNG image	PTL= <i>res,tile</i>
DPT	Specify the rendering depth	DPT= <i>depth</i>
RMD	Specify the rendering mode	RMD= <i>mode</i>
ROL	Specify the roll angle of the sectioning rotation	ROL= <i>angle</i>
SCL	Specify the scale used in the sectioning transformation	SCL= <i>scale</i>
SEL	Specify a component of a compound object to be displayed and its colour. See 2.4 .	SEL= <i>E,R,G,B,A</i>
UPV	Specify the up vector for the UP_IS_UP mode	UPV= <i>X,Y,Z</i>
WLZ	Specify the Woolz object	WLZ= <i>path</i>
YAW	Specify the yaw angle of the sectioning rotation	YAW= <i>angle</i>

Table 1: Extended command overview

Object	Purpose
IIP-server	Identify if WLZ-IIP is running
Max-size	The size of the section
Tile-size	The size of a tile
Wlz-true-voxel-size	The voxel size of the object
Wlz-volume	The volume of the object
Wlz-n-component	The number of components in a compound object
Wlz-distance-range	The range of the sectioning plane distance
Wlz-sectioning-angles	The pitch, yaw and roll angles of of the sectioning plane
Wlz-coordinate-3D	The 3D coordinates defined in 2D by the PRL command
Wlz-grey-stats	Simple statistics of the Woolz object image values.
Wlz-grey-value	The grey or RGB value of a point specified either the PRL or the PAB commands
Wlz-transformed-coordinate-3d	The display coordinates and displacement from the sectioning plane of a 3D point defined by PAB
Wlz-foreground-objects	The components of the compound object that 2D/3D query point is a foreground

Table 2: Extended object overview

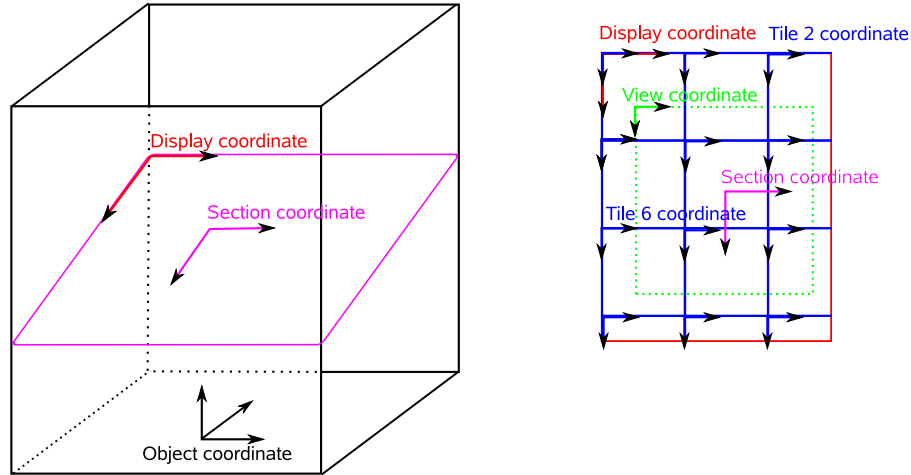


Figure 1: Coordinate systems

2.1 Woolz object specification

WLZ	Purpose	Specify the Woolz object. It is similar to the FIF command, however instead of loading Pyramidal Tiled TIFF images, it specifies a Woolz object. Compared with FIF , the Woolz object is not loaded until a later operation requires this. The Woolz objects are cached. 3D domain and value objects, and compound objects with 3D domain or value components are supported. For a compound object the domain of the first object must include all other object domains. If file name ends with .gz , gzipped Woolz object is expected.
	Syntax	WLZ= <i>path</i>
	Input Parameters	PATH <i>path</i> Full path of the Woolz object.
	Response	WLZ returns nothing when accompanied by other commands
	Example	\Rightarrow WLZ =/home/bill/pics/lobster3d.wlz
	Notes	Equivalent if FIF [?, p.25]

2.2 Setting sectioning and query parameters

The following commands are optional and can be used in any combination.

DST	Purpose	Specify the distance of the sectioning plane
	Syntax	DST= <i>dis</i>
	Input Parameters	FLOAT <i>dis</i> The distance to the sectioning plane
	Response	none
	Example	⇒DST=12.5
	Default value	0.0
FXP	Purpose	Specify the view rotation fixed point in the object coordinate system.
	Syntax	FXP= <i>X,Y,Z</i>
	Input Parameters	FLOAT <i>x</i> The x coordinate FLOAT <i>y</i> The y coordinate FLOAT <i>z</i> The z coordinate
	Response	none
	Example	⇒FXP=10.5,20,15.0
	Default value	0.0,0.0,0.0
FXT	Purpose	Specify the second fixed point in the object coordinate system of the viewing section rotation used only with MOD=FIXED_LINE.
	Syntax	FXT= <i>X,Y,Z</i>
	Input Parameters	FLOAT <i>x</i> The x coordinate FLOAT <i>y</i> The y coordinate FLOAT <i>z</i> The z coordinate
	Response	none
	Example	⇒FXT=30,-20.2,15.0
	Default value	0.0,0.0,0.0

MAP	Purpose	Defines a colour or grey value mapping. Grey values outside the mapped region are clamped to the minimum and maximum output colour component or grey values. There may be 1,2,3 or 4 mapping specifications given; with a single mapping specification meaning a grey value mapping, two specifications a grey with alpha mapping, three an RGB mapping and four an RGB α mapping.
	Syntax	MAP= <i>mspec</i> [, <i>mspec</i> [, <i>mspec</i> [, <i>mspec</i>]]] where <i>mspec</i> = <i>t</i> , <i>il</i> , <i>iu</i> , <i>ol</i> , <i>ou</i> , <i>p0</i> , <i>p1</i>
	Input Parameters	<i>t</i> \in <i>IDENTITY</i> <i>LINEAR</i> <i>GAMMA</i> <i>SIGMOID</i> The mapping function FLOAT <i>il</i> Input lower grey value FLOAT <i>iu</i> Input upper grey value FLOAT <i>ol</i> Output lower colour component or grey value FLOAT <i>ou</i> Output upper colour component or grey value FLOAT <i>p0</i> Gamma (γ) or first sigmoid parameter (μ) FLOAT <i>p1</i> Second sigmoid parameter (σ)
	Response	none
	Example	\Rightarrow MAP=LINEAR,0,4095,0,255 Input grey values in the range 0-4095 are mapped to output values with the range 0-255
	Default value	IDENTITY

MOD	Purpose	Specify projection mode.
	Syntax	MOD= <i>mode</i>
	Input Parameters	<i>mode</i> ∈ STATUE UP_IS_UP FIXED_LINE ZERO_ZETA ZETA
		The projection mode
	Response	none
	Example	⇒MOD=FIXED_LINE
PIT	Default value	UP_IS_UP
	Purpose	Specify the pitch angle of the sectioning rotation
	Syntax	PIT= <i>angle</i>
	Input Parameters	FLOAT <i>angle</i> rotation angle in degrees
	Response	none
	Example	⇒PIT=180.0
PAB	Range	0.0-180.0
	Default value	0.0
	Purpose	Specify a 3D query point absolute in the object coordinate system.
	Syntax	PAB= <i>X,Y,Z</i>
	Input Parameters	FLOAT <i>x</i> The x coordinate FLOAT <i>y</i> The y coordinate FLOAT <i>z</i> The z coordinate
	Notes	If both PRL and PAB are specified then PRL has priority of feature queries
PAB	Response	none
	Example	⇒PAB=200,-50,30

PRL	Purpose	Specify a 2D query point relative to tile or display coordinate system.
	Syntax	PRL = <i>T</i> , <i>X</i> , <i>Y</i>
	Input Parameters	RANGE <i>T</i> The <i>T</i> tile number RANGE <i>X</i> The <i>x</i> coordinate RANGE <i>Y</i> The <i>y</i> coordinate
	Notes	PRL either specified a coordinate in a given tile, or if the tile number <i>T</i> =-1 then in the display coordinates
	Response	none
	Range	<i>T</i> =-1 .. maxtile <i>X</i> , <i>Y</i> are limited to the tile or section size
DPT	Example	⇒ PRL =2,20,1
	Default value	none
	Purpose	Specify the rendering depth
	Syntax	DPT = <i>depth</i>
	Input Parameters	FLOAT <i>depth</i> rendering depth for projection rendering modes, depth zero implies no limit
	Response	none
RMD	Example	⇒ DPT =10
	Default value	0
	Purpose	Specify the rendering mode
	Syntax	RMD = <i>mode</i>
RMD	Input Parameters	<i>mode</i> ∈ SECT PRJN PRJD PRJV The rendering mode
	Response	none
	Example	⇒ RMD =PRJV
	Default value	SECT

ROL	Purpose	Specify the roll angle of the sectioning rotation
	Syntax	ROL= <i>angle</i>
	Input Parameters	FLOAT <i>angle</i> rotation angle in degrees
	Response	none
	Example	⇒ROL=20.0
	Range	0.0-360.0
	Default value	0.0
SCL	Purpose	Specify the scale used in the sectioning transformation.
	Syntax	SCL= <i>scale</i>
	Input Parameters	FLOAT <i>scale</i> the scale. A values over one is up, while lower than one is down-scaling.
	Response	none
	Example	⇒SCL=2.5
	Range	positive
	Default value	1.0
UPV	Purpose	Specify the up vector for the UP_IS_UP mode.
	Syntax	UPV=X,Y,Z
	Input Parameters	FLOAT <i>x</i>
		The x component
		FLOAT <i>y</i>
		The y component
	Response	FLOAT <i>z</i>
		The z component
		none
	Example	⇒UPV=0,-1,2
	Default value	0.0,0.0,-1.0
YAW	Purpose	Specify the yaw angle of the sectioning rotation.
	Syntax	YAW= <i>angle</i>
	Input Parameters	FLOAT <i>angle</i> rotation angle in degrees
	Response	none
	Example	⇒YAW=2.0
	Range	0.0-360.0
	Default value	0.0

PTL	Purpose	Retrieve a tile as a PNG image. This command is equivalent to the JTL command returning JPEG tiles.
	Syntax	<i>PTL=res,tile</i>
	Input Parameters	INT <i>res</i> resolution INT <i>tile</i> tile number
	Response	Requested tile <i>tile</i> on the resolution <i>res</i>
	Example	\Rightarrow PTL=1,2
	Default value	0,0
CVT	Purpose	Request an image to be returned as a composed image. CVT accept now also PNG format requests.
	Syntax	<i>CVT=format</i>
	Input Parameters	PNG JPEG <i>format</i> output format
	Response	Requested image
	Example	\Rightarrow CVT=png
	Default value	jpeg

SEL	Purpose	Specify a component of a compound object to be displayed and its colour. The component can be specified either by it's index or by a image processing expression which can combine the components through the use of image processing operators. Where a Woolz object is not a compound object, the SEL command will regard it as a compound object with a single component of index zero. Image processing expressions are explained in 2.4 . If the command is not present then the first object is returned. Multiple SEL will stack up the section in the request order.
	Syntax	SEL=E,R,G,B,A SEL=E,R,G,B SEL=E,A SEL=E
	Input Parameters	UINT E compound object index or image processing expression UINT R red value UINT G green value UINT B blue value UINT A alpha value
	Response	none
	Example	\Rightarrow SEL=0,255,0,255,127
	Default value	0,0,0,0,0

2.3 Extended object reference

IIP-server	Purpose	Identify if WlzIIPsrv is running.
	Syntax	IIP-server
	Response	IIP-server:255.552255
	Example	\Rightarrow OBJ=IIP-server \Leftarrow IIP-server:255.552255
	Note	This object should be used to verify if the IIP server has Woolz sectioning capabilities

Max-size	Purpose	Return the size of the section. For a Woolz object, the size is dependent on the sectioning parameters
	Syntax	Max-size
	Response	Max-size:width height INT <i>width</i> The width in pixels of the section at the current scale INT <i>height</i> The height in pixels of the section at the current scale
	Example	⇒OBJ=Max-size ⇐Max-size:512 1024
	Note	The size is dependent on the viewing plane defining parameters
Tile-size	Purpose	Return the size of a tile.
	Syntax	Tile-size
	Response	Tile-size:width height INT <i>width</i> The width in pixels of the tile INT <i>height</i> The height in pixels of the tile
	Example	⇒OBJ=Tile-size ⇐Tile-size:64 64
	Note	The size is constant throughout the life of the server (see also section 6.4).

Wlz-true-voxel-size	Purpose	Returns the voxel size of the object.
	Syntax	<code>Wlz-true-voxel-size</code>
	Response	<code>Wlz-true-voxel-size:X Y Z</code> <code>FLOAT X</code> The x size <code>FLOAT Y</code> The y size <code>FLOAT Z</code> The z size
	Example	<code>⇒OBJ=Wlz-true-voxel-size</code> <code>⇐Wlz-true-voxel-size:2 1 2.2</code>
	Note	The voxel size is object specific, but view independent.
Wlz-volume	Purpose	Returns the volume of the object.
	Syntax	<code>Wlz-volume</code>
	Response	<code>Wlz-volume:volume</code> <code>INT volume</code> The volume of the Woolz object
	Example	<code>⇒OBJ=Wlz-volume</code> <code>⇐Wlz-volume:748</code>
	Note	The volume is object specific, but view independent.

Wlz-n-components	Purpose	Returns the number of components of a compound object.
	Syntax	<code>Wlz-n-components</code>
	Response	<code>Wlz-n-components:n</code> <code>INT n</code> The number of components of the Woolz object
	Example	<code>⇒OBJ=Wlz-n-components</code> <code>⇐Wlz-n-components:3</code>
	Note	The number of components will be 1 if the object is not a compound object or is compound and only has a single component; in other cases the number of components will be greater for a valid object.
Wlz-distance-range	Purpose	Returns the range of the sectioning plane distance.
	Syntax	<code>Wlz-distance-range</code>
	Response	<code>Wlz-distance-range:min max</code> <code>FLOAT min</code> The minimum distance <code>FLOAT max</code> The maximum distance
	Example	<code>⇒OBJ=Wlz-distance-range</code> <code>⇐Wlz-distance-range:-20 80</code>
	Note	The distance range is view-dependent.
Wlz-sectioning-angles	Purpose	Returns in degrees pitch, yaw and roll angles of the rotation of the sectioning plane.
	Syntax	<code>Wlz-sectioning-angles</code>
	Response	<code>Wlz-sectioning-angles:pitch yaw roll</code> <code>FLOAT pitch</code> The pitch angle of viewing plane rotation <code>FLOAT yaw</code> The yaw angle of viewing plane rotation <code>FLOAT roll</code> The roll angle of viewing plane rotation
	Example	<code>⇒OBJ=Wlz-sectioning-angles</code> <code>⇐Wlz-sectioning-angles:0 40 30</code>

Wlz-3d-bounding-box	Purpose	Returns The first and last plane, line and column number of the object.
	Syntax	Wlz-3d-bounding-box
	Response	Wlz-3d-bounding-box:plane1 lastpl line1 lastln column1 lastcl FLOAT plane1 The first plane number of the object FLOAT lastpl The last plane number of the object FLOAT line1 The first line number of the object FLOAT lastln The last line number of the object FLOAT column1 The first column number of the object FLOAT lastcl The last column number of the object
	Example	⇒OBJ=Wlz-3d-bounding-box ⇐Wlz-3d-bounding-box:0 10 -15 15 30 90

Wlz- transformed- 3d- bounding- box	Purpose	Returns The first and last plane, line and column number of the object after a section transform.
	Syntax	Wlz-transformed-3d-bounding-box
	Response	Wlz-transformed-3d-bounding-box:plane1 <i>lastpl line1 lastln colum1 lastcl</i> FLOAT plane1 The first section plane number FLOAT lastpl The last section plane number FLOAT line1 The first line number of the section FLOAT lastln The last line number of the section FLOAT colum1 The first column number of the section FLOAT lastcl The last column number of the section
	Example	⇒OBJ=Wlz-transformed-3d-bounding-box ⇐Wlz-transformed-3d-bounding-box:10 100 -23 12 308 490
Wlz- coordinate- 3D	Purpose	Returns the 3D object coordinates defined in 2D by the PRL command.
	Syntax	Wlz-coordinate-3D
	Response	Wlz-coordinate-3D:X Y Z FLOAT x The x coordinate FLOAT y The y coordinate FLOAT z The z coordinate
	Example	⇒OBJ=Wlz-coordinate-3D ⇐Wlz-coordinate-3D:20 30 10

Wlz-grey-stats	Purpose	Computes and returns simple statistics of the grey values of a Woolz object.
	Syntax	<code>Wlz-grey-stats</code>
	Response	<p><code>Wlz-grey-stats:n t gl gu sum ss mean stddev</code> or <code>Wlz-grey-stats:NULL</code> <code>UINT n</code> Number of image values in the Woolz object. <code>WlzGreyType t</code> The Woolz image value type, with $t \in \text{WLZ_GREY_INT} \mid \text{WLZ_GREY_SHORT} \mid \text{WLZ_GREY_UBYTE} \mid \text{WLZ_GREY_FLOAT} \mid \text{WLZ_GREY_DOUBLE} \mid \text{WLZ_GREY_RGBA}$ <code>FLOAT gl</code> Minimum image value. <code>FLOAT gu</code> Maximum image value. <code>FLOAT sum</code> Sum of image values. <code>FLOAT ss</code> Sum of squares of image values. <code>FLOAT mean</code> Mean image value. <code>FLOAT stddev</code> Standard deviation of image values.</p>
	Example	\Rightarrow OBJ=Wlz-grey-value \Leftarrow Wlz-grey-stats: 489600 UBYTE 0 255 8.5857e+06 1.42684e+09 17.5362 51.0567
	Note	<p>For RGBα Woolz objects the computed values will be for the modulus of the image values.</p> <p>When the object does not have image values <code>NULL</code> will be returned.</p>

Wlz-grey-value	Purpose	Returns in grey or RGB value of a point specified either the PRL or the PAB commands.
	Syntax	Wlz-grey-value
	Response	Wlz-grey-value:grey or Wlz-grey-value:R G B INT <i>grey</i> The grey value of the pixel INT <i>R</i> The red channel of the pixel INT <i>G</i> The green channel of the pixel INT <i>B</i> The blue channel of the pixel
	Example	⇒OBJ=Wlz-grey-value ⇐Wlz-grey-value:0 255 10
	Note	The returned values are in the range of 0 to 255.
Wlz-transformed-coordinate-3d	Purpose	Returns the the display coordinates and displacement from the sectioning plane of a 3D point defined by PAB
	Syntax	Wlz-transformed-coordinate-3d
	Response	Wlz-transformed-coordinate-3d:X Y D INT <i>X</i> The x coordinate in display frame INT <i>Y</i> The y coordinate in display frame FLOAT <i>D</i> The the signed distance from the sectioning plane
	Example	⇒OBJ=Wlz-transformed-coordinate-3d ⇐Wlz-transformed-coordinate-3d:30 40 60.0002

Wlz-foreground-objects	Purpose	The components of the compound object that 2D/3D query point is a foreground
	Syntax	Wlz-foreground-objects
	Response	Wlz-transformed-coordinate-3d: O_1 O_2 \dots O_n INT O_1 The compound object index that a point is a foreground pixel
	Example	\Rightarrow OBJ=Wlz-foreground-objects \Leftarrow Wlz-foreground-objects: 0 2 5 10

Object queries **Author**, **Copyright**, **Create-dtm**, **Subject** and **App-name** return *author/ copyright/ create-dtm/ subject/ app-name unknown* since they are not defined for a Woolz object.

2.4 Image Processing Expressions

Image processing expressions may be built using the components of Woolz compound objects and basic image processing operators. The operators supported are: background, difference, dilation, domain, erosion, fill, intersection, occupancy, setvalue, threshold, transfer and union. Image processing expressions are built from compound object indices and the image processing operators. Two of the image processing operators (occupancy and union) may given a list indices in their expressions. The occupancy operator may also be given no indices which implies all components of the compound object. The image processing expressions can be written using the syntax defined in table 3. This allows expressions such as:

`erosion(1,2)` erodes the domain of the object with index one in the compound object by two voxels.

`diff(dilation(1,2),3)` dilates the domain of the object with index one in the compound object by two voxels and then computes the difference between that and the object with index 3.

`domain(threshold(0,ge,200))` creates a domain from the object with index 0 where all voxels have values greater than or equal to 200. Without the domain operator threshold would give a object with image values.

The image processing operators available and their parameters are described in table 4.

<i>exp list</i>	$:=$	$(exp \mid idx \text{ list}) (, exp \text{ list})$
<i>exp</i>	$:=$	$idx \mid$ $background(exp, val) \mid$ $diff(exp, exp) \mid$ $dilation(exp, uint) \mid$ $domain(exp) \mid$ $erosion(exp, uint) \mid$ $fill(exp) \mid$ $intersect(exp \text{ list}, exp \text{ list}) \mid$ $occupancy(index \text{ list}?) \mid$ $setvalue(exp, val) \mid$ $threshold(exp, val, cmp) \mid$ $transfer(exp, exp) \mid$ $union(exp \text{ list} \mid idx \text{ list}, exp \text{ list} \mid idx \text{ list})$
<i>idx list</i>	$:=$	$(uint \mid idx \text{ range} \mid idx \text{ list}, idx \text{ list})$
<i>idx range</i>	$:=$	$uint - uint$
<i>idx</i>	$:=$	$[0-9]^+$
<i>uint</i>	$:=$	$[1-9][0-9]^*$
<i>val</i>	$:=$	$[-+]^?[0-9]^* \cdot ? [0-9]^+ ([eE] [-+]^?[0-9]^+)^?$
<i>cmp</i>	$:=$	$(lt) \mid (le) \mid (eq) \mid (ge) \mid (gt)$

Table 3: Syntax for image processing expressions.

Operator	Description
<code>background(<i>exp,value</i>)</code>	Sets the background value (value outside the expression object's domain) to the given value.
<code>diff(<i>exp,exp</i>)</code>	The difference between the two given domains.
<code>dilation(<i>exp,radius</i>)</code>	The dilation of the domain by <i>radius</i> voxels.
<code>domain(<i>exp</i>)</code>	The domain of the expression's object.
<code>erosion(<i>exp,radius</i>)</code>	The erosion of the domain by <i>radius</i> voxels.
<code>fill(<i>exp</i>)</code>	Fills all holes not connected to the outside in the given expression's domain.
<code>intersect(<i>exp list</i>)</code>	The intersection of the domains in the given lists.
<code>occupancy(<i>idx list?</i>)</code>	The domain occupancy in the optional list.
<code>setvalue(<i>exp,value</i>)</code>	Creates an object with the domain of the given expression and all values set to the given value.
<code>threshold(<i>exp,value,comparison</i>)</code>	Creates an object where the image values satisfy the given <i>value</i> and <i>comparison</i> . Here the value is floating point and valid comparisons are lt (less than), le (less than or equal), eq (equal), ge (greater than or equal) and gt (greater than).
<code>transfer(<i>exp,exp</i>)</code>	Creates a new object with the domain of the first expression's object and the values of the second. Values outside the domain of the second but within the first are set to the second's background value.
<code>union(<i>exp list idx list, exp list idx list</i>)</code>	The union of the domains in the given lists.

Table 4: Descriptions of image processing operators

3 HTML query examples

Tile request

```
http://localhost/cgi-bin/iipsrv.fcgi?  
  WLZ=/home/bill/pics/lobster3d.wlz&  
  QLT=50&JTL=1,0
```

Returns a jpeg tile 0 with a quality factor of 50%. The section default parameters, mode UP_IS_UP and zero distance and viewing angles are used.

Sectioning plane distance

```
http://localhost/cgi-bin/iipsrv.fcgi?  
  WLZ=/home/bill/pics/lobster3d.wlz&DST=8&QLT=50&PTL=1,0
```

As above, but the sectioning plane distance is 8 and the returned tile has been compressed with png not jpeg format.

Sectioning mode, plane distance and angle

```
http://localhost/cgi-bin/iipsrv.fcgi?DST=40&YAW=61.5&PIT=3&ROL=0&MOD=ZETA&  
  WLZ=/home/zsolth/small.wlz&QLT=50&CVT=jpeg
```

Returns the whole section with the distance 40, yaw angle 61.5, pitch 3, roll 0 degrees, for a plane at distance 40

Distance range query

```
http://localhost/cgi-bin/iipsrv.fcgi?YAW=61&PIT=3&ROL=0&MOD=ZETA&  
  WLZ=/home/zsolth/small.wlz&OBJ=Wlz-distance-range
```

results in

```
Wlz-distance-range:0 171
```

3D object coordinate query

```
http://localhost/cgi-bin/l.fcgi?WLZ=/home/zsolth/fif/wlz/ts14.wlz&  
  DST=200&YAW=10&PIT=3&PRL=0,30,40&OBJ=Wlz-coordinate-3d
```

results in

```
Wlz-coordinate-3d:418.43 53.5708 15.9916
```

The (30,40) display coordinate in sectioning plane DST=200 has 3D object coordinates (418.43,53.5708,15.9916).

3D coordinate projection to a section

`http://localhost//fcgi-bin/1.fcgi?WLZ=/home/zsolth/fif/wlz/ts14.wlz&
DST=130&YAW=10&PIT=3&PAB=418.43,53.5708,15.9916&OBJ=Wlz-tranformed-coordinate-`
results in

`Wlz-transformed-coordinate-3d:30 40 70.0002`

The (418.43,53.5708,15.9916) object coordinate in sectioning plane DST=130 has display coordinates (30,40) and has a distance 70.0002 from the plane.

Sectioning, selection and image processing operations

`http://localhost/fcgi-bin/iipsrv.fcgi?DST=40&YAW=61.5&PIT=3&ROL=0&MOD=ZETA&
WLZ=/home/bill/compound.wlz&
SEL=(dilation(union(1-3)),255,0,0,255)&
QLT=50&CVT=jpeg`

The section is selected as described above and a domain is created that is the dilated union of components 1, 2 and 3 of the compound object.

4 WlzIIPsrv coding

4.1 Architecture

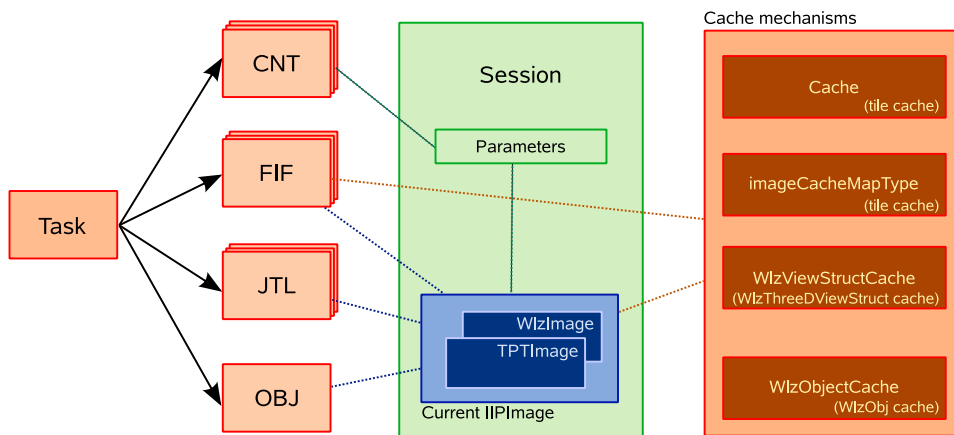


Figure 2: wlziipsrv version 0.9.7 architecture

Figure 2 depicts the architecture of the WlzIIPsrv this being inherited from the original IIPsrv and explaining both. Blocks in the figure represent either C++ classes or structures.

The **Task** class is the base of all command classes that serve independently each IIP command. Its static `factory(string type)` method creates an instance of the appropriate class that serves the `type` IIP command.

Figure 2 shows some of the classes implementing the IIP commands, all derived from the **Task**. Four command types can be identified:

Parameter setting commands such **CNT** and **ROL** pass a parameter to the server. The parameter value is stored by the command in the **Session** structure.

Object defining commands such **FIF** and **WLZ** receive the file name of an object and creates the necessary structures to handle an object. In WlzIIPsrv, either **TPTImage** and **WlzImage** classes store Tiled Pyramidal TIFFs and Woolz objects.

Image providers such **JTL**, **PTL** and **CVT** generate a tile or a full image corresponding to the previously set parameters, stored in the **Session** structure.

Parameter enquiry with **OBJ** command returns a parameter computed or previously specified.

The **Session** structure stores all session related parameters, including the current image object.

To reduce the computation overhead, three types of caches are used:

- for tiles (**RawTile**),
- for flat images (**IIPImage**)
- for Woolz objects and view structures (**WlzObj**).

The interactions between classes, figure 2, are multiple:

- task creates the requested command class,
- commands set session related parameters and the current object
- the **OBJ** command provides the current object parameters,

- before loading from disc, when an operation requires access object data the appropriate cache structure is checked first. Also, when a tile is requested, if its parameters result in a cache hit then the cached tile is returned. The Woolz view structures are cached and looked before a section is generated.

4.2 IIPsrv and WlzIIPsrv beside the IIP specification

IIP [?, p.25] defines a set of mandatory and optional commands that an IIP server should implement. The IIPsrv implements a subset of these commands, while also adding extra commands. WlzIIPsrv further extends the command set, while some of the original IIP commands are incompatible with Woolz objects.

The list of commands from Table 5 compares command sets of the IIP specification, the IIPsrv and the WlzIIPsrv with supported (S), unsupported (N), partially implemented (P) and commands in an unknown, undocumented or nonfunctional state (U).

5 Woolz IIP Proxy

The Woolz IIP Proxy filters and forwards IIP requests to one or more WlzIIP servers. The requests conform to the FCGI protocol. Though it was designed to work for IIP and Woolz requests, it is generic and can be applied to any FCGI request. Therefore, it is possible to chain multiple proxies.

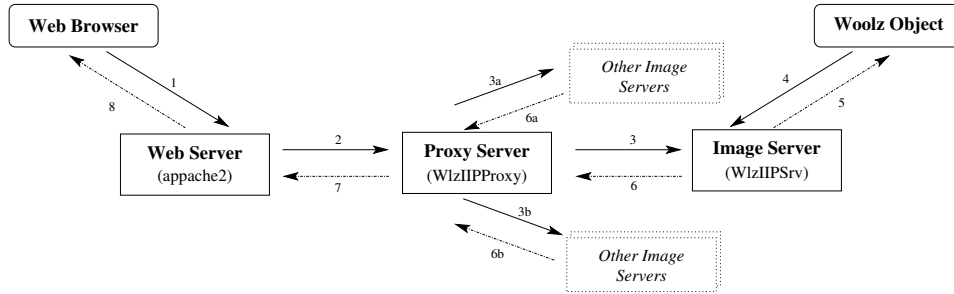


Figure 3: Architecture of Woolz IIP Server using a proxy server. The proxy forwards the user web requests served first by apache to the individual IIP Servers that have direct access to the Woolz Object. The numbered lines show the ordering of the requests (continuous line) and the replies (dotted lines).

Command	IIP spec	IIPSrv	WlzIIPSrv
AFN	S	N	N
CIN	S	N	N
CNT	S	S	S
CTW	S	N	N
CVT	S	S	S
FIF	S	S	S
FTR	S	N	N
HEI	S	U	U
ICC	S	S	U
JTL	S	S	P
OBJ	S	S	S
QLT	S	S	S
RAR	S	N	N
RFM	S	N	N
RGN	S	S	S
ROI	S	S	S
RST	S	N	N
SDS	S	S	N
TIL	S	S	P
WID	S	U	U
JTLS	N	S	N
SHD	N	S	U
DST	N	N	S
FXP	N	N	S
FXT	N	N	S
MAP	N	N	S
MOD	N	N	S
PAB	N	N	S
PIT	N	N	S
PRL	N	N	S
PTL	N	N	S
DPT	N	N	S
RMD	N	N	S
ROL	N	N	S
SCL	N	N	S
SEL	N	N	S
UPV	N	N	S
WLZ	N	N	S
YAW	N	N	S
Affine-transform	S	N	N
App-name	S	U	P
Aspect-ratio	S	N	N
Author	S	U	N
Basic-info	S	S	S
Colorspace	S	S	U
Color-twist	S	N	N
Comment	S	U	N
Comp-group	S	N	N
Contrast-adjust	S	N	N
Copyright	S	U	S
Create-dtm	S	U	N
Edit-time	S	U	N
File-class-id	S	N	N
Filtering-value	S	N	N
ICC-profile	S	N	N
IIP-opt-comm	S	S	S
IIP-opt-obj	S	S	S
IIP-server	S	S	S
IIP-socket	S	N	N
IIP	S	P	P
Keywords	S	U	N
Last-author	S	U	N
Last-printed	S	U	N
Last-save-dtm	S	U	N
Max-size	S	S	S
Property	S	N	N
Render-path	S	N	N
Resolution-number	S	S	S
Rev-number	S	U	N
ROI	S	N	N
Security	S	N	N
Stream	S	N	N
Subject	S	U	P
Summary-info	S	U	P
Title	S	U	N
View-info	S	N	N
Horizontal-views	N	S	U
Tile-size	N	S	S
Vertical-views	N	S	U
Wlz-3d-bounding-box	N	N	S
Wlz-coordinate-3D	N	N	S
Wlz-distance-range	N	N	S
Wlz-foreground-objects	N	N	S
Wlz-grey-stats	N	N	S
Wlz-grey-value	N	N	S
Wlz-n-components	N	N	S
Wlz-sectioning-angles	N	N	S
Wlz-transformed-3d-bounding-box	N	N	S
Wlz-transformed-coordinate-3d	N	N	S
Wlz-true-voxel-size	N	N	S
Wlz-volume	N	N	S

Table 5: Command and object request support by IIP protocol iipsrv0.9.7

The WlzIIPProxy is an independent program running on the proxy server. Apache2 server forwards the FCGI request to this sever, on a configurable port. WlzIIPsrv check the html request string (the FCGI_PARAMS packet QUERY_STRING parameter) and if any remote FCGI server definition string is a substring of the request then request is forwarded to this server. If there is no hit then the request is forwarded to the first FCGI remote server. The default setup of the WlzIIPProxy architecture is shown in figure3

5.1 WlzIIPProxy options

Usage:

```
WlzIIPProxy [-p<portnumber>] [-c<conf_filename>] [-l<log_filename>]
            [-v<loglevel>] [-h]
```

where the options are:

- -p Port number. Default value: 123777
- -c Configuration file containing WLZ name to server name mapping. Default value: WlzIIPProxy.conf
- -l Log file name. -v with a value greater than 0 must be used. Default value: /tmp/WlzIIPProxy.log
- -v Log level. Default value: 0
- -h Help, prints usage message.

The log level is 0 to 3 with

- 0 – no log,
- 1 – system startup, shutdown, error messages,
- 2 – as fcgi connection messages,
- 3 – as level 2 and received/sent packet types.

The configuration file has three columns for:

1. search string,
2. server name (or ip),
3. port number.

6 WlziIPsrv installation

6.1 Install FastCGI

Download and install FastCGI [?]. WlziIPsrv was tested with fcgi-2.4.0 only.

6.2 Source code

Obtain the source files from github:

<https://github.com/ma-tech/WlziIPsrv>

The following packages are also required:

<https://github.com/ma-tech/External>

<https://github.com/ma-tech/Woolz>

6.3 Compiling

Having already built the required external libraries including the Woolz libraries; within the WlziIPsrv root directory edit and execute the build script:

```
cp build.sh mybuild.sh
vi mybuild.sh
./mybuild.sh
```

The build script can be edited, but most often this will only need to be done to change the install prefix. The Woolz libraries should be build with external file format support enabled.

For doxygen documentation run `make doc`. The documentation is generated into the `Docs` directory.

Installing implies moving the `src/wlziipsrv.fcgi` into the server's fcgi directory:

```
cp src/wlziipsrv.fcgi /opt/apache/fcgi-bin/
```

Do not forget to set read and execute access modes!

Parameter	Description	Default value
LOGFILE	Location of log file	/tmp/iipsrv.log
LOGLEVEL	Log level (EMERG, FATAL, ALERT, CRIT, ERROR, WARN, NOTICE, INFO, DEBUG)	WARN
FILESYSTEM_PREFIX	Prefix directory for images	”
MAX_WLZOBJ_CACHE_SIZE	Maximum Woolz object cache size in MBs	1024
MAX_WLZOBJ_CACHE_COUNT	Maximum number of Woolz object in cache	100
WLZ_TILE_WIDTH	Tile width in pixels	100
WLZ_TILE_HEIGHT	Tile height in pixels	100

Table 6: WlzIIPsrv extra configuration parameters

6.4 Customisable parameters

In addition to the IIPsrv [?, p.25] configuration parameters, WlzIIPsrv allows with the parameters from Table 6 changing the logging, caches and tile sizes.

The example configuration from appendix A defines a view structure cache with 200 structures, a 1500MB Woolz object cache and 100×100 tile size.

7 WlzIIPProxy installation

- Download and compile / install FastCGI [?].
- Get WlzIIPProxy source code from the CVS repository:

```
cvs checkout -P src/Applications/WlzIIPProxy
```

- Configure and compile WlzIIPProxy: Run from the WlzIIPsrv root directory

```
aclocal; autoheader; automake; autoconf; ./configure; make
```

- Run WlzIIPProxy on the desired port and using the configuration file.

```
WlzIIPProxy -p <portnumber> -c <config_file>
```

- Configure apache2 on the web server:
 1. install mod_fastcgi (you might need to reinstall apache2) **Note:** the mod_fcgi module, provided in SUSE 10.3 can not make remote FCGI request therefore is not compatible with WlzIIPsrv. mod_fastcgi must be used.
 2. add to httpd.conf (/opt/apache/conf/httpd.conf):

```
FastCgiExternalServer <virtual_path_to_fcgi> -host <hostname>:<portnumber>
```

8 Acknowledgement

We gratefully acknowledge the support from NIH under grant #1R01MH070370-01A2.

A fcgi configuration

An example of the apache2 configuration (/opt/apache/conf/httpd.conf) is¹

```
# Create a directory for the iipsrv binary
ScriptAlias /fcgi-bin/ "/opt/apache/fcgi-bin/"

# Set the options on that directory
<Directory "/opt/apache/fcgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    # Set the module handler
    AddHandler fcgid-script .fcgi
</Directory>

# Initialise the FCGI server and set some default values
FastCgiServer /opt/apache/fcgi-bin/wlziipsrv.fcgi \
-initial-env LOGFILE=/opt/apache/logs/wlziipsrv.log \
-initial-env LOGLEVEL=WARN \
```

¹This configuration file is compatible with mod_fcgi. For mod_fastcgi alternative format must be used (see IIPsrv web-page).


```
-initial-env JPEG_QUALITY=50 \  
-initial-env FILESYSTEM_PREFIX=/export/images \  
-initial-env MAX_CVT=3000 \  
-initial-env MAX_WLZOBJ_CACHE_COUNT=100 \  
-initial-env MAX_WLZOBJ_CACHE_SIZE=4000 \  
-initial-env WLZ_TILE_WIDTH="100" \  
-initial-env WLZ_TILE_HEIGHT="100"
```