
EMAP Technical Report

IIP extension for Woolz sectioning

Author: **Zsolt Husz**
Email: Zsolt.Husz@hgu.mrc.ac.uk
Date: **October 20, 2008**
Distribution: **Open distribution is permitted**
Status: **Draft**
Source: **Report/000024-WlzIIPsrv**
Project: **NIH**
Report number: **EMAP/Report/000024**
Keywords: **IIPServer, Woolz, sectioning, WlzIIPsrv**

EMAP is a research programme at the MRC Human Genetics Unit in collaboration with Biomedical Sciences, University of Edinburgh. Contact [Dr Duncan Davidson](#) or [Dr Richard Baldock](#) at MRC Human Genetics Unit, Crewe Road, Edinburgh, EH4 2XU, UK.

Abstract

WlzIIPsrv extends the IIP protocol to provide 2D section views of 3D Woolz objects. This document describes the extensions for both the user of the WlzIIPsrv and for its developers, summarising the changes performed.

Contents

Title Page	
Abstract	i
Contents	ii
1 Introduction	1
2 Extension overview	1
2.1 Woolz object specification	3
2.2 Setting sectioning and query parameters	3
2.3 Extended object reference	6
3 HTML query examples	9
4 WlzIIPsrv coding	10
4.1 Architecture	10
4.2 Altered code	11
4.3 Extra code	12
4.4 IIPsrv and WlzIIPsrv beside the IIP specification	12
5 Woolz IIP Proxy	12
5.1 WlzIIPProxy options	14
6 WlzIIPsrv installation	15
6.1 Install FastCGI	15
6.2 Source code	15
6.3 Compiling	15
6.4 Customisable parameters	16
7 WlzIIPProxy installation	16
A Example: Woolz IIP architecture on edradour	16
B fcgi configuration	18

1 Introduction

The IIP protocol provides quick access to large image databases. However in the Mouse Atlas project 3D objects are represented as Woolz objects not as flattened images. Therefore a service is needed providing specific sections of an object. The WlzIIPsrv server provides this based on a previous IIP server implementation, the IIPsrv. This document assumes understanding of the IIP (I3A, 1997).

For clarity, in the rest of this document IIPsrv will refer to the original server, with code version 0.9.7 (Pillay and Pitzalis, 1997), WlzIIPsrv will refer to the extended server, while IIP will point to the protocol standard of I3A (1997).

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.

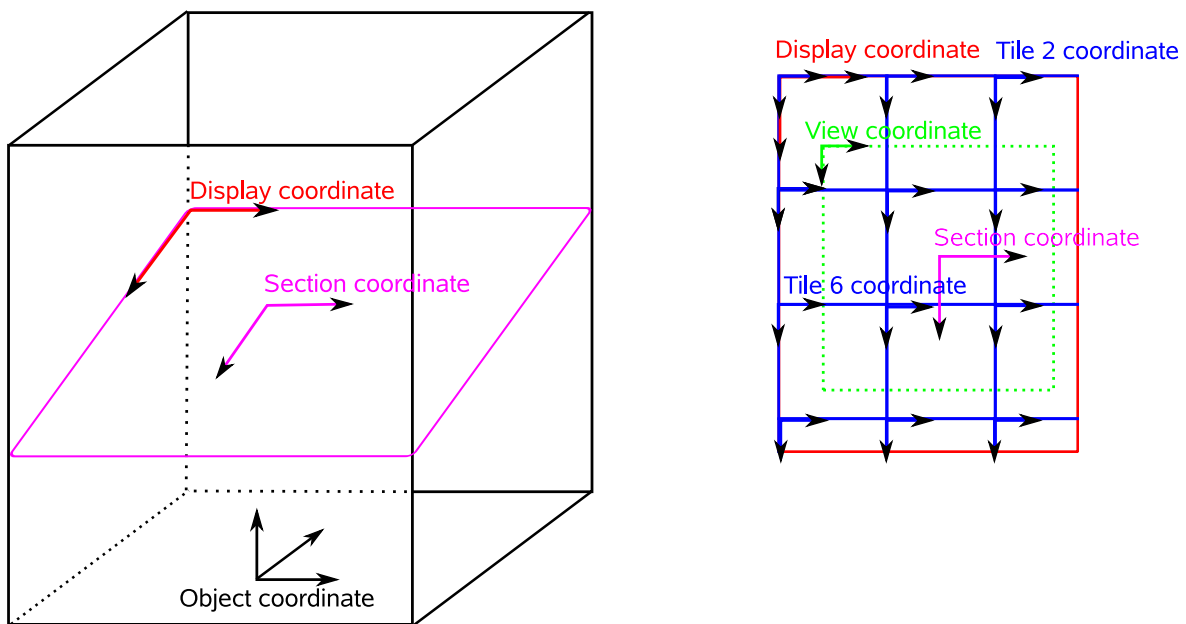


Figure 1: Coordinate systems

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
WLZ	Specify the Woolz object	<code>WLZ=path</code>
DST	Specify the distance of the sectioning plane	<code>DST=dis</code>
FXP	Specify the fixed point of the viewing section rotation	<code>FXP=X,Y,Z</code>
FXT	Specify the second fixed point of the viewing section rotation	<code>FXT=X,Y,Z</code>
MOD	Specify the projection mode	<code>MOD=mode</code>
PIT	Specify the pitch angle of the sectioning rotation	<code>PIT=angle</code>
PAB	Specify the 3D query point absolute in the object coordinate	<code>PAB=X,Y,Z</code>
PRL	Specify the 2D query point relative in tile or display or tile coordinate	<code>PRL=T,X,Y</code>
ROL	Specify the roll angle of the sectioning rotation	<code>ROL=angle</code>
SCL	Specify the scale used in the sectioning transformation	<code>SCL=scale</code>
UPV	Specify the up vector for the UP_IS_UP mode	<code>UPV=X,Y,Z</code>
YAW	Specify the yaw angle of the sectioning rotation	<code>YAW=angle</code>

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-distance-range	The range of the sectioning plane distance
Wlz-sectioning-angles	The pitch, yaw and roll angles of of the sectioning plane
Wlz-3d-bounding-box	The first and last plane, line and column number of the object
Wlz-coordinate-3D	The 3D coordinates defined in 2D by the PRL command
Wlz-grey-value	The grey or RGB value of a point specified either the PRL or the PAB commands

Table 2: Extended object overview

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.
	Syntax	<code>WLZ=path</code>
	Input Parameters	<code>PATH path</code> Full path of the Woolz object.
	Response	WLZ returns nothing when accompanied by other commands
	Example	<code>⇒WLZ=/net/homehost/export/home/bill/pics/lobster3d.wlz</code>
	Notes	Equivalent if FIF (ISA, 1997, p.25)

2.2 Setting sectioning and query parameters

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

DST	Purpose	Specify the distance of the sectioning plane
	Syntax	<code>DST=dis</code>
	Input Parameters	<code>FLOAT dis</code> The distance to the sectioning plane
	Response	none
	Example	<code>⇒DST=12.5</code>
	Default value	0.0
<hr/>		
FXP	Purpose	Specify the view rotation fixed point in the object coordinate system.
	Syntax	<code>FXP=X,Y,Z</code>
	Input Parameters	<code>FLOAT X</code> The x coordinate <code>FLOAT Y</code> The y coordinate <code>FLOAT Z</code> The z coordinate
	Response	none
	Example	<code>⇒FXP=10.5,20,15.0</code>
	Default value	0.0,0.0,0.0
<hr/>		

FXT	Purpose	Specify the second fixed point in the object coordinate system of the viewing section rotation used only with <code>MOD=FIXED_LINE</code> .
	Syntax	<code>FXT=X,Y,Z</code>
	Input Parameters	<code>FLOAT X</code> The x coordinate <code>FLOAT Y</code> The y coordinate <code>FLOAT Z</code> The z coordinate
	Response	none
	Example	<code>⇒FXT=30,-20.2,15.0</code>
	Default value	<code>0.0,0.0,0.0</code>
MOD	Purpose	Specify projection mode.
	Syntax	<code>MOD=mode</code>
	Input Parameters	<code>mode ∈ STATUE UP_IS_UP FIXED_LINE ZERO_ZETA ZETA</code> The projection mode
	Response	none
	Example	<code>⇒MOD=FIXED_LINE</code>
	Default value	<code>UP_IS_UP</code>
PIT	Purpose	Specify the pitch angle of the sectioning rotation
	Syntax	<code>PIT=angle</code>
	Input Parameters	<code>FLOAT angle</code> rotation angle in degrees
	Response	none
	Example	<code>⇒PIT=180.0</code>
	Range	<code>0.0-180.0</code>
Default value	<code>0.0</code>	
PAB	Purpose	Specify a 3D query point absolute in the object coordinate system.
	Syntax	<code>PAB=X,Y,Z</code>
	Input Parameters	<code>FLOAT X</code> The x coordinate <code>FLOAT Y</code> The y coordinate <code>FLOAT Z</code> The z coordinate
	Notes	If both PRL and PAB are specified then PRL has priority of feature queries
	Response	none
	Example	<code>⇒PAB=200,-50,30</code>

PRL	Purpose	Specify a 2D query point relative to tile or display coordinate system.
	Syntax	PRL= <i>T,X,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,y</i> are limited to the tile or section size
	Example Default value	⇒PRL=2,20,1 none
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= <i>X,Y,Z</i>
	Input Parameters	FLOAT <i>X</i> The <i>x</i> component FLOAT <i>Y</i> The <i>y</i> component FLOAT <i>Z</i> The <i>z</i> component
	Response	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	<code>YAW=angle</code>
	Input Parameters	FLOAT <i>angle</i> rotation angle in degrees
	Response	none
	Example	<code>⇒YAW=2.0</code>
	Range	0.0–360.0
	Default value	0.0

2.3 Extended object reference

IIP-server	Purpose	Identify if WlzIIPsrv is running.
	Syntax	<code>IIP-server</code>
	Response	<code>IIP-server:255.552255</code>
	Example	<code>⇒OBJ=IIP-server</code> <code>←IIP-server:255.552255</code>
	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	<code>Max-size</code>
	Response	<code>Max-size:width height</code> 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	<code>⇒OBJ=Max-size</code> <code>←Max-size:512 1024</code>
	Note	The size is dependent on the viewing plane defining parameters
	Tile-size	Purpose
Syntax		<code>Tile-size</code>
Response		<code>Tile-size:width height</code> INT <i>width</i> The width in pixels of the tile INT <i>height</i> The height in pixels of the tile
Example		<code>⇒OBJ=Tile-size</code> <code>←Tile-size:64 64</code>
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> FLOAT <i>X</i> The x size FLOAT <i>Y</i> The y size FLOAT <i>Z</i> 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> INT <i>volume</i> 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-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> FLOAT <i>min</i> The minimum distance FLOAT <i>max</i> 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	<p><code>Wlz-sectioning-angles:pitch yaw roll</code></p> <p><code>FLOAT pitch</code> The pitch angle of viewing plane rotation</p> <p><code>FLOAT yaw</code> The yaw angle of viewing plane rotation</p> <p><code>FLOAT roll</code> The roll angle of viewing plane rotation</p>
	Example	<p><code>⇒OBJ=Wlz-sectioning-angles</code></p> <p><code>←Wlz-sectioning-angles:0 40 30</code></p>
Wlz-3d-bounding-box	Purpose	Returns The first and last plane, line and column number of the object.
	Syntax	<code>Wlz-3d-bounding-box</code>
	Response	<p><code>Wlz-3d-bounding-box:plane1 lastpl line1 lastln</code></p> <p><code>column1 lastcl</code></p> <p><code>FLOAT plane1</code> The first plane number of the object</p> <p><code>FLOAT lastpl</code> The last plane number of the object</p> <p><code>FLOAT line1</code> The first line number of the object</p> <p><code>FLOAT lastln</code> The last line number of the object</p> <p><code>FLOAT column1</code> The first column number of the object</p> <p><code>FLOAT lastcl</code> The last column number of the object</p>
	Example	<p><code>⇒OBJ=Wlz-3d-bounding-box</code></p> <p><code>←Wlz-3d-bounding-box:0 10 -15 15 30 90</code></p>
Wlz-coordinate-3D	Purpose	Returns the 3D object coordinates defined in 2D by the PRL command.
	Syntax	<code>Wlz-coordinate-3D</code>
	Response	<p><code>Wlz-coordinate-3D:X Y Z</code></p> <p><code>FLOAT X</code> The x coordinate</p> <p><code>FLOAT Y</code> The y coordinate</p> <p><code>FLOAT Z</code> The z coordinate</p>
	Example	<p><code>⇒OBJ=Wlz-coordinate-3D</code></p> <p><code>←Wlz-coordinate-3D:20 30 10</code></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 grey The grey value of the pixel INT R The red channel of the pixel INT G The green channel of the pixel INT B 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.

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.

3 HTML query examples

Tile request

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

Returns a jpeg tile 0 with the a quality factor 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=/net/homehost/export/home/bill/pics/lobster3d.wlz&DST=8&QLT=50&JTL=1,0
```

As above, but the sectioning plane distance is 8.

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
```

4 WlzIIPsrv coding

4.1 Architecture

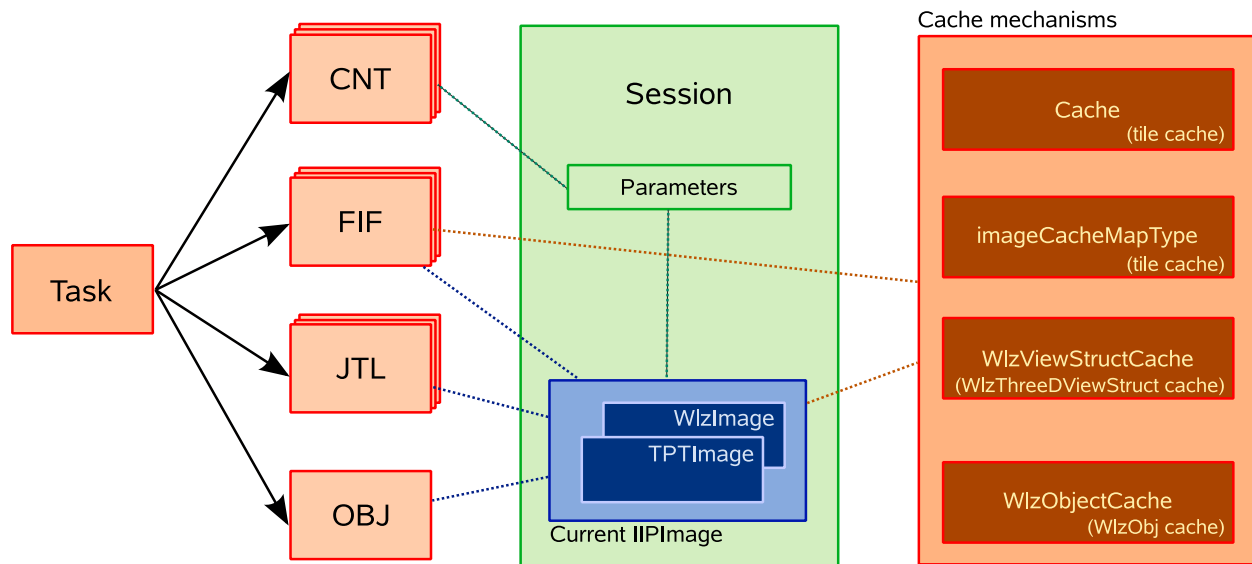


Figure 2: iipsrv version 0.9.7 architecture

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** and **CVT** generate an tile or a full image corresponding to the previously set parameters, stored in the **Session** structure.

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

The **Session** structure stores all session related parameters, including the current image object. To reduce the computation overhead, four types of caches are used:

- for tiles (`RawTile`),
- for flat images (`IIPImage`)
- for Woolz objects (`WlzObj`),
- and for Woolz view structures (encapsulated in `WlzThreeDViewStruct`).

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 it parameters result in a cache hit then the cached tile is returned. The a Woolz view structures are cached and looked before a section is generated.

Classes have been added to `IIPServer` and some of the original code had to be edited to fit them. However, these changes were kept to the minimum. The involved new and changed source code files as summarised below, however for details the doxygen documentation may be consulted.

4.2 Altered code

src/Makefile.am: added `WLZ.cc` and `WlzImage.cc` sources,

src/IIPImage.h: added `getHash()` virtual method,

src/IIPResponse.cc, src/IIPResponse.h: increased error buffer in `setError(const string& , const string)`; added `addResponse` with two and three double parameters,

src/Main.cc: added includes and `viewParameters` structure,

src/Task.h, src/Task.cc: added includes; `*viewParameters` for the current setting; Woolz related command handlers ,

src/OBJ.cc: added Woolz related handlers for parameter query commands; `checkifWoolz()` to check if a Woolz object

src/TileManage.cc: changed `crop(...)`: only if initial title has different size; `getImagePath()` replaced with `getHash()`,

/src/Environment.h added `getMaxViewStructCacheSize()`, `getMaxObjectCacheSize()`, `getWlzTileWidth()` and `getWlzTileHeight()` functions,

/Makefile.am: changed `crop`: crops only if initial title is different size; `getImagePath()` replaced with `getHash()`,

/configure.in: fixed two syntax errors: `fcgi-lib_path` changed to `fcgi_lib_path`; added brackets `FCGI_COMMON_CHECKS()`; `fcgi` files removed from `AC_OUTPUT`,

acinclude.ac: added brackets in `AC_DEFUN([FCGI_COMMON_CHECKS])`; new line added to the end of the file

other files all source files in the `src` directory have CVS strings added, however `fcgi` source files are not altered.

All source files with altered code include "Copyright (C) 2008 Zsolt Husz, Medical research Council, UK".

4.3 Extra code

For the `WlzIIPsrv`, the extra code added to the `IIPsrv` are located in:

WlzImage.cc*, **WlzImage.h***: implementation of sectioned image handling

WLZ.cc: **WLZ** load command implementation

WlzObjectCache.h: Woolz object caching

WlzViewStructCache.h: Woolz view structure caching

ViewParameters.h: section defining and query parameter structure

Main.dox, Doxyfile: doxygen documentation

4.4 IIPsrv and WlzIIPsrv beside the IIP specification

IIP ([I3A, 1997](#), p.25) defines a set of mandatory and optional commands that an IIP server should implement. The `IIPsrv` implements a subset of this commands, while also adds 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 3](#) 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, an 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 the FCGI protocol. Though it was designated 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.

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

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

Table 3: Command and object request support by IIP protocol, iipsrv0.9.7 and WlzIIPSrv. S: supported; N: not supported; P: partially supported; U: unknown, an undocumented or nonfunctional

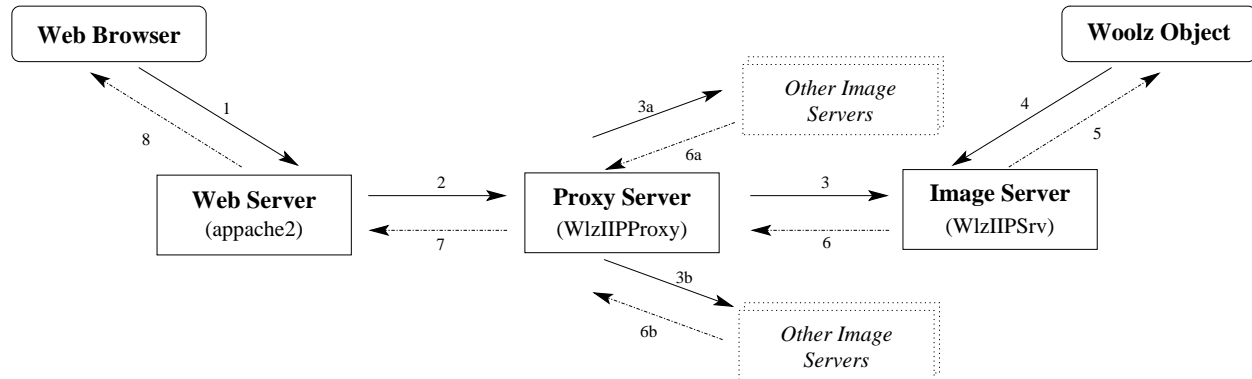


Figure 3: Architecture of Woolz IIP Server using a proxy server. The proxy forwards the user web requests served first by apache to the individual IIP Servers that have direct access to the Woolz Object. The numbered lines show the ordering of the requests (continuous line) and the replies (dotted lines).

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 [GNU \(2003\)](#). WlzIIPsrv was tested with fcgi-2.4.0 only.

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

6.2 Source code

Obtain the source files from the CVS repository.

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

6.3 Compiling

Run from the WlzIIPsrv root directory:

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

`configure` accepts `--with-wlz-incl` and `--with-wlz-incl` parameters that specify the location of either the Woolz installed libraries and includes or the Woolz source Core directory. The default directory is `../../Core` relative to the WlzIIPsrv root directory that normally contains the Woolz source files. The Woolz libraries must be built before compiling WlzIIPsrv.

If the fcgi library can not be automatically located then `--with-fcgi-lib` and `--with-fcgi-incl` specify it. A static link of the fcgi library can be requested with `--enable-static-fcgi`. It requires `--with-fcgi-lib` to set the path of the static library.

Building was tested with GNU automake and aclocal v1.10, autoconf v2.61 and make v3.81.

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

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

```
cp src/iipsrv.fcgi /srv/www/fcgi-bin/
```

Do not forget to set read and execute access modes!

6.4 Customisable parameters

In addition to the IIPsrv (Pillay and Pitzalis, 1997, p.25) configuration parameters, WlzIIPsrv allows with the parameters from Table 4 changing the default cache and tile sizes.

Parameter	Description	Default value
MAX_WLZOBJ_CACHE_SIZE	Maximum Woolz object cache size in MBs	1024
MAX_VIEW_STRUCT_CACHE_SIZE	Maximum Woolz 3D view structure cache element number	1024
WLZ_TILE_WIDTH	Tile width in pixels	100
WLZ_TILE_HEIGHT	Tile height in pixels	100

Table 4: WlzIIPsrv extra configuration parameters

The example configuration from appendix B 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 GNU (2003).
- 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 (/usr/local/apache2/conf/httpd.conf):

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

A Example: Woolz IIP architecture on edradour

Follow the next step on the web server (**WS**), proxy server (**PS**) and WlzIIP server (**IS**) boxes.

1. **WS:** install apache2¹ and mod_fastcgi GNU (2007).
2. **WS:** Add to httpd.conf (/usr/local/apache2/conf/httpd.conf):


```
FastCgiExternalServer /srv/www/fcgi-bin/clusteriip.fcgi -host edradour:8000
```
3. **PS:** build WlzIIPProxy (see section 7)
4. **PS:** *Remark:* The libtiff library has to allow pyramidal tiff structures. If not, then FIF requests of TIFF images are not processed by the server. Add-hoc solution for this: the libtiff library² statically linked by editing Makefile.am, and replacing @TIFF_LIBS@ with /.../libtiff/libtiff/.libs/libtiff.a.
5. **PS:** create configuration file wlziiproxy.cfg (assumed here to be at the same location as the executable) with:

```
***** node-07 8001
Kidney   node-01 8001
lobster  node-02 8001
```

This will use as default server node-07 while kidney requests are forwarded to node-01 and lobster requests are forwarded to node-02.

6. **PS:** start WlzIIPProxy


```
WlzIIPProxy -p 8000 -c wlziiproxy.cfg &
```
7. **IS:** download and compile / install libfcgi GNU (2003),
8. **IS:** build WlzIIPSrv (see section 6) You might need to specify the location of Woolz and FastCGI:

```
./configure \
--with-fcgi-incl=/net/node-00/export/data0/home/zsolth/Proxy/fcgi-2.4.0/include \
--with-fcgi-lib=/net/node-00/export/data0/home/zsolth/Proxy/fcgi-2.4.0/libfcgi \
--with-wlz-incl=/opt/MouseAtlas/include --with-wlz-lib=/opt/MouseAtlas/lib
```

9. **PS:** start WlzIIPSrv


```
WlzIIPSrv --port 8001 &
```
10. **any browser:** test FCGI forwarding


```
http://finlaggan/fcgi-bin/clusteriip.fcgi
```

Here the finlaggan is the WS.
11. **any browser:** test WlzIIP request.


```
http://finlaggan/fcgi-bin/clusteriip.fcgi?DST=4&WLZ=/home/zsolth/Kidney.wlz&QLT=50&JTL=1,1
```

/home/zsolth/Kidney.wlz must be accessible from node-01 with o+r permission.

¹<http://httpd.apache.org/>

²available from <http://www.remotesensing.org/libtiff/> (branch-3-9)

B fcgi configuration

An example of the apache2 configuration (`/etc/apache2/http.conf.local`) is³

```
# Create a directory for the iipsrv binary
ScriptAlias /fcgi-bin/ "/srv/www/fcgi-bin/"

# Set the options on that directory
<Directory "/srv/www/fcgi-bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all

    # Set the module handler
    AddHandler fcgid-script .fcgi
</Directory>

# Set our environment variables for the IIP server
DefaultInitEnv VERBOSITY "5"
DefaultInitEnv LOGFILE "/tmp/iipsrv.log"
DefaultInitEnv MAX_IMAGE_CACHE_SIZE "10"
DefaultInitEnv JPEG_QUALITY "50"
DefaultInitEnv MAX_CVT "3000"
DefaultInitEnv MAX_VIEW_STRUCT_CACHE_SIZE "200"
DefaultInitEnv MAX_WLZOBJ_CACHE_SIZE "1500"
DefaultInitEnv WLZ_TILE_WIDTH "100"
DefaultInitEnv WLZ_TILE_HEIGHT "100"
# Define the idle timeout as unlimited and the number of processes we want
IdleTimeout -1
DefaultMaxClassProcessCount 1
```

References

- GNU (2003). Fastcgi api (libfcgi). <http://http://www.fastcgi.com/dist/fcgi-2.4.0.tar.gz>. version 2.4.0.
- GNU (2007). Mod fast CGI module for Apache2(mod_fastci). http://http://www.fastcgi.com/dist/mod_fastcgi-2.4.6.tar.gz. version 2.4.6.
- I3A (1997). Internet imaging protocol. version 1.0.5.
- Pillay, R. and Pitzalis, D. (1997). IIPsRv. <http://prdownloads.sourceforge.net/iipimage/iipsrv-0.9.7.tar.bz2?download>. version 0.9.7.

³This configuration file is compatible with `mod_fcgi`. For `mod_fastcgi` alternative format must be used (see IIPsRv web-page).