# ImagePro Device Server User's Guide (V1.4)

A.Götz

5 December 2000

This guide describes how to install the ImagePro device server on a Windows 95/98/NT PC, the commands implemented in the TACO ImagePro device server and how to use it from SPEC.

## Contents

# 1 Introduction

ImagePro is an commercial product for doing image acquisition and analysis on Windows 98/NT. It works with a large number of CCD cameras e.g. Princeton, Photonics, Sensicam, Photometrics and supports a rich set of image analysis routines. A useful feature of ImagePro is its macro language for automating acquisition and analysis. The TACO ImagePro device server has been written in order to allow SPEC or any other TACO client to trigger ImagePro commands remotely.

# 2 Getting started

In order to run the device server you need to do the following :

1. install Imagepro (of course !)
2. get the imagepro.zip file (from /segfs/dserver/classes/ccd/imagepro) und unzip it in e.g. c:\DServer
3. copy the oncrpc.dll to the Windows system directory (e.g. c:\WINDOWS) or add the directory C:\DServer\lib to the PATH environment variable.
4. define an imagepro device in the TACO database e.g. ID22/IMAGEPRO/1 and load the ImagePro device server commands in the database (they are in C:\DServer\res\ImagePro.res)
5. Create shortcuts to the portmapper (in C:\DServer\bin\portmapper.exe) and the imagepro device server (in C:\DServer\bin\imagepro.exe). Use Properties on the shortcuts to set the portmapper to come up as an iconized window and imagepro to start with the device server name specified in the TACO database.
6. start ImagePro (by clicking on the "ImagePro" shortcut)
7. start the portmapper (by clicking on the "portmap" shortcut)
8. start the device server with the correct personal name e.g. id22 (by clicking on the "imagepro" shortcut)

Now you can send commands to ImagePro by calling the ImagePro device server. The device server works with version 3 and 4 of ImagePro 32 bit (i.e. 16 bit ImagePro for Windows 3.1 and DOS will not work !).

# 3 Example

Here is an example of calling the device server from SPEC to run a macro in ImagePro :

```
local IP_MACRO_RUN
IP_MACRO_RUN[0] = "my_macro"
IP_MACRO_RUN[1] = ""
esrf_io("id22/imagepro/1","DevIpMacroRun",IP_MACRO_RUN)
```

# 4 SPEC

A set of SPEC macros has been written to setup ImagePro so that it is triggered to do an exposure every "ct" command i.e. at every scan point. This macro set is part of the imagepro zip file and can be found in macro\imagepro.mac. It can be copied to ˜specadm/umacros and loaded using the SPEC command "udo imagepro". Local help can be displayed on this command set using the command "help local imagepro" :

- **NAME Macros for using ImagePro CCD image acquisition software**
- **OVERVIEW This macro set can be used to trigger CCD image acquisitions by ImagePro running on a Windows PC. The macros allow a user to configure the image acquisition basic setup (device, file prefix, exposure time). Triggering is switched on/off using the ipon/ipoff macro macros. When triggering is on image acquisition is triggered at every ct command i.e. during scans. Single shot images can be taking using the ipsnap command.**
- **EXAMPLEipsetup id18/imagepro/1 c:\\mypath myimg Will setup the ImagePro macros to trigger an acquisition from the device id18/imagepro/1 with file prefix "myimg". The file names created will be of the form myimg_n_i where n is the scan number and i is the point number inside a scan.**

  **ipoffSwitch triggering off at ct command**

  **iponSwitch triggering on at every ct command**

  **ipsnap 10 Take a single shot image exposure 10 s and store it in in a new workspace (NOTE: if the file exists already ImagePro will open a dialog box which has to be acknowledged)**

  **ipsave c:\\mypath\\myimageSave the active workspace in a file c:\mypath\myimage.tif in TIF format (this macro can be useful when used in conjunction with ipsnap)**

  **iptake c:\\mypath\\myimage 10 Take a single shot image exposure 10 s and store it in the file c:\mypath\myimage.tif (NOTE: if the file exists already ImagePro will open a dialog box which has to be acknowledged)**

  **ipgetimage Get the current image and store it in the shared memory short array ip_image[](the image can be displayed using the display program called *dis- simply start disand select shared memory segment ip_ image)***

# 5 Commands

## 5.1 DevIpMacroRun

- *description- run a macro on ImagePro either from a file or directly from memory*
- *input- [0] "macro name", [1] "script file" ("" to run from memory)*
- *output- none*

## 5.2 DevIpMacroStop

- *description- stop a macro running*
- *input- [0] "macro name", [1] "mode"*
- *output- none*

### 5.3 DevIpMacroLoad

- *description- load a script file into memory*
- *input- [0] "script file name"*
- *output - none*

### 5.4 DevIpAcqControl

- *description- modify an acquisition parameter e.g. exposure time. This is a general purpose ImagePro command to change various control parameters of a camera. The exact usage of this command is camera dependant. The best way to find out which parameters can be modified and how this is done is by setting the "Record Macro" mode and then using ImagePro to execute the command. The exact format of the IPAcqControl command will be displayed in the macro window. Refer to ImagePro macro manual as well.*
- *NOTE : in order to make the command flexible all parameters are transferred as shorts. This means that if a long parameter has to be transferred then it must be passed as word inverted format in two words e.g. [2]=1000 and [3]=0 to pass 1000 as a long*
- *input- [0] "command", [1] "parameter", [2] "imagepro parameter 1", [3] "imagepro parameter 2"*
- *output- none*

### 5.5 DevIpAcqSnap

- *description- take a single image and store it in a new workspace on the screen. This command returns immediately and changes the state of the camera to DEVRUNNING. The state changes back to DEVON when the acquisition is finished. During this time no acquisitions can be started.*
- *input- "exposure time (in milliseconds)"*
- *output- none*

### 5.6 DevIpWsSaveAs

- *description- save the current workspace in the specified file as TIF format.*
- *input- file name e.g. c:\mypath\myimage*
- *output- none*

### 5.7 DevIpAcqTimed

- *description- start a single or a series of timed acquisitions, this command return immediately and changes the state of the camera to DEVRUNNING. The state changes back to DEVON when the acquisition is finished. During this time no acquisitions can be started.*
- *input- [0] "directory to save images in" ("" to save in memory), [1] "file name prefix" ("" to save in memory, [2] "start number", [3] "no. of frames", [4] "time between frames"*
- *output- none*

## 5.8 DevIpSetVariable

- *description-pass a value to an imagepro internal variable (defined in the .INI file)*
- *input- [0] "variable name", [1] "value"*
- *output- none*

## 5.9 DevIpGetVariable

- *description-get a value from an imagepro internal variable (defined in the .INI file)*
- *input- "variable name"*
- *output- value*

## 5.10 DevIpGetImageSize

- *description-get the current image size (rows and columns))*
- *input- nothing*
- *output- [0] "rows", [1] "columns"*

## 5.11 DevIpGetImage

- *description- returns current image as opaque 16 bit array (note: if reading the image from a Unix box then you have to swap the array before using it using array_op())*
- *input- nothing*
- *output- image as opaque array*

## 5.12 DevIpGetArea

- *description- returns a part (or whole) of the current image as opaque 16 bit array (note: if reading the image from a Unix box then you have to swap the array before using it using array_op())*
- *input- coordinates of area, [0] "left", [1] "right", [2] "top", [3] "bottom"*
- *output- area as opaque array*

## 5.13 DevIpWsChangeDescription

- *description- change the description of any one of the image description tags for the current workspace. Call this command before you save workspace with DevIpWsSaveAs.*
- *input- [0] = tag ("title", "artist", "date", "description", "name", or "range"), [1] = description (e.g. motor settings)*
- *output- none*

# 6 Supported cameras

The following cameras at the ESRF are known to work with ImagePro :

1. Quantix (ID11)
2. Photonics Science XIOS II (ID22 )
3. Photonics Science FDI (ID18)
4. Medoptics (ID11)
5. Frelon (lab)
6. Sensicam (ID10, ID2)
7. Matrox (ID13)

For a full list of cameras for which Imagepro drivers exist refer to the ImagePro website (http://www.mediacy.com). If needed a driver can be written locally. Drivers are quite easy to write and can be done within a day for simple cameras.

# 7 Troubleshooting

Here are some problems you could encounter :

1. *cannot unzip imagepro.zip because WinZip is not installed on on my PC- you can pick up a copy of WinZip at the ESRF from* ***ftp://ftp.esrf.fr/pub/cs/micro/winzip/win9x-nt***
2. *cannot start device server, complains about TACO initialisation- try starting the portmap first*
3. *device of this name not found- make sure you have positioned NETHOST correctly and that there is a device defined in the static database*

# 8 Changes

The following changes have been made in this version :

## 8.1 V1.4

1. added commands DevIpGetVariable and DevIpWsChangeDescription

## 8.2 V1.3

1. added commands DevIpGetImageSize, DevIpGetImage and DevIpGetArea

## 8.3 V1.2

1. added commands DevIpAqcSnap and DevIpWsSaveAs
2. image is displayed after acquisition before being saved (command IpAcqTimed)
3. improved SPEC macros so they don't block, added command ipsave, changed all time arguments from milliseconds to seconds

# 9 Conclusion

The above set of commands allow a user to start any macro or acquisition in ImagePro remotely. It is now possible to read images over the network via the device server e.g. into SPEC. However users are encouraged to use the ImagePro display possibilities directly. More commands (of the few hundred possible) can be implemented on demand if needed.