# AzCamTool User's Manual

Michael Lesser
University of Arizona Imaging Technology Laboratory
Last change 24-Feb-06

## INTRODUCTION

The ***AzCam*** software package has been developed to provide a general purpose scientific image acquisition system suitable for scientific imaging detectors. The intended applications are imaging detector characterization in a laboratory and astronomical imaging and spectroscopic observations. This manual describes ***AzCamTool*****,** the Graphical User Interface (GUI) client program written in National Instruments LabVIEW which runs under the Microsoft Windows and Linux Operating Systems.

This version is for Astronomical Research Cameras, Inc. (ARC) camera controllers with PCI data acquisition interfaces. These controllers have been known as San Diego State University (SDSU) or "Bob Leach" controllers. This manual is applicable to gen1, gen2, and gen3 controllers. ***AzCamTool*** is available for both Windows and Linux operating systems and works identically in both cases. Versions for Macintosh and Solaris are possible but not tested.

***AzCamTool*** has been written by Michael Lesser. Support has been provided by the University of Arizona and the National Science Foundation grant AST-9876630. Please send questions, comments, suggestions, or bug reports to mlesser@as.arizona.edu.

We use the forward slash "/" for filenames, which should be interpreted as reverse slashes "\" under Windows and forward slashes "/" under Linux. In most cases, internal forward slashes are operating system independent and are translated as needed.

The SYSTEM DIRECTORY refers to the directory from which the parameter file is loaded. Most scripts and auxiliary files are expected to be in this directory.

A VI or Virtual Instrument is a LabVIEW construct like a subroutine or function in conventional languages. In some cases VI's can be added to ***AzCamTool*** for customization. In this document we generally do not distinguish between VI's, subroutines, and functions unless required, and refer to all as functions.

A panel (or front panel or window) is a LabVIEW window with which the user interacts.

## INSTALLATION

***AzCamTool*** requires the proper version of the LabVIEW Run-Time Engine from National Instruments to be installed on the host computer. This is available from the ***AzCam*** web site http://www.itl.arizona.edu/azcam/azcam.html or directly from www.ni.com. Download the file and execute it to install the LabVIEW run-time engine if the current version is not already installed on your system.

Copy *setup_AzCamTool.exe* into a temporary directory on the client host machine and execute it. Note this may not be the same machine as the server host. The program will install into <*Program Files*>*\AzCam\AzCamTool.*

## AZCAM COMMUNICATIONS

The ***AzCam*** data acquisition system uses a client-server model for data control and acquisition. ***AzCamServer*** is an executable program on the server-side (the computer with the fiber-optic interface card installed). It is written in the C language. The server communicates directly with the hardware, a client does not.

Client-server communication occurs only over standard TCP/IP sockets in order to not constrain the platform or programming language of clients.

Communication with ***AzCamServer*** is straightforward - a remote or local client opens a socket connection to the server on the appropriate port and then sends string-type socket commands to the server. The server replies to commands with a response string containing at least a status value (“OK” or “ERROR”). The status is usually returned only when the issued command has been completed, although there are exceptions. After the last command is sent, the client should properly close the socket connection by issuing the *CloseConnection* command to the server and then close the client side of the socket connection. ***AzCamServer*** expects to receive and reply with <CR><LF> terminated strings, although other options are possible.

Image data transfer is slightly more complicated because image data is received into a server-side image buffer. Image file creation is optional, and might not be desired when only displaying images or sending image data to a client application.

## AZCAMTOOL OPERATION

***AzCamTool*** is the client-side graphical user interface (GUI) by which user commands are issued to ***AzCamServer*** to operate the controller.

***AzCamTool*** is designed for both detector characterization applications and for observing at telescopes. Versions of ***AzCamTool*** run under Windows (98, 2000, NT, and XP), Linux, or can be ported (with the appropriate license) to any other OS under which National Instrument’s LabVIEW operates (see www.ni.com). The executable version ***AzCamTool*** (but not the LabVIEW source VI’s) may be distributed without requiring a LabVIEW installation and license. Only the LabVIEW Run-Time Engine is required. Note the Windows and Linux versions of ***AzCamTool*** have different executable files, although they are generated from identical source code.

**Very important!** ***AzCamServer*** must be running on the server machine (the one with the PCI fiber optic interface card to the controller) in order to use ***AzCamTool***, except in demo mode. The two programs do not need to be running on the same computer, although they may.

See the AzCam User’s Guide for operating instructions.

## STARTUP

***AzCamTool*** can be run from the Start Menu (Start®Programs®AzCam®AzCamTool), from an icon/desktop shortcut or from Windows Explorer (*/Program Files/AzCam/AzCamTool/AzCamTool.exe* for Windows or *AzCamTool.e* for Linux). ***AzCamServer*** should be running on the server machine before starting ***AzCamTool***.

These are several command line options for ***AzCamTool***. The options must be delimited by spaces. For example, "/Program Files/AzCam/AzCamTool/AzCamTool.exe –parfile /junk/parameterfile.ini".

- ***-startfile*** *FileName* will read *FileName* (include the full path) to find the parameter file to be loaded. This indirect method of specifying a parameter file allows changing the parameter file name without modifying the command line. If a *startfile* is not used then no default parameter file name can be saved.

- ***-parfile*** *ParFileName* will immediately load the parameter file *ParFileName* at startup. This is useful for single configuration systems. No default configuration may be saved in this case, since the specified file is always loaded.

- ***-nofileserver*** specifies that the default FileServer program should not be started when ***AzCamTool*** starts.

If ***–startfile*** and ***–parfile*** are both set, then the ***–parfile*** option will be used.

## PARAMETER FILES

***AzCamTool*** uses parameter files to save useful configuration information. At start up, ***AzCamTool*** reads the parameter file specified by the command line, *startfile*, or dialog box. Parameter files are typically located in the */AzCam/systems* subdirectories. A common practice is to copy an existing system subdirectory to a new name and edit the *DeviceName.ini* file (with any text editor) to generate a new system configuration (or uses **Parameters®Save current parameters as…** as below).

Selecting **Parameters®Save current parameters** from the menu saves current parameters to the current parameter file. Selecting **Parameters®Save current parameters as...** allows the current configuration to be saved with a different filename.

Selecting **Parameters®Set current configuration as default** makes the current parameter file the default for the next time ***AzCamTool*** starts, assuming a *startfile* is used.

From the **Parameters** menu you may also read parameter groups from other parameter file. These are the Detector, Customization, and PlugIn groups. After reading these groups, the new parameters will saved in the current parameter file when the various Save menu options are selected.

## AZCAM DEMO MODE

The demonstration mode of ***AzCamTool*** is useful for practicing data acquisition techniques or for code development without requiring hardware installation (such as a PCI interface card). Both ***AzCamServer*** and ***AzCamTool*** have demo modes. When ***AzCamServer*** is in demo mode, incoming client commands receive positive responses indicating the commands were executed successfully. Synthetic ramp images are generated when exposure commands are received. When ***AzCamTool*** is in demo mode, it does not communicate with ***AzCamServer***, but internally generates positive responses indicating successfully completion of commands. Image data cannot be generated when ***AzCamTool*** runs in demo mode but a ramp image is generated when ***AzCamServer*** runs in demo mode.

To run ***AzCamTool*** in demo mode, ***AzCamTool*** must be started and not be able to communicate with a server (simply close the server first). Then press the **Demo button** in the Server Communications window which will appear when a connection cannot be made to ***AzCamServer***. Any parameter file may be used in demo mode.

## Clock Files

***AzCamTool*** supports "clock files" which are text files which define voltages on the ARC clock and video boards. Clock files are supported for Gen2 and Gen3 systems (not Gen1). The files contain human readable clock names (usually set to the CCD manufacturer's naming conventions), minimum and maximum voltages for the DAC's (allowing hardware modifications from the default values), board and DAC addresses, and default values. The **Tools®ARC controller tools®Set Clocks and Biases** function reads, writes, and loads these files.

To create a new file it is best to copy and rename an existing clock file. The header at the top of the clock file defines the entries.

## Resetting the Controller

Pressing **Reset** opens a panel with options relating to resetting the controller to a known state. It is import to reset a controller after its power has been turned off and then back on, especially if ***AzCamTool*** is not restarted at the same time.

## Image Writing

Image files are only written to disk if the **Save Image** checkmark is on in **Preferences®Image File**. Click to toggle. The image name and image format are set on the **Filename** panel. Supported image types are FITS, binary, and Multi-Extension FITS (MEF).

There are three methods of sending image data to a disk file. These are determined by **Host** panel accessed from the **Filename** panel. In a system such as ***AzCam***, with a server and multiple clients, this is sometimes confusing.

Ø If **Use Server file system** is selected, the filename is considered to be valid on the **server** machine.

This may be a local disk file or the server may have mapped a directory or file (using Windows Explorer "map drive letter") to another machine, but the file must be accessible to the server via normal operating system naming conventions. (We have used Samba to map Linux drives to the Windows server). This is the most efficient way images can be written to the server machine

Ø If **Enable** next to **Remote File Host Name** is not selected, then the filename is considered to be valid on the local client; that is the machine which is running ***AzCamTool***. The data is sent over a socket to the client where a file writing application reads the socket data and writes the file. ***AzCamTool*** will automatically start such a file writing client (unless the **–nofileserver** command line option is specified). Note that while this mode works when the client and server run on the same machine, it is less efficient than specifying server mode file writing. **File host port** is the socket port to which image data is sent.

Ø If **Enable** next to **Remote File Host Name** is selected, the filename is considered to be valid on some other machine, specified by the **Remote File Host Name** option. Data is sent from the server to the file host machine, where a file writing application must be running. This mode is useful for archiving images on a separate machine, running a separate display server, or when the sending data to a machine other than where ***AzCamTool*** is running. **File host port** is the socket port on the remote machine which is used for listening for the images.

The Multi-Extension FITS (MEF) format is used to create image data files with multiple amplifier data. Each Header Data Unit (HDU) in the MEF file corresponds to data from one amplifier. MEF data writing mode is selected in the **Filename** panel by the **Image Format** parameter.

The **Amplifier Configuration String** (**Detector®Deinterlace**) is a code string used to identify the orientation of each amplifier in the mosaic. The string reads from left to right in the order data comes out of the controller. The code is: 0 => no flip; 1 => flip columns (flip in X); 2 => flip rows (flip in Y); 3 => flip both. As an example, a 2 CCD mosaic with 2 amplifiers per CCD, the first with 2 amps in the lower serial register and second with 2 amps in the upper serial register, would have a code string of "0123". If all amps were in the upper registers, the string would be "2323". This string is used to determine the keywords in the FITS header, as well as some deinterlace and display parameters.

It is also possible to create standard FITS image files from some multiple amplifier images that can be deinterlaced into a single, contiguous image. Selection is also by the **Image Format** parameter in the **Filename** panel. The built-in deinterlacing algorithms currently supports single, quad, split parallel and split serial configurations. Note that these modes may place overscan rows and columns within the image area, depending on amplifier configurations.

Data from a single amplifier device may be flipped on X, Y, or both, by setting the **Amplifier Configuration String** in the **Detector** panel. The same flips are applied to data from any amplifier in MEF mode, as described above. Generic flipping in FITS mode with multiple amplifiers is not yet supported.

Multiple amplifier modes in general do not support regions of interest (ROI's), although first and last rows other than the full detector often are acceptable.

## HELP

Most of the help for ***AzCamTool*** is on-line. Under the **Help** menu, the **Show Help Tips** and **Show Context Help** may be turned on (checked) or off.

The documentation for ***AzCam*** is contained in this manual, the AzCamTool User's Guide, AzCamServer Manual, and the AzCam Programming Manual, all of which are viewable from the ***AzCam*** start menu.

## DISPLAY

Image display is performed only through an image file server program. ***AzCamTool*** does not directly support a display tool because it does not receive image data, but it does automatically start ***AzCamFileServer*** (unless the –nofileserver command line option is specified) which can display images in conjunction with other programs.

***AzCamFileServer*** supports display to SAO's ***ds9*** program if ***xpans*** and ***ds9*** are running to the same machine. For Linux systems, ***xpans*** runs when ds9 starts. For Windows, it must be started manually. A batch file named *StartDisplay.bat* may be started from the ***AzCam*** start menu to perform the required functions under Windows for standard installations. Note that you must have **Save Image** selected in order to display an image.

***AzCamFileServer.exe*** is an executable version of the file server which can be run on a different machine than the ***AzCamTool*** client machine, to save and display images on the remote machine. The file host should be set in ***AzCamTool*** to the name of the machine where ***AzCamFileServer.exe*** is running.

## BINARY DATA

Binary image data may be sent to a remote machine for data analysis and other functions. This is in addition to the image writing process. **Preferences®Binary File®Send binary image** must be checked and the **binary host** and **binary port** must be specified. The binary files are unsigned short integers, little-endian, *Ncols* by *Nrows* in size, with no header information in the file. The exact data format is specified in the AzCam Programming Manual.

## TOOLS

The **Tools** menu provides special purpose camera functions. The **Focus Exposure** function creates a focus exposure which shifts the CCD between multiple exposures, interacting with instruments or telescopes as needed. **Guide Mode** controls the guide exposure mode which is optimized for the highest speed image acquisition and data transfer. **Run or edit a script** starts the script control function for text-based scripting.

**ARC controller tools** contain camera test and debug functions intended for laboratory or telescope checkout. **Set Clocks and Biases** changes clock and bias voltages (not supported for Gen1 systems), **Test**

**Shutter** exercises the camera shutter, **Test Fiber Communications** exercises the camera fiber communications, **Digital I/O** reads and writes the utility board digital I/O lines, and **Temperature Control** manipulates the detector temperature servo and other sensors. Other specialty functions are also called from this panel.

## CONTROLLER DSP CODE

The controller DSP code used for ***AzCam*** is based closely the ARC version 1.7 DSP code with modifications made to utilize a similar readout algorithm as used by the ICE CCD readout routines. The ICE *rdccd* DSP code was originally written by Skip Schaller at the University of Arizona’s Steward Observatory. This code uses a set of general DSP subroutines to clock a CCD. Arbitrary binning, region of interest, row and column underscan and overscan, as well as split parallel and serial registers are all supported within the same DSP program. Note that ***AzCam*** DSP files (and not those provided by ARC) must always be used since many functions read or write/memory locations or use commands which are defined only in these files.

The Motorola DSP development toolkits have been used to compile and link the DSP code for the PCI card and controller cards. The PCI card and Gen3 timing boards use the ASM56300 version while the other cards use the ASM56000 version, due to DSP differences.

The Motorola DSP assembler and linker in the directory structure required for the ***AzCam*** scripts in the system directories are located in *\AzCam\AzCamServer\systems\common* directory.

A **DSP Compiler** program is available from the **ARC controller tools** panel to quickly execute the batch files which compile and link DSP code. You must either press **Reset** on the ***AzCamTool*** panel or **Load** in **DSP Compiler** to load the newly compiled code into the controller.

## COMPARISON LAMPS

Comparison lamps are used when the *Image Type* is set to Comparison. The main window *Comparison* drop-down menu is used to select the lamp name. The lamp is turned on before an exposure and turned off after an exposure. This is accomplished through the I*nstrument VI* and so *Enable Instrument* must be checked in *Preferences*.

## CUSTOMIZATION

***AzCam*** supports customization of certain software components which can be loaded at run-time, allowing some level of user configuration beyond parameter files. The availability of custom functions is expected to increase with future releases. Current customs are described below.

- When the **Use Sound** option is enabled in **Preferences®Sound**, **AzCamTool** will play a sound (a WAV file) when the exposure is finished, readout is finished, or the program exits. The file names are *snd_exposure_finished.wav* , *snd_readout_finished.wav*, and *snd_goodbye.wav* and are located in the SYSTEM directory..
- Text based script files may be run from the **Run or edit a script** menu option under **Tools**. If **Use**

**AzCam Script** is selected then the script file is considered to be a text file of commands which are directly sent to ***AzCamServer***. If this option is not selected, then the script file name is passed to the batch file *startscript.bat* in the SYSTEM directory. Change the file *startscript.bat* to specify the scripting language.

Ø **PlugIns** are optional VI modules which can be ADDED to ***AzCamTool*** and called from the **Parameters->Manage PlugIns** menu. From the *Manage PlugIns* window it is possible to add PlugIns to ***AzCamTool*** by reading any parameter file which has PlugIns defined. Some plugins are built-in and enabled by checking their name in the *Manage PlugIns* window. PlugIns may also be added by adding them directly to the parameter file using a text editor. Saving the current configuration after adding PlugIns will then allow the new PlugIns to be available by default.

Ø **Customizations** are optional VI modules which REPLACE certain ***AzCamTool*** VI's. Customization VI's follow specific I/O requirements depending on their function and are not generally meant to be user-developed, but written as part of ***AzCamTool*** extensions for specific configurations. The customizable functions are: instrument, telescope, detector, exposure, focus, tools, and reset. These custom VI's are useful because the parameters and options required for different camera configurations relating to these functions are often quite varied. See the author for more information.

Ø Two special cases of customization modules are VI's which will interact with an instrument and a telescope, intended for such operations as moving filter wheels and focusing. The *InstrumentName* and *TelescopeName* keys in the parameter file specify these VI's. The functions are called when the **Instrument** or **Telescope** front panel buttons are pushed, when instrument or telescope keywords are needed for an image header, and when focusing or comparison exposures are made. The **Enable Instrument** and **Enable Telescope** options must be checked in **Preferences** to call these functions. These VI’s are selected by editing the parameter file.

Ø The names and control codes used to select comparison sources for Comparison exposure types can be specified in the [Comparison] section of the parameter file, as:

```
[Comparisons]
None=0
Fe-55=1
LED=2
```