JPL Project Number 102654

NLC:
NTM Logger / Chipper
User Guide

04 May 2005
Spiral Development Version: 1.2

Prepared for:
National Geospatial-Intelligence Agency (NGA)
Geospatial Intelligence Advancement Testbed (GIAT)
4600 Sangamore Road, Bethesda, MD 20816-5003

by:

Cartographic Applications Laboratory
Modeling and Data Management Systems Section 387
Optical Processing and Algorithm Target Recognition Task
Earth Science Technology Directorate (ESTD)

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109-8099

Questions concerning this document should be addressed to:

Thomas L. Logan, PhD
818-354-4032
tom.logan@jpl.nasa.gov

NLC Developers:

Nevin A. Bryant, Ph.D
Albert L. Zobrist, Ph.D
Thomas L. Logan, Ph.D
Walter L. Bunch
Richard K. Fretz

Contents

This Page Intentionally Left Blank.

1. SCOPE

1.1 Identification

This User Guide pertains to the task identified as the Jet Propulsion Laboratory (JPL) "Optical Processing and Algorithm Target Recognition," Task Order NMO-715744, Project Number 102654-2.01.01, "Test and Training," performed under management of the California Institute of Technology (CALTECH) via contract with the National Aeronautics and Space Administration (under NASA-CALTECH contract NAS7-03001), as prepared for the National Geospatial-Intelligence Agency (NGA), Innovision Directorate, Geospatial Intelligence Advancement Testbed (GIAT), headquartered in Bethesda, Maryland.

1.2 Overview

This document describes how to use spiral development version 1.2 of the NTM Logger / Chipper (NLC) software package, including a description of the underlying VICAR/IBIS software (upon which NLC is built), the concept of operations (CONOPS), Users Guide, and supporting documentation and software. In brief, the NLC software package creates small, user-defined image "chips" containing 'training' targets and 'test' imagery (that may contain targets), extracted from a parent NTM image. The chips are output in GeoTIFF format for input to the SAFIRE testbed, where a variety of ATR (Automatic Target Recognition) algorithms can be applied to determine the best approaches for automatically detecting desired object entities. The NLC is distributed via the AFIDS software package (Section 3.3 and 4.0).

1.3 License Agreement

NLC is built upon the VICAR/IBIS software. The United States Government has a non-exclusive, non-transferable, royalty-free worldwide license to VICAR. Third party use of VICAR is limited to uses for, or on behalf of NGA and the US Army (who hold a current license), but any further use must be negotiated with the NASA Patent Office. Other organiziations wishing to use NLC should obtain a VICAR license from Caltech/JPL. For licensing information, contact the Caltech/JPL Administrator for Software Licensing, 4800 Oak Grove Drive, Mail Code 202-233, Pasadena, CA 91109-8099, Phone 818-393-3424.

This Page Intentionally Left Blank.

2. APPLICABLE DOCUMENTS and DATABASES

2.1 Applicable Documents

(1) JPL, "NLC: NTM Logger / Chipper User Guide," Version 1.2, 04MAY05. This document.
(2) JPL, "AFIDS Automated Fusion Image Data System User Guide," Version 2.9, 04MAY05.
(3) JPL, "VICAR/IBIS Multi-Sensor Co-Registration System User Guide," Version 1.3, 01DEC03.
(4) JPL, "VICAR User's Guide," Document D-4186 Rev B, 14OCT94.
(5) JPL, "The Multi-Mission Image Processing Laboratory (MIPL)," and "The VICAR Image Processing System," http://www-mipl.jpl.nasa.gov/

2.2 Supporting Databases

Beginning with NLC Version 1.0, a JPL-prepared global database of SRTM-derived elevation cells (with voids filled) became available to support the NLC elevation requirement. This database, also known as the "AFIDS/SRTM Global DEM," contains global 1 arc-second (30m) DEM (Digital Elevation Model) files. The database occupies about 480GBs of disk space.

2.2.1 Digital Elevation Database

The DEM database is a one arc-second (~30m) SRTM (Shuttle Radar Topography Mission) digital elevation model (DEM) global dataset, that has had its "voids" (radar shadow areas) filled with NGA's "DTED-1" data, oceans flattened, and coastlines cut to DTED-1 boundaries. This global database uses DTED-1 data above 60 degrees Latitude, and uses "GLOBE" 1KM to fill in any remaining elevation holes. The resulting "srtm_filled" database comprises 19,240 1x1 degree files that are mosaicked together by NLC (as needed) to provide the topographic relief offsets that are essential to any image coregistration process. The data is provided in Geographic (Platte Carree) projection. This dataset is currently restricted to qualifying U.S. Government Agencies.

This Page Intentionally Left Blank.

3. NLC OVERVIEW

3.1 Software Foundations

NLC is built upon the VICAR/IBIS software. VICAR (Video Image Communication And Retrieval) is a comprehensive command-line image processing system originally developed in the mid-1960s to support the Nation's unmanned space exploration program. IBIS (Image-Based Information System) is a raster-based Geographic Information System (GIS) developed in the 1970s as a fully integrated subsystem of VICAR. While both systems have continued to evolve through the years, it is the IBIS portion that has focused on developing an integrated and comprehensive set of semi-automated image co-registration software capabilities. This software set is generally referred to as the "GT" routines for their integrated use and compatibility with GeoTIFF georeferencing algorithms.

VICAR makes use of the Transportable Applications Executive (TAE) for its command-line interface. TAE was developed by the NASA Goddard Space Flight Center and provides a standardized interface between the user and the large library of VICAR/IBIS application programs. While this interface and library are fully available to the user, a series of user-modifiable text "procedures" contain the command-lines that call the application programs to perform the co-registration process. Note these "procedures" use the ".pdf" suffix, but are *not* related to the Adobe Acrobat 'pdf' file format (VICAR usage precedes the Acrobat usage). Refer to Appendix A for a tutorial on the command-line use of VICAR.

NLC employees a user-friendly interface (gui) based on "tcl tk" (tool command language toolkit; pronounced "*tickle tk*") has been used to reduce the need for users to interface with VICAR command line procedures. "tcl tk" is copyrighted by the University of California, Sun Microsystems and others, and is classified by the DOD as "Commercial Computer Software," but carries no costs, license, or limitations on its use.

3.2 Operating Systems

The NLC version of VICAR/IBIS currently operates on SUN Microsystem computers running the Solaris 8 or 9 operating system (Solaris 10 is untested). A RedHat Linux version running on generic PC workstations is expected in mid-2005.

3.3 Concept of Operations

The NLC creates small, user-defined image "chips" containing 'training' targets and 'test' imagery (that may contain targets), extracted from a parent NTM (National Technical Means) image. The Chips are output in GeoTIFF format for input to the SAFIRE testbed, where a variety of ATR (Automatic Target Recognition) algorithms can be applied to determine the best approaches for automatically detecting desired object entities. Note that NTM images **larger** than 2GBs must adhere to the AFRL "Extract v7.5" NITF standard to be successfully processed.

The NLC provides several options for creating image Chips from parent NTM imagery. The options include: 1) Chipping the Entire Image; 2) Creating a number of Random chips based upon a user-supplied percentage of the parent image's original size, or a user-selected number; and 3) Creating chips based upon user-supplied target x,y locations. Note that in all three options, the output chip locations are 'chunked' out using the 'grid' as defined when chipping the entire image (i.e., chips are NOT CENTERED on target x,y locations). The user-supplied 'overlap' parameter prevents a target entity from being cut at the chip boundary by automatically generating an adjacent chip image. This often results in additional chips being generated (beyond the requested number, or expected number when using the 'percentage' option.

The NLC is a co-system that functions in parallel with the AFIDS (Automated Fusion of Image Data System), developed at the NASA/Caltech/Jet Propulsion Laboratory in the Cartographic Applications Laboratory (Section 387). The NLC and AFIDS systems are distributed together as one package. For information, contact: Dr. Nevin A. Bryant, 818-354-7236. Refer to the NLC and AFIDS Users Guides for further information.

3.4 Update Capabilties Since NLC Version 1.0

Since delivery of NLC version 1.0, several significant changes have occurred. These changes include: 1) A major gui simplification from five to three steps; 2) Removal of the DEM preparation step (assuming use of the AFIDS/SRTM global DEM database); 3) Removal of the NTM/Quickbird/Ikonos logging step; 4) Speed-up of the RPC georeferencing of the chip’s four corners; 5) An option for creating one (and only one) chip; and 6) An external (vicar) script (Section 6.0) for converting a given input NITF image to GeoTiff format with updated georeference label parameters.

4. NLC/AFIDS INSTALLATION AND LAUNCHING

Version 1.2 of NLC is provided through the installation of Version 2.9 of AFIDS. The NLC/AFIDS is currently operational on Sun Solaris 8 and 9 computer systems. A Sun System Administrator should perform the installation.

4.1 NLC/AFIDS Distribution Files

NLC/AFIDS is nominally distributed on a CD-ROM containing the following files:

- afids_2.9.tar.Z	(Complete AFIDS/VICAR software package)
- NLC_User_Guide.doc	(NLC User Guide in Acrobat & Word formats)
- AFIDS_User_Guide.doc	(AFIDS User Guide in Acrobat & Word formats)
- Readme_Install.txt	(Section 4 from the NLC User Guide)

4.2 File Locations

In the **/opt** directory (or wherever you want to install NLC), uncompress and untar the "afids_2.9.tar.Z" file.

```
uncompress  afids_2.9.tar.Z
tar  xvf  afids_2.9.tar
```

This creates a directory (about 1.51GB in size) called:

afids_2.9 NOTE: If you choose an alternate path to afids_2.9, be sure there are NO CAPITAL LETTERS in the path name!

Change your location to /opt/afids_2.9 (or wherever), and from within /opt/afids_2.9, install the software by typing:

```
./install
```

4.3 Shell and Path Suggestions

VICAR requires the "tcsh" shell. However, the NLC installation internalizes the tcsh shell such that users can operate NLC regardless of the shell they prefer. In order to run NLC *outside* the installation directory, the following Path information should be entered:

cshell or tcshell - Add this to the end of your ~/.cshrc file:
setenv PATH "/opt/afids_2.9:$PATH"

bash shell - Add this to the end of your ~/.bash_profile:
PATH=/opt/afids_2.9:$PATH
export PATH

NOTE: Replace "/opt" with your actual path to /afids_2.9.

sh shell - Add this to the end of your ~/.profile:
PATH=/opt/afids_2.9:$PATH
export PATH

Note that these changes will not take effect until a new shell is started.

4.4 Setting Up A User Directory

In the course of NLC processing, a directory and many scratch files are created. To contain these files, it is recommended that each user create a directory where they intend to (repeatedly) use NLC (and have ownership permissions).

1) Create an NLC Working directory, for example: mkdir nlc
2) Change your directory location to the new directory: cd nlc
3) Launch NLC (See next Section).

4.5 Launching NLC

To launch NLC, move to your working directory (/nlc). Open an "X" window and type:

./nlc (or ./afids to launch the afids software)

or type the full path (to /opt/afids_2.9) if you have not enhanced your PATH variable as described in Section 4.3 above.

The "NTM Logger/Chiper" main interface ‘gui’ should appear (Figure 4-1). Click the "?" help buttons for information on how to run NLC, or review the (expanded) Help Files in Section 5.

4.6 Exiting NLC

To quit and exit NLC, click the “Exit” button at the bottom of the main user interface (Figure 4-1).

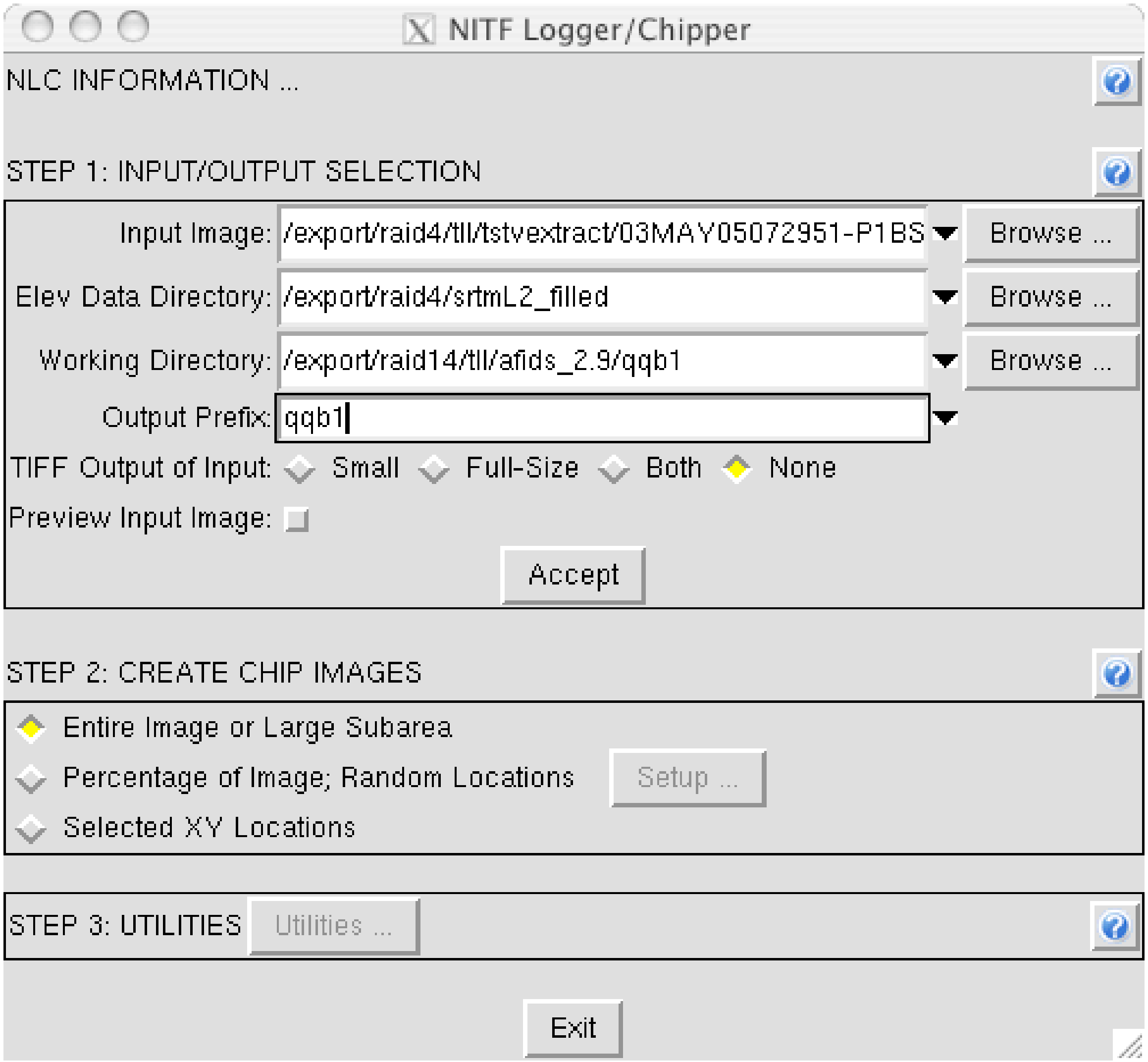

Figure 4-1: Initial NLC user interface. Note that fields 'remember' their last entry, and the 'black diamond' can be clicked to reveal the last three field entries.

This Page Intentionally Left Blank.

5. NLC HELP FILES (User Guide)

The following User Guide information is an expanded version of the Online Help files.

5.1 Check List – Here's What You Need To Proceed

5.1.1 √ **NTM Image: The directory path and filename** of an NTM image (in NITF format) containing target(s) of interest.

5.1.2 √ **DEM (Digital Elevation Model): Identify the Directory Path to the "AFIDS/SRTM L2 World DEM" database, OR if it's not available, Obtain Digital Terrain Elevation Data (DTED) CDROMs covering your area of interest**. With the directory path to your resident AFIDS/SRTM L2 World DEM database, the NLC software will automatically create the required DEM file without user intervention. However, if the World DEM is not available, the DTED CDROMs can be used directly, or the CD's files can be copied to a hard disk. If reading from the CDROM, be sure to put it in the computer's CDROM tray. If copied to disk, be sure to preserve the DTED directory structure (e.g., ~/dtedraw/dted/e042/34.dt1; Longitudes always have 3 digits) and use LOWER CASE characters. If the image's geographic area crosses a DTED CDROM boundary, the data from both CDROMs must be copied to disk. Both DTED Level 1 and Level 2 are supported. DEM data is used to improve the accuracy of the chip's lat/long georeference through an iterative processing of the parent image's RPC (Rational Polynomial Coefficients) metadata.

5.1.3 √ **Target List (Optional for "Test" Purposes): A single target location or list of target locations.** The target locations must be x,y pixel location(s) relative to the parent (input) NTM image (lat/long coordinates are not currently supported due to georeference inaccuracies). Multiple targets can be provided in a file, where the file format is standard ascii text, two columns (x,y) with NO additional letters, headers, columns, or other characters.

5.2 STEP 1: Input/Output Selection

The first step identifies all your input and output files, directories, and prefixes required for creating chips. After filling in the required fields and clicking the “Accept” button, the NLC software will begin logging the input NTM image into VICAR (which may take a minute or two---progress bars are provided).

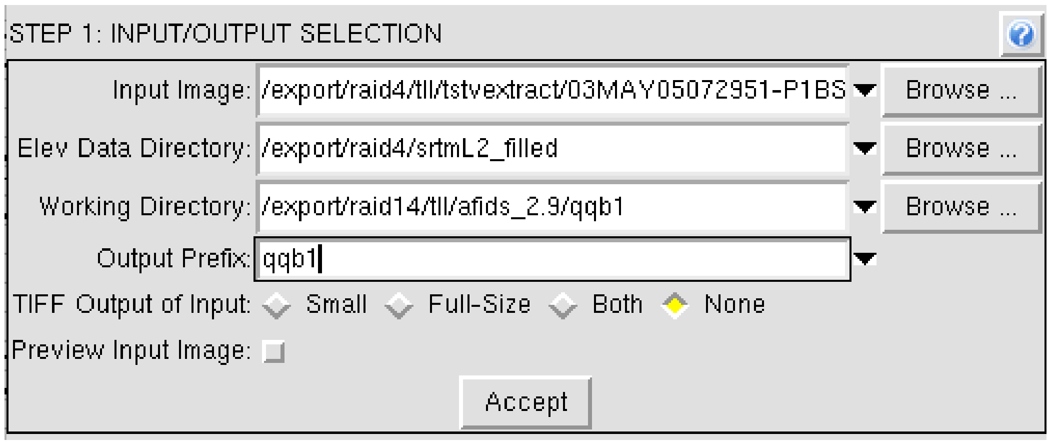

Figure 5-1: STEP 1 – Enter the required fields, then click the "Accept" button.

- The following parameters refer to the gui interface shown in Figure 5-1.

- **Input Image**: Use the ‘Browse’ button to locate the parent INPUT image in NTM format. The ‘black diamond’ will remember the last three field entries.

- **Elev Data Directory**: Use the ‘Browse’ button to locate the DIRECTORY where the “AFIDS/STRM L2 World DEM” database is located. If unavailable, locate a DTED CDROM or disk directory where DTED files have been placed (see Section 5.1.2). The ‘black diamond’ will remember the last three field entries.

- **Working Directory**: Use the ‘Browse’ button to locate or create a new DIRECTORY where all subsequent NLC output files (e.g., chips, etc.) will be placed. To create a new directory, use the Browse tool to find a location, then enter the new directory name after the path (in the field). Avoid names that start with an x, and don't use any special characters or spaces. The ‘black diamond’ will remember the last three field entries.

- Output Prefix: Enter a prefix to be attached to the beginning of every output chip’s filename. Avoid names that start with an x, and don't use any special characters or spaces. The ‘black diamond’ will remember the last three field entries.

- **TIFF Output of Input Image**: This option creates a full-size or 16x-reduction (“Small”) of the parent input image. Alternatively both sizes or ‘none’ can be requested. The file(s) are output in GeoTIFF format. The default is “None.”

- **Preview Input Image**: Click this button to display the parent input image for locating targets, identifying image subareas of interest, or for examining data quality. The image is displayed using the vicar “xvd” display program. Click the "Zoom Factor" button and choose 'Zoom to Fit' to see the entire image. Select "Tools/Stretch..." to display a variety of optional contrast enhancement options ('Gaussian' usually works best). Select "File/Exit" to quit and close the display program.

- Click the "**Accept**" button to begin the Step 1 processing. A progress box will appear while the parent input image is logged and converted to VICAR format (Figure 5-2).

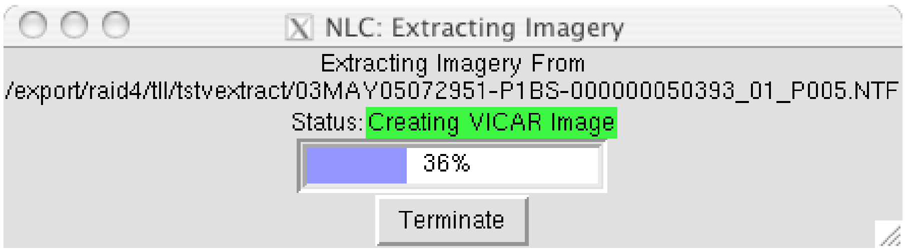

Figure 5-2: Progress Bar for Converting the Parent Image to VICAR format.

5.3 STEP 2: Create Chip Images

This step provides several options for creating image Chips from the parent NTM image. The options include: 1) Chipping the Entire Image; 2) Creating a number of Random chips based upon a user-supplied percentage of the parent image's original size, or a user-selected number; and 3) Creating chips based upon user-supplied target x,y locations. Note that in all three options, the output chip locations are 'chunked' out using the 'grid' as defined when chipping the entire image (i.e., chips are NOT CENTERED on target x,y locations). The user-supplied 'overlap' parameter prevents a target from being cut at the chip boundary by automatically generating an adjacent chip image. This often results in additional chips being generated (beyond the requested number, or expected number when using the 'percentage' option.

Figure 5-3: To Chip-Out an entire NTM image for ATR Test Purposes, Select the First Option in Step 2. For Random 'Test' Chips, Select Option 2. For Specific Training Chips, Select the Third Option.

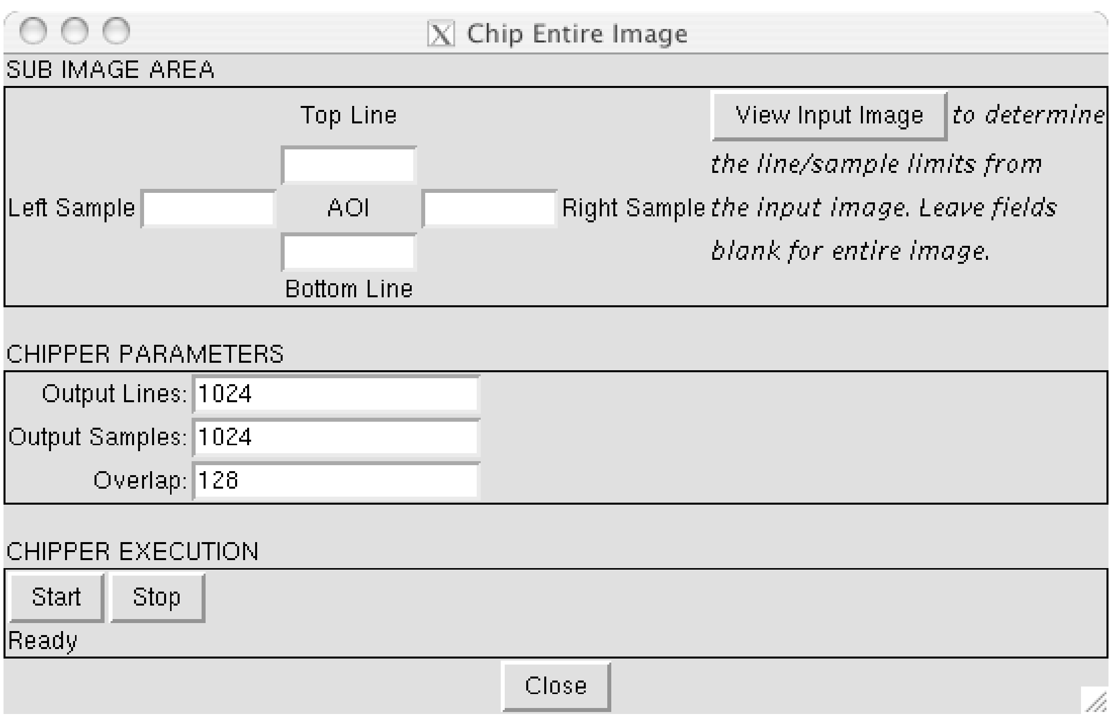

Figure 5-4: Dialog Box for Chipping-Out an entire Parent Image.

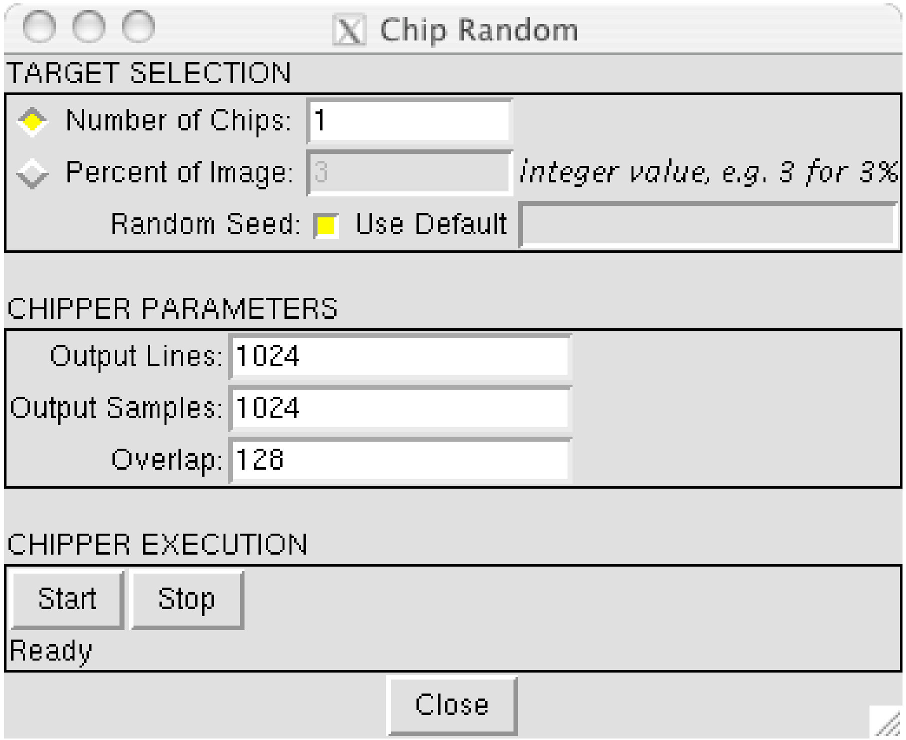

Figure 5-5: Dialog Box for Generating a Selected Number of Random Chips.

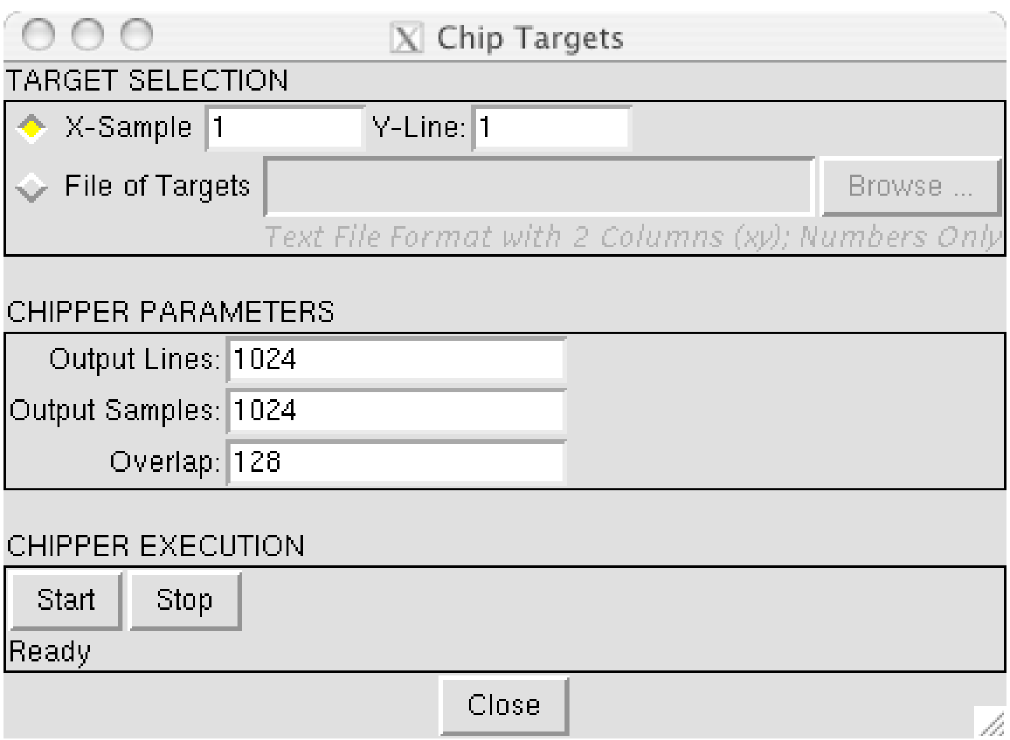

Figure 5-6: Dialog Box for Generating Specific Targeted Training Chips.

The following parameters refer to the gui interfaces shown in Figures 5-4 thru 5-6.

- Sub Image Area (AOI) Selection: Leave blank to process the entire image, or enter starting and ending image lines (rows) and samples (columns) to define a subarea for processing. To generate a Single Chip, enter the starting top line and left sample, then enter the number “999999” (six nines) in the two ending fields. A single chip will be generated with the size specified in the “Chipping Parameters” section. If the starting fields are provided with offsets, but the ending fields are blank, or if the ending fields are erroneously specified as smaller than the starting entries, the offset coordinates are subtracted from the parent image’s total number of lines and samples and used as the ending coordinates. If the ending coordinates are greater than the starting coordinates, but less than or equal to the Chip Size, the ending coordinates are set to the Chip Size. This will result in four output chips (not one as might be expected, unless the overlap is set to zero).

Click the "View Input Image" button to display the image and help identify the line/sample subarea limits, identify target locations, or evaluate image quality. In the display program, click the "Zoom Factor" button and choose 'Zoom to Fit' to see the entire image. Select "Tools/Stretch..." to display a variety of optional contrast enhancement options ('Gaussian' usually works best). Select "File/Exit" to quit and close the display program. The display may take a minute or two; be patient!

- Target Selection: Used to identify input target (train or test) locations.

- **Number of Chips:** The desired number of Random Chips to be generated from the parent image (Integer). Note that this process works by generating random x,y target locations, and if a location falls in a chip's “overlap” area, additional ships will be generated (producing more chips than requested).

- **Percent of Image:** The entered (integer, e.g, 3 = 3%) percent of the (approximate) total number of Possible chips will be produced at Random. Note that this process works by generating random x,y target locations, and if a location falls in a chip's “overlap” area, additional ships will be generated (producing more chips than requested).

- **Random Seed:** This is a number used to feed the random generator. The default seed number is the average of a 25x25 pixel box located at the center of the parent image. By unselecting the default button, the user can input an independent integer seed number.

- **X-Sample and Y-Line:** For generating a chip covering a single desired target location, enter the x (sample/column) and y (line/row) coordinates of the target. Note that if the location falls in a chip's “overlap” area, additional ships will be generated (producing more chips than requested).

- **File of Targets:** Use the "Browse" button to locate a Pre-Prepared text file containing multiple x,y target locations. This file MUST be standard ascii text, containing ONLY two columns (x,y), and with NO additional letters, headers, columns, or other characters. Use spaces, tabs, or commas to separate the x and y columns, and use a carriage-return and line-feed to separate the rows. The x,y coordinates are x,y target locations in the parent image.

- Chipper Parameters:

- **Output Lines**: This is the output chip's number of lines (y, rows). The default is 1024.

- **Output Samples:** This is the output chip's number of samples (x, columns). The default is 1024.

- **Overlap:** This is the number of pixels that will overlap between two adjacent chips, if the given target is at the edge of the primary chip (insures full spectral coverage of the target). The default is 128 pixels.

- Chipper Execution:

- Click the "**Start**" button to begin the process. Note that the first action is to automatically generate the required elevation file from the user supplied DEM Directory (Section 5.1.2 and 5.2). This may take a few minutes. A progress box will be displayed (Figure 5-7).

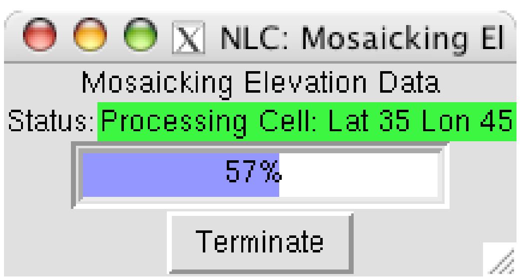

Figure 5-7: Progress Bar for Mosaicking DEM Data.

5.4 STEP 3: Utilities

This set of optional utilities includes Processing, Displaying, and Informational file tools.

5.4.1 Process Utilities

"Process Utilities" provide special image processing functions, including trimming and matching images in VICAR and GeoTIFF (.tif) formats (Figure 5-8), usually for export to another image processing software package.

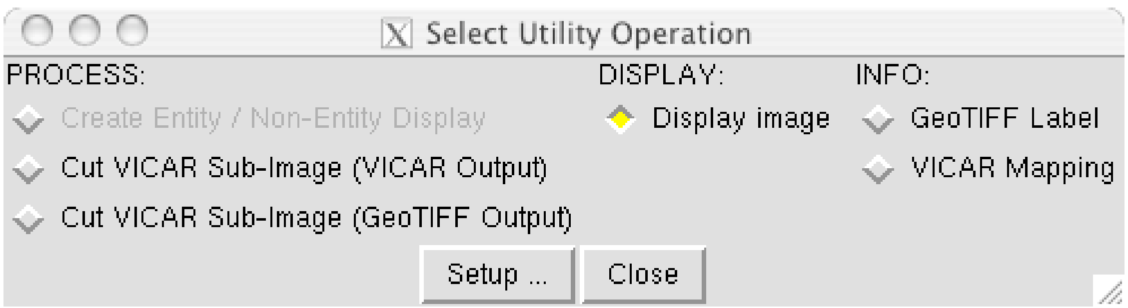

Figure 5-8: Optional Utilities selection box. Pick a Utility and click the "Setup…" button.

- Cut VICAR Sub-Image (VICAR Output): Use this option to cut a piece out of a parent image (in VICAR format). Both the input and output will remain in VICAR 'gt' format.

- Click the "**Setup**" button to display a dialog box. Use the browse buttons to locate the input image, usually the parent image in VICAR format ("prefix.img"), and an output filename and path.

- Click the "**View Input Image**" button to view the input band, and write down the Beginning Line and Sample, and Ending Line and Sample of the desired subarea. Enter those numbers into the AOI slots.

- Click the "**View image when complete**" button to automatically display the subarea image after it has been generated.

- Click the "**Start**" button to begin the process.

- Cut VICAR Sub-Image (GeoTIFF Output): Use this option to cut a piece out of a parent image (in VICAR format). The output will be converted to GeoTIFF (georeferenced TIFF) format for export to another image processing software package.

- Click the "**Setup**" button to display a dialog box. Use the browse buttons to locate the input image, usually the parent image in VICAR format ("prefix.img"), and an output filename and path.

- Click the "**View Input Image**" button to view the input band, and write down the Beginning Line and Sample, and Ending Line and Sample of the desired subarea. Enter those numbers into the AOI slots.

- Click the "**View image when complete**" button to automatically display the subarea image after it has been generated.

- Click the "**Start**" button to begin the process.

5.4.2 Display Utilities

"Display Utilities" are for displaying and comparing image datasets (Figure 5-8). The VICAR *xvd display* program is used for all image displays.

- Display Image: This option displays either a VICAR or GeoTiff formatted image.

- Click the "**Setup**" button to display a dialog box. Use the Browse button and a select an image to be displayed (usually the parent image in VICAR format ("prefix.img")).

- Click the "**Start**" button to begin the process.

5.4.3 Information Utilities

"Information Utilities" are for displaying image georeference and label/metadata information (Figure 5-8).

- GeoTIFF label: This option lists the metadata label of a selected image (either VICAR or GeoTIFF). The output is displayed in the process window.

- Click the "**Setup**" button to display a dialog box. Use the Browse button and a select an image to be processed.

- Click the "**Start**" button to begin the process.

- VICAR Mapping: This option lists a VICAR image's GeoTIFF map coordinates and georeference information (i.e., VICAR program 'gtlist'). The output is displayed in the process window.

- Click the "**Setup**" button to display a dialog box. Use the Browse button and a select an image to be processed (usually the parent image in VICAR format ("prefix.img")).

- Click the "**Start**" button to begin the process.

5.5 Chip Metadata label Fields

The following are the metadata fields stored in the label of an NLC-generated chip (inaddition to the default number of lines and samples, image format, and type, etc.). Note that the fields change slightly depending upon whether the RPCs are from version A or B. Type B is shown below.

NITF_IMAGEDATIM=
NITF_CETAG=
NITF_ACQ_DATE=
NITF_OBL_ANG=
NITF_ANGLE_TO_NORTH=
NITF_SUN_EL=
NITF_SUN_AZ=
NITF_MEAN_GSD=
NITF_DYNAMIC_RANGE=
NITF_ROLL_ANG=
RPC_FIELD1=
RPC_FIELD2=
RPC_FIELD3=
RPC_FIELD4=
RPC_FIELD5=
RPC_FIELD6=
RPC_FIELD7=
RPC_FIELD8=
RPC_FIELD9=
RPC_FIELD10=
RPC_FIELD11=
RPC_FIELD12=
RPC_FIELD13=
RPC_FIELD141=
RPC_FIELD142=
RPC_FIELD143=
RPC_FIELD144=
RPC_FIELD145=
RPC_FIELD146=
RPC_FIELD147=
RPC_FIELD148=
RPC_FIELD149=
RPC_FIELD1410=
RPC_FIELD1411=
RPC_FIELD1412=
RPC_FIELD1413=
RPC_FIELD1414=
RPC_FIELD1415=
RPC_FIELD1416=
RPC_FIELD1417=
RPC_FIELD1418=
RPC_FIELD1419=
RPC_FIELD1420=
RPC_FIELD151=
RPC_FIELD152=
RPC_FIELD153=
RPC_FIELD154=
RPC_FIELD155=
RPC_FIELD156=
RPC_FIELD157=
RPC_FIELD158=
RPC_FIELD159=
RPC_FIELD1510=
RPC_FIELD1511=
RPC_FIELD1512=
RPC_FIELD1513=
RPC_FIELD1514=
RPC_FIELD1515=
RPC_FIELD1516=
RPC_FIELD1517=
RPC_FIELD1518=
RPC_FIELD1519=
RPC_FIELD1520=
RPC_FIELD161=
RPC_FIELD162=
RPC_FIELD163=
RPC_FIELD164=
RPC_FIELD165=
RPC_FIELD166=
RPC_FIELD167=
RPC_FIELD168=
RPC_FIELD169=
RPC_FIELD1610=
RPC_FIELD1611=
RPC_FIELD1612=
RPC_FIELD1613=
RPC_FIELD1614=
RPC_FIELD1615=
RPC_FIELD1616=
RPC_FIELD1617=
RPC_FIELD1618=
RPC_FIELD1619=
RPC_FIELD1620=
RPC_FIELD171=
RPC_FIELD172=
RPC_FIELD173=
RPC_FIELD174=
RPC_FIELD175=
RPC_FIELD176=
RPC_FIELD177=
RPC_FIELD178=
RPC_FIELD179=
RPC_FIELD1710=
RPC_FIELD1711=

RPC_FIELD1712=
RPC_FIELD1713=
RPC_FIELD1714=
RPC_FIELD1715=
RPC_FIELD1716=
RPC_FIELD1717=
RPC_FIELD1718=
RPC_FIELD1719=
RPC_FIELD1720=
GTMODELTYPEGEOKEY=
GEOGANGULARUNITSGEOKEY=
GEOGRAPHICTYPEGEOKEY=
GTRASTERTYPEGEOKEY=
GEOGELLIPSOIDGEOKEY=
CHIP_PARENT_OFFS_LINE=
CHIP_PARENT_OFFS_SAMP=
NITF_CORNERLON1=
NITF_CORNERLAT1=
NITF_CORNERELV1=
NITF_CORNERLON2=
NITF_CORNERLAT2=
NITF_CORNERELV2=
NITF_CORNERLON3=
NITF_CORNERLAT3=
NITF_CORNERELV3=
NITF_CORNERLON4=
NITF_CORNERLAT4=
NITF_CORNERELV4=
NITF_CENTERLON=
NITF_CENTERLAT=
NITF_CENTERELV=
NITF_CENTER_GSD=
NITF_CENTER_ELEV_LOOK_ANG=
NITF_CENTER_AZIM=
NITF_CENTER_UPISUP=
MODELTRANSFORMATIONTAG=

6. Command-Line "ChipScript" Procedure

6.1 Chipscript Introduction

"chipscript.pdf" is a VICAR procedure for directly generating a chip without using the NLC gui interface. The procedure reads an NTM image and corresponding elevation file, uses the input image's rational polynomial coefficients (rpcs) to improve the output chip's georeference and four-corner latitudes and longitudes, and outputs a GeoTIFF file. Image subsetting is not supported.

6.2 Chipscript Syntax

The syntax for running the chipscript is:

```
chipscript  input.NTF  output.TIF  elev.IMG
```

Where:

- input.NTF is the input NTM path and filename.
- output.TIF is the output chip's path and filename.
- elev.IMG is the path and filename of a DEM covering the input image.

Example:

```
chipscript  03MAY05_P005.NTF   quickbird_03may05.tif   /export/raid4/qb_dem.img
```

6.3 Command-Line VICAR

The chipscript procedure requires that VICAR be available so that it can be invoked via the command-line. To make VICAR accessible, type the following in a 'tcsh' shell:

source /opt/afids_2.9/setup_afids_env.csh (where **/opt** is your path to /afids_2.9)

Then, to launch chipscript, type at the Solaris prompt:

```
vicar  –s  "chipscript  input.NTF  output.TIF  elev.IMG"
```

Note that the entire command must fit on one line.

The chipscript procedure can also be run from within VICAR. Refer to Appendix A for a tutorial on running vicar programs and procedures.

6.4 Elevation File Preparation

The third input used by chipscript is an elevation file that entirely covers the input image. This file can be generated using the AFIDS gui from DTED data or the AFIDS/SRTM L2 World DEM database. Launch AFIDS (see Section 4.5) by typing:

./afids

When the main gui interface displays, enter a dummy Session ID, then in Step 2, select either the "DTED Formatted DEM" option or the "AFIDS/SRTM World DEM DB" option. Use the DTED option if you only have DTED CDROMs, and use the AFIDS/SRTM option if you have the world DEM database. Both dialog boxes are similar. Figure 6-1 shows the DTED option.

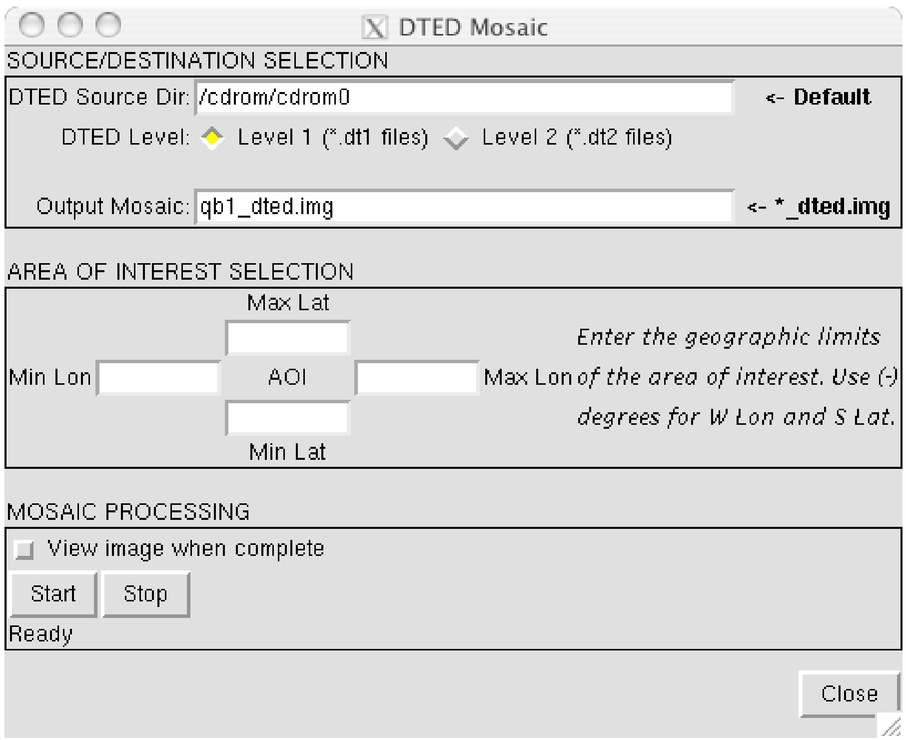

Figure 6-1: AFIDS dialog box for creating elevation files.

This Page Intentionally Left Blank.

APPENDIX A:
VICAR/IBIS Tutorial

While the gui interface reduces the need for most direct interaction with VICAR command-line procedures, some interaction may be necessary (or desired) depending upon the situation. For this purpose, a quick introduction into the use of VICAR syntax is provided. Refer to the "VICAR User's Guide" (vug.tar in /afids_2.9/doc) for comprehensive instructions, reference information, and a list of basic programs. It is assumed that NLC/AFIDS has been fully installed and tested prior to this point.

To reveal VICAR from behind NLC/AFIDS, type in a 'tcsh' shell (from any directory):

    source /opt/afids_2.9/setup_afids_env.csh   (where /opt is your path to /afids_2.9)

If you aren't currently using tcsh, one is available in /opt/afids_2.9/bin.

Then launch VICAR by typing **vicar** at the Unix prompt. The system will respond with a Welcome Banner, several messages, and the vicar prompt:

%VICAR>

To leave/quit VICAR, type **exit**

%VICAR>**exit**

To execute a Unix command while in VICAR (except for 'ls'), type **ush** and the command:

%VICAR>**ush df -k**

To execute a VICAR application program, type it's name and parameters at the command line. For example, VICAR program "gen" is used to generate a test image with the output filename "a", 552 lines (rows), and 818 samples (columns):

%VICAR>**gen out=a nl=552 ns=818**

To verify the new file, read the file's label with the program "label-list":

%VICAR>**label-list inp=a**   or just:
%VICAR>**label-l a**

To list the first 20 lines by 10 samples of the image "a" to your screen, use VICAR program "list":

%VICAR>**list a size=(1,1,20,10)**

To display the image on your screen, use the interactive VICAR display program "xvd":

%VICAR>**xvd a**

To identify all the parameters associated with a program, use the tutor mode (below). At the bottom of the tutor screen, type "**help *parameter***" to display information about the parameter (then **exit** to leave the parameter description). The tutor mode can also be used to run the program by filling in the parameter fields with values. Type **exit** to quit tutor mode, or **run** to execute the program (after filling in the parameter values):

%VICAR>**t gen**   (or **tutor gen**)

To determine the function of a particular program (a partial list of programs can be found in the "VICAR User's Guide"), use the help mode (**exit** to quit):

%VICAR>**h gen**   (or **help gen**)

The directory in which VICAR is run must have a "ulogon.pdf" to specify where VICAR searches for its programs. The location of basic VICAR programs is hardwired, but most of the co-registration programs are in a separate directory that must be specifically identified, for example "/opt/vdev". An example ulogon.pdf is:

```
procedure
body
write  "Executing from /home/tll"
setlib-delete library=($R2LIB)
setlib-add library=(/home/tll,/opt/vdev,$R2LIB)
end-proc
```

The "ulogon.pdf" demonstrates the basic syntax of a VICAR command-line 'procedure' or 'script'. The VICAR 'procedure' is a text file with the suffix ".pdf" (that predates Adobe Acrobat). The file must have the keywords "procedure", "body", and "end-proc", with some VICAR applications after the 'body' keyword. (The space between 'procedure' and 'body' is reserved for special declaration statements, if needed.) An example procedure is:

```
procedure
body
gen      a nl=250 ns=255
label-l  a
list     a size=(1,1,20,10)
hist     a
end-proc
```

The procedure can be written using any text editor such as **textedit**, **joe** or **vi.** Just be sure to end the filename with a ".pdf". To run the procedure in realtime, simply type its name at the VICAR prompt. To obtain a log of the running process, run the job in batch mode (below). If the procedure were named "testjob", the output logs would be named "testjob1.log" and "testjob1.log.stdout". (Note, logs can be printed using the **cat** command while the job is running.)

%VICAR>**testjob|run=batch|**

VICAR uses the exclamation point (!) to identify comment lines in the procedure. This can be useful for adding notes and descriptive text as well as bypassing program calls. The "goto *here*" command is also useful for jumping to different locations in a procedure. For example:

```
!This is a test
procedure
body
gen      a nl=250 ns=255
!Jump over the label-list program
goto next           !Any single word could be used in
!                   place of 'next'.
label-l  a
next>               !The ">" is required.
hist     a
end-proc
```

If a VICAR command-line must continue to a second line, place a plus sign (+) at the end of the first line to tell VICAR to continue to the next line.