The Forestry Corp.

Watershed Assessment Model

GIS Application System Guide

August 1997

TABLE OF CONTENTS

1.0 Introduction

The Watershed Assessment Model (WAM) is an ArcInfo AML application used to generate watershed and stream characteristics from existing digital elevation, stream network, and plot location data. The application is intended to run in a client-server environment where all processing is performed using the WAM interface on an ArcInfo UNIX server, and the results are viewed and queried from ArcView on a networked PC.

The balance of this document will discuss some of the issues faced during the development of the application, the programming standards, how to install and launch the application, how to make changes in the application, and will illustrate program flow. The WAM System Guide is one of three documents delivered with the installation of WAM. The *WAM Final Report* describes the general approach and methods involved in completing the project from the initial concepts through to the completion of development. The *WAM User Guide* describes how to use the application, from launching the main menu to detailed descriptions of the resultant data. These documents can be referenced when information beyond the scope of this document is required.

The System Guide is written for the person(s) responsible for application administration. The responsibilities of the administrator include installation, system environment configuration, and performing any changes to the application which may be required. To perform these tasks, the administrator should be knowledgeable in both UNIX system administration and in ArcInfo, although the standard application startup and maintenance is not complex. The System Guide should be used in conjunction with the *WAM User Guide* to make the administrator aware of how to use the application, it's resultant data, and the potential impact of any changes made to the application.

2.0 Development Issues

2.1 Introduction

This section discusses unforeseen technical issues which arose during development of the application. Each issue is described generally, then an explanation as to how each was resolved and the impact, if any, to the original application framework.

2.2 Project Area Selection

Originally a single elevation dataset for the entire Foothills Model Forest area was to be built from which all WAM application analysis would be completed. Using the system in place at the time, the dataset was simply too large to process as insufficient virtual memory could be allocated for the ArcInfo *topogrid* command. This command is used to build hydrologically correct elevation grids from elevation point data. To solve this problem, functions were added to manage WAM analysis by project. The projects are defined by a geographic area for which an elevation grid is generated and all subsequent analysis is completed using the geographic extents of this dataset. The problem which can be encountered with this approach is that the user may select a project area which does not fully encompass the *drainage basin* (see User Guide glossary) for a selected point and will in turn calculate incorrect area values. The potentially incorrect basin area values are written to the output plots table, and are also used to calculate feature densities (density of a linear or point feature inside a basin area). To minimize the chance of an incomplete project area being defined, the selected areas are buffered and then overlaid with a Foothills Model Forest wide basin coverage to determine all overlapping basins. The extent of all selected basins is then used to define the extent of the project. This problem will likely only be completely solved once a dataset covering the entire management area can be created.

2.3 Generating Stream Routes

The application functionality includes the measurement of distances up and down stream from a point along a stream. To accomplish this the application applies ArcInfo's *dynamic segmentation* data structures to build *route* features from contiguous streams sharing the same order. Originally it was thought that the order *routes* could be generated using the single *arcroute* command and specifying the order item as the grouping attribute. But *arcroute* is unable to base *route* direction on the underlying arc direction meaning that using this command would result in some streams flowing uphill. The only method to build route systems and control the direction of individual routes is using the *makeroute* command in Arcedit and specifying an X and Y starting coordinate for the new route. Therefore an extra step was introduced (*build_routes*) to programmatically generate *routes* for individual stream order using the *makeroute* method. This workaround is very time consuming and processor intensive. The most likely fix for this problem is added functionality to the arcroute command to control route direction based on arc direction.

2.4 Arc Vertex Limit

During the *calc_attribs* process, stream lengths are calculated for those streams closest to input plot locations. Originally, the lengths were simply extracted from the AAT of the resultant coverage from the *eventarc* command. Currently there is an ArcInfo limitation which restricts a single arc to be comprised of a maximum 500 vertices. Once this limit is reached, the system automatically splits the arc. A fix for this problem sums the length of all arcs belonging to a single *route*. This process does not require significant additional processing time.

2.5 ArcView Line Graph Limit

During the *calc_attribs* process, stream profile tables are generated for buffer and source to mouth stream *routes*. Arcview can only generate line graphs for a maximum of 100 samples points. Although an option exists with the *surfacxsection* command to define a distance for sampling along the stream arcs, it was discovered that additional points were added in the tables and therefore could exceed the 100 point maximum. An additional process was added to ensure that there were no sample points closer than the specified sample distance. This process impacts processing time slightly, depending on the number of streams in the profile table. This problem can only be resolved by removing the limit of sample points for line graphs in Arcview or by a change to the *surfacexsection* functions which guarantees exact sample distances in the arc profiles.

3.0 Programming Structure

3.1 Introduction

The application administrator should be aware of the program structure of the application in the event they wish to perform changes to the AML code. These changes may be required for example, if the application is moved to another site, the input data structure changes, or to take advantage of updates to the ArcInfo GIS software. All application code is written in AML and is not encrypted allowing complete access and flexibility to the administrator should the need for changes arise. This section will discuss the programming and documentation standards used for development to assist the administrator in navigating through the application code.

3.2 Programming Model

The application programming structure is modeled after ArcInfo's Arctools. Each task or step in the application is performed using a tool designed specifically for this task. A tool is comprised of an AML, a menu and a help file, all with the same prefix name (eg. build_routes.aml, build_routes.menu, build_routes.hlp). All tools are launched, executed, and exited by running the AML and specifying a command line argument which executes a routine in the AML of the same name. All tools contain the standard routines described in the table below.

Standard Tool Routines

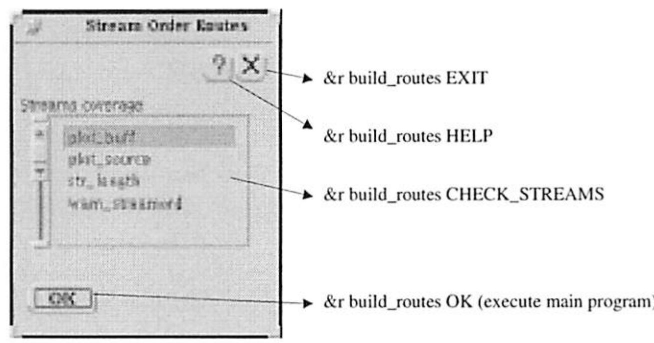
Routine	Function
INIT	launch tool menu (default)
USAGE	returns valid command line syntax
HELP	pops up a text window with menu help (if the help file exists)
EXIT	closes the menu, closes all sub-menus spawned by the tool, and deletes all tool variables
BAILOUT	called in the event of an AML error

Additional functions required by the tool such as validating widget input is run using the same syntax to execute the validation routine. Below is an example of the routines called from a menu. The following command would launch the *build_routes* menu.

```
Arc: &r build_routes init
```

Once the menu is present, all tasks associated with the widgets are called from the menu program using the standard syntax.

Example of menu program calls



The Arctools programming structure allows all tools to operate independently and are therefore more easily incorporated into existing applications. This advantage also allowed the easy integration of existing generic Arctools into the WAM application. All tools relating to data management were implemented in this way and can be identified in the AML header information.

3.3 In-Line Documentation

Header and in-line documentation also follow a similar standard to that used in Arctools. Standard header information is illustrated below.

```
/*------------------------------------------------------------------------
/*             The Forestry Corp
/*------------------------------------------------------------------------
/*    Program: AML_NAME.AML
/*    Purpose:
/*
/*------------------------------------------------------------------------
/*      Usage: AML_NAME INIT {'position'} {'stripe'} {MODELESS | MODAL}
/*      Usage: AML_NAME <routine_name>
/*
/*  Arguments: routine  - name of the routine to be called.
/*             position - (quoted string) opening menu position.
/*             stripe   - (quoted string)  menu stripe displayed.
/*             MODELESS | MODAL - keyword indicating modality of menu thread
/*
/*    Globals: (none external)
```

```
/*------------------------------------------------------------------------
/*     Calls: AML_NAME.MENU
/*------------------------------------------------------------------------
/*     Notes:
/*------------------------------------------------------------------------
/*   History: Author - date - original code
/*========================================================================
```

The history section of the header should be maintained to track any changes which are made to the AML after the initial installation. In addition, most general tasks in the code are described using in-line documentation

3.4 Temporary Files

Many of the processes in the application require the generation of temporary datasets in the form of grids, coverages, info tables, and system files. It is strongly recommended that the user not create files which start with 'XX' as this is the convention used in the application and by most ArcInfo processes to name temporary data. If all processes complete correctly there should be no residual temporary files. If for some reason a process has failed and temporary files do remain, they can be removed.

3.5 Variables

The standard global variable naming convention also adheres to the Arctools model. All global variable names have two components. The first part being the same name as that of the tool in which it was created, and the second describing what the variable represents. For example, *.build_basin$outcov* is set and referenced in the *build_basin* tool (AML or menu) and will represent the output coverage. All variables not required outside the program in which they were set is defined as a local variable. This convention allows easy identification and deletion of variables set by a particular process.

4.0 Installing And Running The Application

4.1 Introduction

This section will describe the system requirements for to running the WAM application, how to install the application code, how to set the system environment to run the application, how to launch the application, and finally the program flow.

4.2 System Requirements

The WAM application was developed in and requires ArcInfo (Rev 7.0.4) on the UNIX operating system . In addition to the core ArcInfo program the application requires the GRID raster modeling module and the TIN (Triangular Irregular Network) three dimensional surface analysis module. The application will run in any X graphic environment that is supported by ArcInfo. Menu and form placement on the display may not be consistent across different X environments, but this will not affect functionality. The actual application code requires approximately 800K of disk space and can reside anywhere on the system under a common parent directory. Running the application does require a significant amount of both memory and disk space as many of the processes are very CPU intensive and create large temporary and resultant datasets. For example, an 25m resolution elevation grid with 2.2 million cells requires about 3.6Mb but this may vary depending on theme variability.

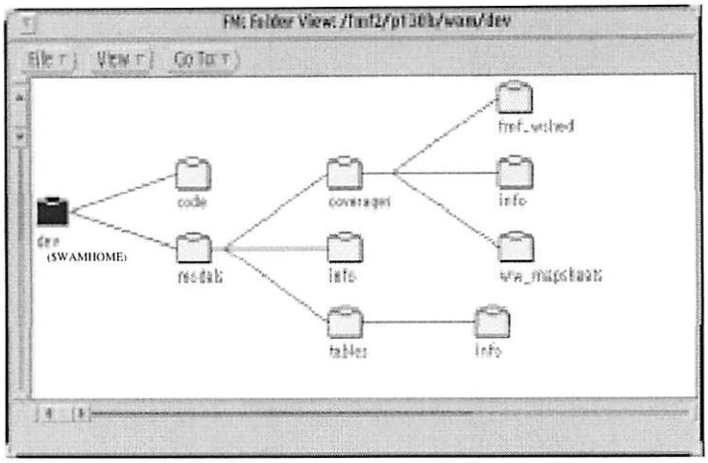
4.3 Application System Setup

WAM uses UNIX system environment variables to dynamically set the location of the application code, tables and data. In the user's .cshrc startup file, the *$WAMHOME* environment variable must be set to the directory which contains the subdirectories *code* and *models*. Examples are shown below, you must substitute the correct locations for your site.

setenv WAMHOME /opt/app/wam

Once the application is installed, the directory structure should be the same as that illustrated below.

Directory structure



The arc *topogrid* command used to generate the hydrologically correct elevation grid may require additional virtual memory allocation to complete successfully. To override default memory allocation limits it is recommended that the GRIDALLOCSIZE system environment variable be set to a value of at least 100, if it has not previously been set. This variable is set also in the .cshrc file using the same syntax as described for the WAMHOME variable.

setenv GRIDALLOCSIZE 100

4.4 Application Data

The application uses two ArcInfo coverages located in the *$WAMHOME/models/coverages* directory to determine the geographic extent of the DEM data required to build an elevation grid for a project. When the user selects a project area, either from an existing polygon coverage or from on-screen digitizing, the program first overlays that area with the *fmf_wshed* coverage to determine all overlapping watersheds. The extracted polygons are then buffered to try and minimize the possibility of a small portion of a watershed being eliminated due to the large scale digitizing from which the *fmf_wshed* coverage was created. This buffer result is then overlaid with the *ww_mapsheets* coverage to determine all overlapping 1:20000 mapsheets. Using the list of mapsheets the program then builds the elevation grid by extracting point data from the TINs which are available by 1:20000 mapsheet. The program was modeled this way as FMF's best DEM data existed in the TIN structure, by 1:20000 mapsheet. If the format for the DEM data changes there will have to be some custom programming performed in the *build_elevgrid* tool which allows the system to read other data formats.

At installation time the names of the two coverages mentioned above, and the location of the TIN datasets by mapsheet will have to be entered into the *wam_defaults* table. Below is a table listing the three variables in the table for which the values will have to be set at installation and if the coverage names are ever changed.

Variable	Description	Value

.build_elevgrid$masterwshed	master watershed coverage	$WAMHOME/models/coverages/fmf_wshed
.build_elevgrid$sheetbdys	name of coverage used for TIN selection	$WAMHOME/models/coverages/ww_mapsheets
.build_elevgrid$tinpath	path to all available tins	/fmf4/tins

4.5 Launching the Application

Once the *$WAMHOME* environment variable is set, the user can launch the application from within ArcInfo as described below.

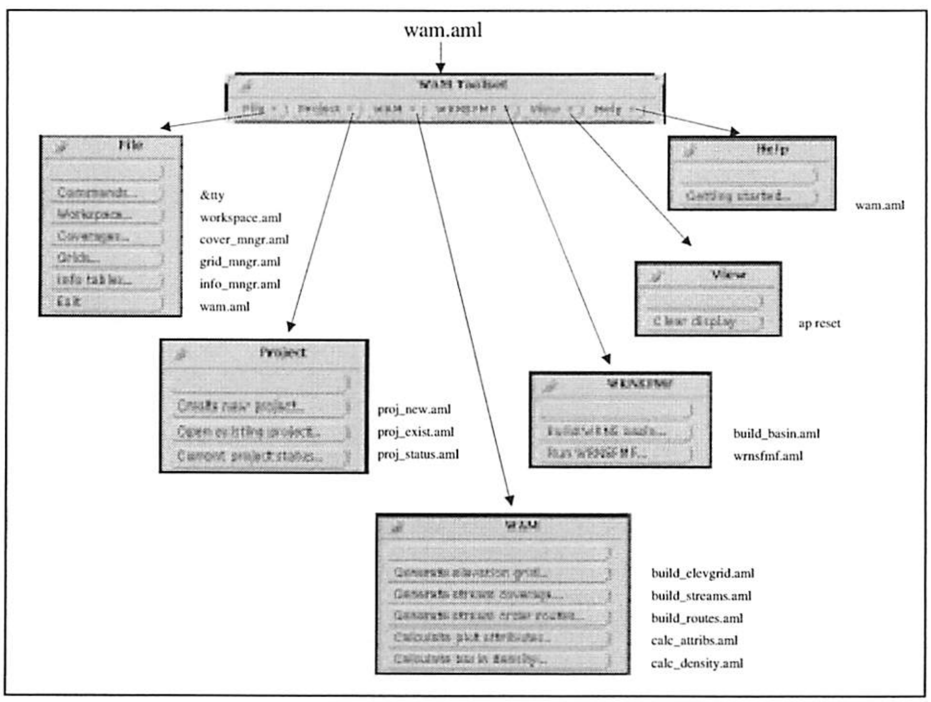
```
Arc: &r $WAMHOME/code/wam
```

The administrator may wish to set the above command in an alias which could be called either from the X window main dropdown menu or via a single command in a shell window. Either of these options will make it faster and simpler for the user to launch the application.

4.6 General Flow

The diagram below illustrates the commands executed by each of the main menu options. The launching program *wam.aml* is executed at the arc prompt as described in the previous section.

Main Menu Program Calls



5.0 Application Tables

5.1 Introduction

This section will instruct the system administrator on how to make *standard* changes to the application, and will discuss how the application tables are used.

5.2 Application Defaults

Standard changes are parameters which were identified during the development as having the potential to change in the event, for example, of the input DEM data resolution changing. To make changing these parameters simpler than altering program code, the parameter value and the variable used to hold the parameter value are stored as records in an INFO table. The *wam_defaults* table stores these parameters, where applicable. Some of the parameters are static, meaning that in order to change them the administrator must manually edit the table to change the value. Others are simply default values which initially populate menu widgets to simplify the interface. See Appendix A for a listing of the *wam_defaults* table and it's default values at installation. Below is a description of the variable name and what each represents. The *var_name* prefix will indicate the tool with which each is associated.

Wam defaults table

VAR_NAME	DESC
.build_basin$outcov	*Default name of generated basin coverage.*
.build_elevgrid$aspect_cov	*Default name for output aspect coverage.*
.build_elevgrid$cellsize	*Default elevation grid cell size.*
.build_elevgrid$elev_class	*Default name for classified elevation coverage.*
.build_elevgrid$elev_grid	*Default output name for elevation grid.*
.build_elevgrid$masterwshed	*Name of coverage of watersheds for the FMF used in finding all watersheds overlapping the selected project area.*
.build_elevgrid$sheetbdys	*Name of coverage used to find all required TIN data structures to build an elevation grid for the project area.*

.build_elevgrid$tinpath	*Path to all available tins datasets.*
.build_elevgrid$wshed_cov	*Default name for output watershed coverage*
.build_elevgrid$wshedbuff	*Distance to buffer selected watersheds to ensure that they fully encompass the watersheds defined by the DEM.*
.build_streams$flowtol	*Default number of accumulated cells required to assign a cell as part of the flow path when generating streams from the elevation grid.*
.build_streams$order_alg	*Default stream ordering method.*
.build_streams$stream_order	*Default name of output stream coverage.*
.calc_attribs$armtol	*Search tolerance for 'arcrroutemeasure' command used to generate measures along routes from point features (the plots).*
.calc_attribs$attrib_plots	*Default name of output point coverage from the 'calc_attribs' process.*
.calc_attribs$basincov	*Default name of output basin and drainage area coverage.*
.calc_attribs$buffer	*Default buffer distance up and down stream from the plot location.*
.calc_attribs$neartol	*Default search tolerance to snap plots to streams using for the arc 'near command'.*
.wrnsfmf$aspectitem	*Aspect item on aspect PAT*
.wrnsfmf$bufdist	*Buffer distance around harvest blocks*
.wrnsfmf$htitem	*Height item name on fc PAT.*
.wrnsfmf$outdbf	*Default output dbf file.*
.wrnsfmf$site	*Site value.*
.wrnsfmf$sp1item	*First species item name on fc PAT.*
.wrnsfmf$yrcutitem	*Year of cut item on the harvest PAT.*

The administrator should be aware of the consequences of changing any of the default parameters, specifically those which impact output results such as flow tolerance. These changes should first be discussed with end users to ensure that they are aware that any new datasets will be created using the altered parameters.

5.3 Project Tracking

As discussed in section 2.2 each WAM run that is performed on a project basis. This was done as a measure to solve the problems incurred by datasets which were simply too large for the ArcInfo software to process on available hardware. Prior to any analysis in the application the user must define a new project or select an existing one. This will automatically, in the event of a new project, create a new workspace, and then take the user to the appropriate workspace. The location path and name of the project workspace, any new datasets created by the application, and the parameters used to generate new datasets are written to an application INFO table called *Projects*. When defining a new project the menu allows the user to assign characteristics or metadata to the project which is also stored in the table.

5.4 Stream Flow Calculations

During the *calc_attribs* process, flood volumes, peak flows, and monthly flows are calculated at each plot location. The calculations are based on the generated basin areas and formulae developed for sites within the Foothills Model Forest which are published in the *Hydrologic Operational Manual*.

Below is a table which describes the application tables which are associated with the flow calculations. These tables should only be altered by persons knowledgeable in how the values are derived. The full tables and their original values are listed in Appendix A.

Table name	Description
peak_flow.vol	Peak flow hydrography volumes for three elevation zones. This is required for peak flow calculations.
monthflow_a1250.vol	Mean monthly unit flows for areas above 1250m. This is required for monthly flow calculations.
monthflow_b1250.vol	Mean monthly unit flows for areas below 1250m. This is required for monthly flow calculations.

Appendix A

Application Tables

To follow is a listing of all application tables. These tables are located in the $WAMHOME/models/tables directory.

wam_defaults – This table is used to set default values for menu widgets and for site-specific installation parameters.

VAR_NAME	DESC	VALUE
.calc_attribs$armtol	search tolerance for arcrroutemeasure	10
.calc_attribs$buffer	buffer distance from plot along stream	500
.calc_attribs$attrib_plots	name of output, attributed plot coverage	wam_plots
.calc_attribs$neartol	search tol to snap plots to streams using near	500
.calc_attribs$basincov	name of output basin and drainage area coverage	wam_basins
.build_elevgrid$cellsize	elevation grid cell size	25
.build_elevgrid$elev_grid	default output name for elevation grid	wam_elevgrid
.build_elevgrid$elev_class	default name for classified elevation grid	wam_elevclass
.build_streams$flowtol	cell tolerance for flowaccumulation command	100
.build_elevgrid$sheetbdys	name of coverage used for tin selection	$WAMHOME/models/coverages/ ww_mapsheets

.build_elevgrid$masterwshed	master watershed coverage	$WAMHOME/models/coverages/fmf_wshed
.build_elevgrid$wshedbuff	buffer distance for selected master watershed poly	500
.build_elevgrid$tinpath	path to all available tins	/fmf4/tins
.build_streams$stream_order	name of output stream coverage	wam_streamord
.build_streams$order_alg	stream ordering method	STRAHLER
.build_elevgrid$aspect_cov	default name for output aspect coverage	wam_aspect
.build_elevgrid$wshed_cov	default name for output watershed coverage	wam_wshed
.wrnsfmf$bufdist	buffer distance around harvest blocks	100
.wrnsfmf$yrcutitem	year of cut item on the harvest PAT	yrcut
.wrnsfmf$aspectitem	aspect item on aspect PAT	aspclass
.wrnsfmf$site	default site value	fair
.wrnsfmf$sp1item	first species item name on fc PAT	sp1
.wrnsfmf$htitem	height item name on fc PAT	ht
.build_basin$outcov	default name of generated basin coverage	wrns_basin
.wrnsfmf$outdbf	default output dbf file	wrns_dmp.dbf

aspect.remap – Used to generate the output aspect class polygon coverage and to create aspect class polygons when generating the WRNSFMF output dataset.

VALUE	SYMBOL	ASPECT
1.0	0.0	F
2.0	96.0	N
3.0	98.0	EW
4.0	99.0	S
5.0	90.0	EW

floodvol.remap - This table is used to remap the area inside the drainage area for calculating the flood volumes.

VALUE	CLASS
1250.0	1250.0
1500.0	1500.0
4500.0	4500.0

monthflow_a1250.vol – Used in conjunction with drainage area to calculate monthly flows.

PERIOD	LOW_10	LOW_5	MEAN	HIGH_5	HIGH_10
march	0.0009	0.001	0.0015	0.0023	0.0029
april	0.0028	0.0034	0.0051	0.0079	0.0096
may	0.0031	0.0042	0.0076	0.014	0.019
june	0.0029	0.0038	0.0067	0.013	0.017
july	0.0019	0.0028	0.0062	0.014	0.022
august	0.0014	0.0021	0.0046	0.0098	0.014
september	0.0017	0.0023	0.0043	0.008	0.011
october	0.0015	0.0019	0.0032	0.0069	0.0093

monthflow_b1250.vol - Used in conjunction with drainage area to calculate monthly flows.

PERIOD	LOW_10	LOW_5	MEAN	HIGH_5	HIGH_10
march	0.0009	0.001	0.0015	0.0023	0.0029
april	0.0028	0.0034	0.0051	0.0079	0.0096
may	0.0031	0.0042	0.0076	0.014	0.019
june	0.0029	0.0038	0.0067	0.013	0.017
july	0.0019	0.0028	0.0062	0.014	0.022
august	0.0014	0.0021	0.0046	0.0098	0.014
september	0.0017	0.0023	0.0043	0.008	0.011
october	0.0015	0.0019	0.0032	0.0069	0.0093

monthflow.remap – This table is used to remap the area below and above 1250 meters inside the drainage area for calculating monthly flows.

VALUE	CLASS
1250.0	1250.0
4500.0	4500.0

peak_flow.vol - Used in conjunction with drainage area to calculate instantaneous peak flows.

RETURN_YRS	BELOW_1250	F1250_1500	F1500
2.0	25.0	45.0	50.0
10.0	65.0	75.0	90.0
20.0	80.0	90.0	110.0
50.0	95.0	110.0	135.0
100.0	110.0	125.0	160.0

elevgreen.lut – Used as a symbol look-up for drawing the elevation grid classified by 50 meter intervals. The shadeset must be *colornames* to display the correct shades. This table is not used directly by the application.

Value	Symbol
600.0	77.0
700.0	62.0
800.0	61.0
900.0	64.0
1000.0	65.0
1100.0	68.0
1200.0	69.0
1300.0	73.0
1400.0	110.0
1500.0	108.0
1600.0	105.0
1700.0	101.0
4500.0	95.0

class_50m.remap - Used as a symbol look-up for drawing the elevation grid classified by 50 meter intervals. The shadeset must be *colornames* to display the correct shades. This table is not used directly by the application. This table is not illustrated due to its size.