US Army Corps
of Engineers
Construction Engineering
Research Laboratory

USACERL ADP Report N-92/01
April 1992
CADD/GIS Integration

GRASS-Intergraph Data Conversion Guide

by
Victoria Harmon
Douglas M. Youngs
Melissa A. Records

This guide outlines methods for bi-directionally translating vector data between Intergraph Graphics Design Software (IGDS) and the Geographic Resources Analysis Support System (GRASS).

GRASS is a workstation and a microcomputer-based image-processing and geographic information system (GIS) used to develop, manipulate, analyze, and display geographic datasets. Since both CADD and GRASS use digital spatial data (e.g., maps), this data translation capability can create joint applications between the two systems.

This report describes the necessary translation paths, data format organization, data editing, and procedures to translate data between the two systems.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
April 1992

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
GRASS-Intergraph Data Conversion Guide

5. FUNDING NUMBERS
OMA-WY-PD9
FAD 89-08037, Jan 1984

WY-JM9 MIPR
E8789L090, Oct 1989

WU-RH0-FAD 90-080586, Feb 1990

6. AUTHOR(S)
Victoria Harmon, Douglas M. Youngs, and Melissa A. Records

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Construction Engineering Research Laboratory (USACERL)
2902 Newmark Drive, PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
ADP N-92/01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

USA Engineering and Housing Support Center
ATTN: CEHSC-F
Bldg 358
Fort Belvoir, VA 22060-5516

Chief Installations Brach
ATTN: DAEN-ZCI-P
Rm 1E671 Pentagon
Washington DC 20310

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This guide outlines methods for bi-directionally translating vector data between Intergraph Graphics Design Software (IGDS) and the Geeographic Resources Analysis Support System (GRASS).

GRASS is a microcomputer-based image-processing and geographic information system (GIS) used to develop, manipulate, analyze, and display geographic datasets. Since both CADD and GRASS use digital spatial data (e.g., maps), this data translation capability can create joint applications between the two systems.

This report describes the necessary translation paths, data format organization, data editing, and procedures to translate data between the two systems.

14. SUBJECT TERMS
Geographic Resources Analysis Support System (GRASS)
Intergraph CADD vector data

15. NUMBER OF PAGES
60

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

FOREWORD

This study was done for the Office of the Chief of Engineers (OCE) and for the Engineering and Housing Support Center (EHSC), Fort Belvoir, VA. OCE funding was provided under the Operation and Maintenance, Army (OMA) project; Work Unit PD9, "CADD/GIS Integration"; Funding Acquisition Document (FAD) 89-08037, dated January 1989, and Work Unit JM9; Military Interdepartmental Purchase Request (MIPR) No. E8789L090, dated October 1989. EHSC funding was provided under Work Unit RH0; FAD 90-080586, dated February 1990. The OCE technical monitor was Mr. Stanley Shelton, ZCI-P, and the EHSC technical monitor was Mr. Donald Bandel, EHSC-FN.

This research was performed by the Environmental Division (EN) of the U.S. Army Construction Engineering Research Laboratory (USACERL). The principal investigators were Ms. Victoria Harmon and Mr. Douglas M. Youngs. Dr. Edward W. Novak is Acting Chief, USACERL-EN. The USACERL technical editor was Mr. William J. Wolfe, Information Management Office.

COL Daniel Waldo, Jr. is Commander and Director of USACERL, and Dr. L.R. Shaffer is Technical Director.

CONTENTS

CONTENTS (Cont'd)

TABLES

FIGURES

GRASS-INTERGRAPH DATA CONVERSION GUIDE

1 INTRODUCTION

Background

The Geographic Resources Analysis Support System (GRASS) is a geographic information system (GIS) composed of over 250 computer programs that capture, organize, process, analyze, model, and display digital geographic maps for monitoring and managing natural and manmade resources. GRASS is currently used at many U.S. Army Corps of Engineers Districts, military installations, labs, government agencies, and educational institutions.

In 1987, the Corps awarded a Computer Aided Drafting and Design (CADD) contract to Intergraph Corp.,* a leading developer and distributor of computer-aided drafting and mapping systems. Many Corps of Engineers Districts and military installations have acquired Intergraph hardware and software or are planning to acquire it in the near future. Common applications of Intergraph CADD software are facility planning, structural design, engineering mapping, and master planning.

Because the Intergraph CADD system and the GRASS geographic information system use digital spatial data (digital maps), there is a need to move data between the two systems to decrease the cost and time required to digitize maps. Data translation capabilities would create opportunities for job applications between the two systems.

Objective

The objective of this study is to present: (1) a guide for translating Intergraph CADD vector data into GRASS vector format and GRASS vector data into Intergraph vector format, and (2) CADD vector data documentation and digitizing requirements.

Approach

To enable Intergraph CADD and GRASS GIS to work together, USACERL initiated an effort to address (1) hardware compatibility, (2) data translation capabilities between the two systems, and (3) integrated and complementary applications. Hardware compatibility was attained in October 1989 by porting GRASS to the Intergraph Interpro series of 200, 300, 3000, and 6000 workstations. Data translation capabilities were addressed by evaluating vector-to-vector (line-to-line), vector-to-raster (line-to-cell), and raster-to-raster (cell-to-cell) conversions. Four translation paths were described as well as the two vector file organizations, and the procedures for running existing translation programs. Digital maps in the Intergraph Interactive Graphics Design Software (IGDS) vector format were acquired from Fort Sill, OK, and Fort McClellan, AL. Four existing translation paths were evaluated to move the digital data in two directions: from Intergraph to GRASS and from GRASS to Intergraph.

* Intergraph Corp., 1-T Madison Industrial Park, Huntsville, AL, 35801-4201, tel. 205/772-2000.

Scope

Information in this document is applicable to: GRASS 4.0 and future releases of GRASS; Intergraph VAX-based 1990 IGDS versions of *DLGOUT* translator and *DLGIN* translator; Intergraph Microstation and Microstation 32-based versions of *dxfout* and *dxfin* translators are also valid. The Intergraph Microstation GIS Translator (MGT), which contains the Microstation 32 version of *DLGOUT* and *DLGIN*, was not evaluated at the time of this printing.

Although this study addressed data translation capabilities by evaluating vector-to-vector, vector-to-raster, and raster-to-raster conversions, this document is concerned only with the vector to vector conversion. Applications incorporating both Intergraph data and GRASS data will be addressed in the future.

Mode of Technology Transfer

Information regarding distribution of all ports of GRASS can be obtained by contacting the GRASS Information Center, by phone: (800)-USA-CERL, X220 or (217)-373-7220; by U.S. mail: GRASS Information Center, USACERL, P.O. Box 9005, Champaign, IL, 61826-9005; or by electronic mail: grass@cerl.cecer.army.mil.

2 DIFFERENCES BETWEEN CADD AND GIS

Functional Differences

Both Computer Aided Drafting (CAD) and Geographic Information Systems (GIS) can capture, edit, display, and manage cartographic information. However, each system was designed for a different purpose. CAD systems were designed to automate the drafting function. Some of the chief strengths of CAD systems are the ease with which maps or drawings can be stored and retrieved, and the speed with which they can be updated, corrected, and otherwise modified. CAD capabilities that support the engineering design process resulted in the technology commonly referred to as "Computer Aided Drafting and Design" (CADD). CADD does incorporate some modeling of relationships between graphic components. For example, CADD can be used to analyze the design performance of electrical, water, or storm drainage distribution systems. CADD-based capabilities are particularly well suited to special applications such as managing facility information, or performing engineering analysis on utility or structural components. However, CADD offers limited analytical and spatial decisionmaking support for applications involving environmental or natural resource planning and management. The need for such capabilities has spurred the development of GIS, a computer graphics technology founded on the principles of spatial analysis.

GISs emphasize map transformation, analysis, and geographic process modeling. GISs allow for the merging, aggregation, generalization, and association of mapped data. Some common uses of GISs are resource management, and urban and economic planning. For example, one GIS application creates a map describing soil erosion potential. The output map is derived from input maps drawn from various types of data: elevation, slope, soil type, land cover, crop management practices, and rainfall. The accuracy of this kind of map analysis depends on digitizing precision and the quality of the data sources.

Although CADD-based automated mapping and facility management capabilities are sometimes touted as geographic information processing, they lack the true spatial analysis functionality of GIS technology. CADD systems are fundamentally concerned with the display and manipulation of graphic material, whereas GISs are concerned with data relationships, spatial modeling, and analysis.

The Intergraph CADD Mapping Database

Thematic Map Types

Intergraph CADD files are stored and displayed in a vector (linear) format. All graphic objects are composed of lines, arcs, circles, etc. that are described using x and y coordinate positions. CADD thematic map types that may be useful in a GIS are elevation contours, planimetric data (e.g., roads, buildings, streams), and utilities (Table 1). Other CADD data themes such as sidewalks, electrical circuitry, etc. may not be as useful in a typical GRASS GIS analysis. Generally, three categories of data files are produced: Planimetric, Contour, and Utility. (Others may exist, e.g., future plans.) Each file can contain a maximum of 63 "levels" to organize data. A planimetric file will usually contain the most diverse range of thematic data, each theme having its own dedicated level(s).

Intergraph Data Sources

Intergraph mapping databases created for comprehensive planning at many military bases are usually created via photogrammetry. Aerial photographs taken at controlled altitude capture ground details. Altitude is one determining factor in the resultant scale accuracy of hardcopy maps. Using ground-truth

Table 1

Vector Data Themes

Map type	GRASS	IDGS CADD
Contours	Used to derive raster elevation model 100/500-yr flood limits	(5, 10, 25-ft*) spot elevations 100/500-yr flood limits Text (100 & 400 scale)
Planimetric	Boundaries, survey monuments roads, buildings, land cover Land uses & management Safety zones	Boundaries, survey monuments, roads, sidewalks, buildings, streams, trees Misc. ground features, i.e., golf course, helipads Safety zones Text (100 & 400 scale)
Utility systems	Distribution and points (i.e. poles)	Valves, hydrants, poles, etc. Text (100 & 400 scale)
Natural resources	Streams & watersheds Soils Wetlands Wildlife habitats	Areas, points, lines
Cultural resources	Areas, points, buildings	Areas, points, buildings
Predictive models	Erosion	Not derived from analysis
Future plans	Areas and points	Text (100 & 400 scale)

*1 ft = 0.305 m; 1 in = 25.4 mm.

surveying, these aerial photographs are placed and calibrated in a device called an analytical stereoplotter. An operator then views these photographs in 3D stereo, and digitizes themes into dedicated levels by tracing thematic lines. Contour maps can be created because the operator can follow a line of equal elevation throughout the photographs due to the stereo effect, which has been referenced to a ground truth survey. Below-ground utility maps are usually created by digitizing paper construction documents.

Intergraph Data Scales

The most common target scales for Intergraph mapping databases has been 1:1200 (1 in. = 100 ft) and 1:4800 (1 in. = 400 ft). A scale of 1:1200 is used to produce maps of urban areas and 1:4800 is used for larger geographic coverages. The 100-scale maps are often used to locate underground utility pipes, to avoid accidental pipe disruption.

Intergraph Coordinate Systems

Intergraph mapping databases are usually created using the State Plane coordinate system. State Plane data must be transformed for use in another coordinate system, such as Universal Transverse Mercator (UTM) or Latitude-Longitude.

Intergraph Vector File Structure

The Intergraph vector file structure addressed by this report is the IGDS. This is the most common vector format used by Intergraph (Table 2). IGDS files contain 63 levels on which data can be digitized and stored. This file structure aids graphic organization while digitizing complex maps and designs. The user may place different graphic elements and themes on various levels. For example, roads may be placed on one level within an IGDS file, road center lines on another level, sidewalks on a third level, and text on a fourth level. Intergraph IGDS files are called "design files" and have the extension ".DGN". The internal file structure of IGDS files is licensed by Intergraph Corp.

Intergraph File Organization

Intergraph mapping databases are commonly divided into a grid pattern of sections called "facets," which geographically delineate the contents and the size of a database map file. This is done to control the sheer volume of mapped detail inherent in facility-oriented mapping efforts. An entire military installation is comprised of many facets that must be attached to view the entire area. An example facetization scheme for the Chanute Air Force Base database is shown in Figure 1. The C2 facet would have at least three associated individual files with dedicated levels reserved for planimetric, utility, and contour information. If the planimetric, utility, and contour files for the C2 facet were merged, each data theme in the resultant file should have its own dedicated level.

There are usually two sizes of facets corresponding to two target hardcopy map scales. Installation cantonment areas are usually mapped at a scale of 1:1200 and the whole installation is usually mapped at 1:4800. The facets for the 1:1200 scale data are generally 2500 ft north-south by 3000 ft east-west. Facets for the 1:4800 scale data are generally 10,000 ft north-south by 12,000 ft east-west.

Attributes may be associated with the graphic elements in an Intergraph IGDS file. If attributes are present, they are stored in a separate attribute database. The attribute database for VAX-based IGDS software is called Data Management and Retrieval System (DMRS). The attribute databases for Intergraph Microstation based products are the relational database management systems Informix, Oracle, or Ingres.

Table 2

The Intergraph Vector Format

Software	Vector Format
VAX-based software	IGDS
Workstation-based Microstation 32	IGDS
PC-based Microstation	IGDS
Workstation-based Microstation GIS	IGDS

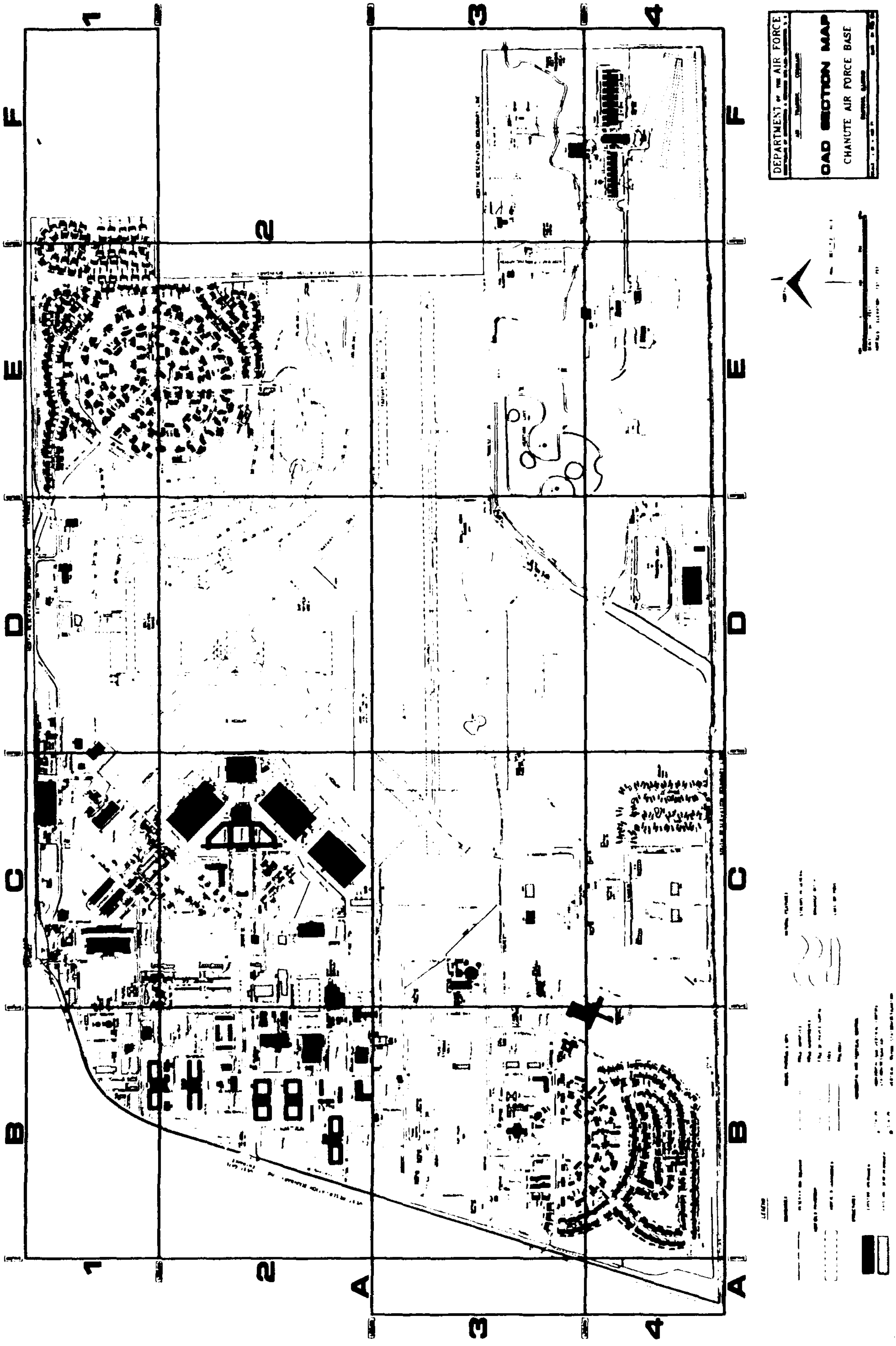

Figure 1. Chanute AFB CAD section map.

The Grass GIS Database

Thematic Map Types

Typical GIS map types are of roads, hydrography, soils, geology, landcover, elevation, slope, aspect, land use, watersheds, wetlands, wildlife habitats, satellite imagery, and almost any type of map useful for environmental, economic, or land resource monitoring (Table 1). The diversity of small-scale environmentally related maps is usually greater in a GIS than in a CADD system. The GRASS GIS contains both a vector file *(dig)* for a map and a raster file *(cell)*. Associated with each vector file is a vector attribute file *(dig_att)*, a vector topology file *(dig_plus)*, and a vector category file *(dig_cats)*.[1] Associated with each raster file is a header file *(cellhd)*, a histogram range file *(cell_misc)*, a category file *(cats)*, a raster color file *(colr)*, and a history file *(hist)*. For GRASS release 4.0, vector files are used for display and reference; raster files are used for display, reference, and map analysis. All GRASS vector files can be converted to GRASS raster files using the GRASS program *v.to.rast*.[2]

Although this report deals only with vector to vector translation, at this point it is instructive to explain the difference between the vector and raster graphic data formats. The vector format describes graphic positions by storing the x,y coordinate pairs that define a line or arc. Collections of coordinate pairs make up a line segment or arc segment, and segments eventually make up linear or polygonal features. The raster format describes graphic position by assigning integer values to the individual grid cells of a two-dimensional matrix overlaid on the study area. If the data theme is a linear feature, such as roads, and there is only one category of road (single lane, paved), then the grid cells are assigned a value of either 1 for the presence of the road or zero for the absence of the road. Examples of more complex maps in raster format are soils maps and satellite images.

GRASS Data Sources

GIS maps are obtained from a wide array of sources. The selection of map source depends on map availability, project application scale, and application focus. If the application requires large scale elevation data, sources supplying large scale data will be sought.[3] If large scale elevation data does not already exist, it may have to be created using existing aerial photographs, or by scheduling an air photo mission.

If the best vector linear or vector areal data can be acquired by digitizing mylar map separates, or by having maps scanned electromechanically, this approach is often taken.[4] If data already exists in CADD vector format, data translation may be the most cost effective option.

GRASS Data Scales

The most common map scale used by GRASS is 1:24000. Many digital map layers in a GIS are the result of digitizing 1:24000 U.S. Geological Survey (USGS) quadrangles. Other scales used depend on the focus of the project or the application. GRASS can use global data with a scale as small as 1 to 5 degrees as well as maps having a scale as large as 1:1200 or larger. Scale is not necessarily limited.

---

[1] Michael Shapiro, et al., *GRASS 3.0 Programmer's Manual*, Automatic Data Processing (ADP) Report N-89/14 (U.S. Army Construction Engineering Research Laboratory [USACERL], September 1989).

[2] James Westervelt, et al., *GRASS Reference Manual*, N-87/22 (USACERL, September 1988).

[3] Stuart Bradshaw and Pam Thompson, *Options for Acquiring Elevation Data*, Technical Manuscript (TM) N-89/20/ADA2200934 (USACERL, January 1989).

[4] Jean Messersmith, "Map Preparation," in James Westervelt, et al., *GRASS Reference Manual*.

However, the 1:24000 scale is used most often because it best represents features on the ground at a scale that covers an intermediate to large geographic area.

GRASS Coordinate Systems

The coordinate systems used by GRASS include the UTM, State Plane, Latitude/Longitude, and Cartesian Coordinate system.

GRASS Vector File Structure

There are six GRASS vector files: the binary vector file *(dig)*, the optional American Standard Code for Information Interchange (ASCII) vector file (*dig_ascii* —this can be created from the binary vector file using the GRASS program *v.out.ascii*) the topology file (*dig_plus*), the attribute file (*dig_att*), the optional category file (*dig_cats*), and the *reg* file, which contains digitizing registration points.[5]

The *dig* and *dig_ascii* files are flat files (no levels) containing header information, nonintersecting curves called arcs, intersection points called nodes, and corresponding x,y coordinate pairs. Arcs can be designated as lines (linear features) or areas (polygonal features). Each line or area is assigned a single integer attribute value called a category number, which is stored in the *dig_att* file.

GRASS File Organization

GRASS vector maps are usually digitized as one file or map, covering the entire geographic area of interest. If vector data must be digitized in segments, it is patched together after digitizing, or digitized into one UNIX vector file. GRASS vector attributes are stored in an associated vector attribute file *dig_att*.

It should be noted that Intergraph map databases are partitioned (facetized) to facilitate the production of 100- and 400-scale paper maps specified in typical mapping scopes of work for military installations. The decision to partition has nothing to do with any limitation of the Intergraph CADD software. Complete coverage can be acquired for the entire geographic area of interest.

[5] For a thorough description of the GRASS vector file structure, refer to the *GRASS 4.0 Programmers Manual*, Chapter 6.

3 THE HARDWARE ENVIRONMENT

GRASS development has been accomplished on a variety of UNIX-based machines to enhance software portability. At the time of this writing, GRASS is officially supported on the following platforms:[6]

- Masscomp mc6300
- Sun Sparcstation 1, Sun Sparcstation 2, Sparcstation IPC
- Sun 386i
- AT&T 6386
- AT&T 3b2
- PC 386 (Compaq and Dell)
- Silicon Graphics IRIS 4D/20
- Apple Mac II
- Tektronix XD88
- NCR 3345
- Intergraph Interpro Series models 240, 2020, 3050, 6000, 6040
- IBM System 6000/Model 320 Configuration
- Data General Avion 300c.

Users may want to perform translation on various hardware configurations. This chapter offers a generic example of a hardware environment made up of a mixture of Intergraph, PC, and GRASS workstation platforms, illustrated in Figure 2.

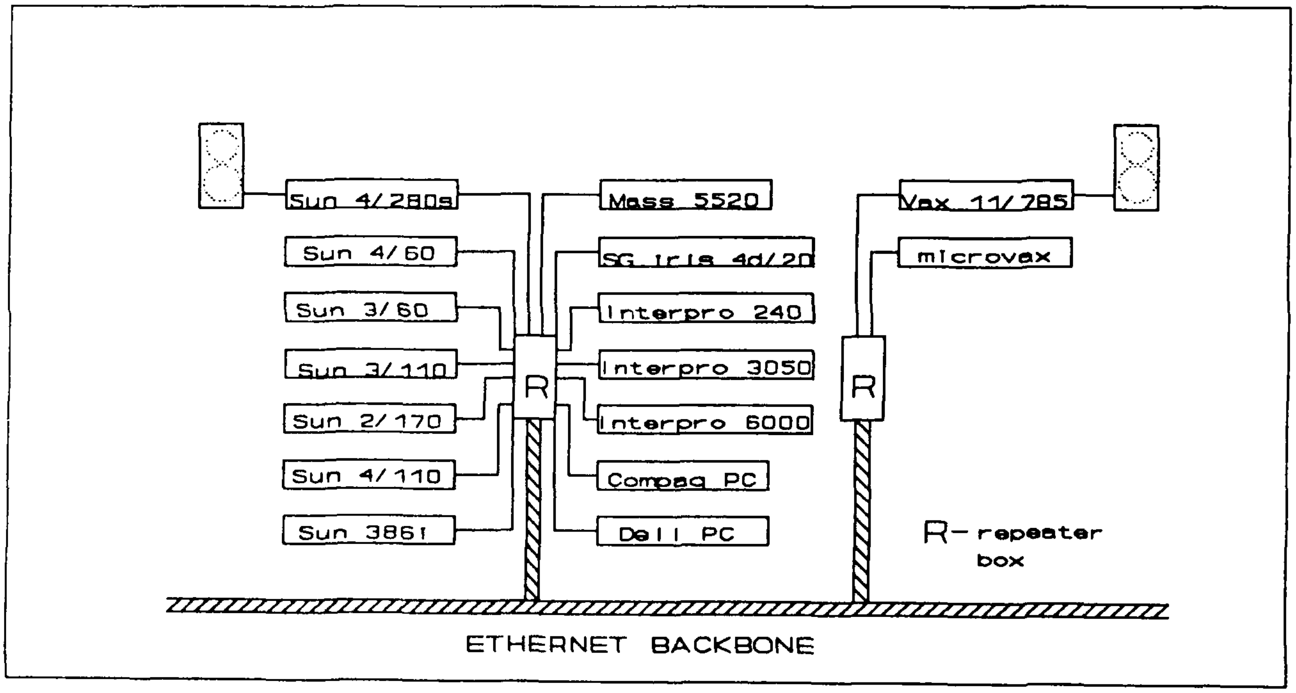

Figure 2. Schematic of a Mixed Hardware Environment.

[6] More specific configuration details may be found in: Douglas Brooks, Michael Shapiro, and Mark Johnson, *GRASS Hardware Configuration Guide*, ADP N-89/21/ADA220954 (USACERL, March 1989).

All Sun workstations use the Sun 4.1 operating system, which is characteristic of the Berkley version of UNIX. All other workstations (listed above) use operating systems more characteristic of AT&T System V UNIX. Interpro workstations use a System V version called CLIX. The VAX 11/785 and microvax hosts use the VMS operating system. The PC 386 workstations use System V UNIX operating system called Dell UNIX or Interactive, with Etherlink cards connecting to the ethernet. PC hard disks commonly can be partitioned with both DOS and Unix partitions (i.e., a 400 Mb UNIX and 100 Mb DOS partition, allowing the DOS-based microstation to still be operated). PC 386 configurations often have additional software packaged with the operating system to move data between the DOS and UNIX partitions, allowing data generated in the DOS partition to be transferred out to the network.

With the exception of the VAX machines, all other workstations run Network File System (NFS) software, which comes with the purchase of the Sun Operating System. The Sun 4/280 functions as a file server. NFS software from Intergraph is used to connect intergraph workstations to the rest of the existing GRASS workstations.

IGDS data is binary compatible between the VAXs, Interpros, and PC-based IGDS products. Data has been successfully downloaded from magnetic tape to the VAX 11/785, transferred across the network and from UNIX to the DOS partition to be edited in Microstation PC. However, some data compatibility problems have been encountered in transferring GRASS binary vector data between the Sun and Interpro environments. This may be due to byte-swapping differences between the Sun and Interpro machines. In these cases, data is simply transferred in ASCII format to the target workstation, where it is then converted to binary form.

4 ASPECTS OF IGDS MAPPING DATABASES THAT AFFECT TRANSLATION

Digital Integrity

Often, attention has not been paid to the delivery of a digitally "clean" database product suitable for translation into a topologic GIS format. Emphasis has been placed on the production of a paper product with traditional paper-based graphics standards. Therefore, digital files contain many overshoots, undershoots, and discontinuous and duplicate lines.

For instance, contour files have discontinuous contour lines, which require extensive editing and "best guesses" to connect into continuous form. Editing may be so extensive as to prohibit conversion of the data. Contour interval text labels are inserted in the breaks of the discontinuous contour lines. Many discontinuous lines exist where sharp changes in topographic relief would result in close, tightly fitted contour line representations. Depressions are represented with special graphics symbols that must be edited. Again, this results from an emphasis on a paper map product.

Global Origin and Coordinate Representation

The global origin and coordinate representation should be set up so that the associated coordinates of all ground features depicted in the database identify the actual State Plane coordinate position of features. Some databases have been delivered with data elements that do not match actual ground coordinates, or with coordinates in a truncated, abbreviated form.

Design File Organization

Files must be delivered complete and in clear order, with proper facetization. The database requires a consistent facetization scheme or complete geographic thematic coverage. Data themes must be located on correct levels, with consistent feature color coding, linestyles, and lineweights. Data themes (e.g., sidewalks, roads, fences) must be kept consistently on their designated levels. Facets should cover all important geographical features, such as installation boundaries.

Database Documentation

Databases should be delivered in a consistent and organized manner; delivery of a 9-track tape and file listings of contents is insufficient. Databases should be delivered with a documentation report, if possible. Names of database files should indicate contents. File types should be separated and data source or scale technical details should be included. Refer to Chapter 15 (p 44) for documentation guidelines on the creation of multi-use data.

Visibility of Data Elements Depicting Map Features

Some data elements, such as Complex Patterning, are only visible when specific display switches are set. Enable the pattern display and observe the graphics display to evaluate existing data which might not be visible when pattern display is off.

Graphics Standards

The graphic symbolism of some features (such as topographic depressions) will have no graphic counterpart in GRASS, where topography is described via an elevation model. Such features must be edited, either in the CADD system or GIS. Care must be taken if it is desirable to translate Microstation CADD symbols called "cells" (telephone poles, transformers, etc.). Cells may have to be decomposed into individual graphic elements before translation. The dxf import program distributed with GRASS will only recognize points, lines, polylines and text; other types of data are ignored. Decomposing cells using the Microstation DROP COMPLEX STATUS COMMAND will increase file size and require editing labor. A cell symbol, such as a telephone pole may become decomposed into a circle filled with many individual line segments. Cell symbols may be exchanged globally for simpler representations which do not require editing. To do this, refer to the CELL REPLACER UTILITY (rcl.exe) which comes bundled with Microstation.

Attached Reference Files and File Compression

Attachment of too many reference files can obscure actual file contents and require labor to detach. Files are often not compressed, ballooning the size of the database and slowing network transmission. However, IGDS data compression sometimes results in file corruption. It is advisable to "back up" before compressing IGDS files.

Data Redundancy and File Size

Titleblocks and legends are often found in every file. Database files with 100- and 400-scale contain duplicate information. Often duplicate data elements (i.e., contour lines) are found and copied in an overlapping fashion at the same coordinate location. Contour file sizes are very large, even on a per-facet basis. Record increment digitizing thresholds may be set to capture more data points than necessary to describe topography. Under these conditions, merging all contour facets together will create a very large and unwieldy file for transport over a network and conversion to raster DEM format.

5 UNDERSTANDING THE IGDS MAPPING DATABASE

Before IGDS files are translated into GRASS, it is first necessary to assess the quality of the graphic data and identify the presence of factors that will affect translation. **To make an assessment and prepare the data, there is no alternative to learning the Intergraph CADD system.** The documentation, which should accompany the database (Chapter 16, "Documentation Needed for IGDS Data Translations," p 49-50), serves as a general guide to conditions that should be found in the database. The individual performing the translation must check the database by sampling the files to verify consistent compliance with the documentation and to note anomalies. Digitizing errors and exceptions to documented information may exist. Targeting a data theme that is not overly complex is a good way to begin the process and learn translation pitfalls. It also will be necessary to gather information needed to run the translation programs. Chapters 7 to 11 (pp 25-39) list and describe the translation programs. An analysis of IGDS data will determine the condition of the database and the labor involved in performing conversion. This will help determine the translation program and path to use when choosing between the *dlg* and *dxf* alternatives described in this report.

IGDS Hardware Environments

There are three Intergraph hardware environments for an individual performing translation to display, query, and edit IGDS data: the *VMS* VAX IGDS environment; the *UNIX* computer workstation environment using Microstation 32 software; and the *DOS* PC environment using Intergraph Microstation software. These three systems offer similar display and query capabilities, but the commands and procedures differ slightly from system to system. This section, describes the query procedures in generic Intergraph terms instead of specifying step-by-step commands.

Some IGDS Analysis and Editing Tools

IGDS Graphics Display Environment

Within the IGDS graphics display environment, the user may control data visibility by turning on and off design file levels to determine what thematic components are on each level. This is the most appropriate method for initial evaluation of data. Selection of one of the element manipulation commands, such as the *change color* command, can also be used to echo the design file level of a particular graphic element to the screen. After selecting the element manipulation command and identifying the graphic element with the cursor, the data level will be displayed. Choose the *reject* option to abort the program so an element manipulation will not actually occur.

The supplement library can also be accessed to display tabular information about a particular graphic on a virtual screen. A command that displays this information in the VAX environment is: *uc=pro_dd_sup:anl.*[7] This command displays the data level, element type, line style, line color, and other useful information.

---

[7] *Intergraph Microstation Reference Guide*, DSA027110 (Intergraph Corp., 1989).

Edit Display Graphics

You may also analyze and edit data outside the graphics environment using the Edit Display Graphics (EDG) utility. EDG is the Intergraph Edit Graphics Utility (refer to *EDG User's Guide*, DSYS103 (1987), and *Microstation Reference Guide*). EDG is an extremely useful, powerful program that allows one to edit graphics outside the display environment. EDG should be used with caution since commands can globally alter data. (Be sure to make backups.) EDG offers an automation alternative for target data found to be in excellent condition (via evaluation by graphics display). In addition to its function as an editing tool, EDG can supply the user with ASCII (tabular) information about design files, which can be stored in a readable file and printed for later use. Among EDG's querying options is one that provides general information about a design file, including a list of element types, data levels, line weights, line styles, line colors, fonts, number of active elements, number of deleted elements, and the total number of elements. The level location of the design file header, as well as whether the file is 2D or 3D, is also included.

A full listing of all of the elements in a design file can be requested; however, be aware that printouts of this type of information for normal to large-scale files may be long (more than 100 pages). A specific list of element symbology for one or more element types, or a list of the elements and the symbology for a particular level can also be requested. EDG procedures and options vary with the IGDS software and hardware environment.

Organization and Preliminary Data Analysis

After thoroughly evaluating the documentation accompanying the database, copy the data from the storage medium to the computer's disk using standard input, and organize the data files according to filetype: planimetric, contour, or utility. Make sure that all database files for the coverage are present. It is a vitally important requirement throughout the translation process to maintain a high degree of organization and detailed bookkeeping. Begin an analysis of a simple target data theme of interest, such as boundaries. Methodically sample each file, making sure it will load, while checking the following important characteristics.

Data Levels, Element Type, and Symbology

Intergraph IGDS design files contain 63 levels on which data may be digitized. All 63 levels are available in each IGDS file regardless of whether they are empty or contain data. IGDS levels may contain nondisplayable header data, graphic data, or displayable text. There are many graphic element types used to construct the graphics that describe map themes, e.g., *"line," "linestring," "curve," "connected string," "ellipse,"* etc. (refer to *EDG User's Guide*, DSYS103). Element symbology refers to linestyle (dashed, solid, etc.) linecolor, and lineweight (narrow or wide).

The thematic category of graphic data that resides on each level is important. For example, thematic categories representing a road map might be primary roads, secondary roads, and tertiary roads. The thematic categories of graphic elements representing topography might be index contours, intermediate contours, and spot elevations. Index contours might reside on level 50, intermediate contours on level 51, spot elevations on level 52, and contour text labels on level 55.

To run the two IGDS-to-GRASS translation programs described in this report, knowledge of which design file levels contain graphic data and textual data is required. (Refer to your database documentation.) However, you must check to verify that the theme of interest is consistently on the correct level. At the same time you can zoom in on areas to evaluate the condition of the graphics (i.e., overlaps, undershoots, etc.). Also check to verify that there are no *reference* file attachments to confuse your visual analyses. In addition, you should set the display visibility switches to show complex patterns that may have been used to represent railroads or an installation border. The color of graphic elements may change depending on the graphic display capabilities of your platform. Data may display with the color, black, which will not be visible. There may be duplicate lines that must be removed before translation.

Global Origin

The global origin is set up prior to the digitizing process and must be retained to reflect the data's coordinate system. For example, the global origin for the Fort Sill database is go=-1700000,-300000. These negative values indicate that the graphic elements in the database will not be truly georeferenced in the State Plane coordinate system, and that the global origin must be moved. When this global origin is correct in the design file display environment, the coordinate values of graphic elements, when queried, will echo the correct ground coordinates to the screen (in this case, State Plane). To check the global origin within the display environment, in either VAX-IGDS, Microstation 32, or PC Microstation, type: *go=$*. If the global origin has not been changed, the correct global origin will be displayed. The digitizing contractor can provide the correct globai origin in which the data were digitized. This value should be retained in the seed file used to construct the database.

Coordinate System

It is important to check the coordinates of the design file in the IGDS display environment to confirm that they are accurate (i.e., State Plane, UTM, etc.). After checking the global origin, the coordinates can be queried by selecting the *key point snaplock* option in VAX-IGDS, Microstation 32, or PC Microstation, and then using the tentative point (T) button on the nine-button cursor. When the cross-hair pointer snaps to the nearest graphic point, the coordinates of that point will be displayed on the screen. If the correct coordinates are not displayed, the global origin may not be set to the correct x,y values. You can correct the misregistration of a file's coordinate system if you know the true ground position of an element in the database (preferably a survey comment). Input this coordinate to the GLOBAL ORIGIN command and tag the crosshair target on the known monument to re-register the coordinate syste. Diagrams of survey monuments with corresponding state plane coordinates are generally available from Engineering Divisions at installations. However, one should be cautious, as sometimes installations will devise their own unique coordinate system.

File Size

Note that the IGDS file sizes may be deceiving. IGDS files retain marked deleted graphic elements. Only when a file is compressed are these deleted elements actually removed from the file. Files need not be compressed, unless file sizes are unwieldy. IGDS data compression sometimes yield file corruptions; backups are required. EDG repairs on these file corruptions are varied in complexity. File size is an important factor in data translation because some translation programs temporarily increase file size (*dxfout*).

6 TARGETING AND EDITING IGDS DATA

Targeting Data

It is necessary to evaluate the source of any data to justify conversion. Some data themes may have been digitized from out-of-date paper sources. Data themes should be prioritized for conversion. It may be useful to refer to existing map products produced from the database to prioritize data themes. Note that, at the time of this writing, the program to convert contour line data to elevation model raster format (*r.surf.contour*) is computationally intensive. Preliminary results from *r.surf.idw2* have greatly reduced processing time. It is best to evaluate the contour data conversion path and results before doing extensive editing.

Editing IGDS Data

Following the analysis of IGDS data, some aspects of the design file may need to be edited (changed) to facilitate translation into GRASS (refer to Chapter 4, "Aspects of IGDS Mapping Databases that Affect Translation," p 17). Before editing a file, it is a good idea to make a binary copy in case mistakes are made. IGDS target data for conversion should be stripped and isolated out of the original file into a new (smaller) file containing only the data level(s) of interest. This can be done by using EDG or graphically, by controlling level visibility, and Fence Filing the data level(s). After level stripping, be sure to note the resultant file size of the target data using the appropriate command of the supporting operating system: DOS, VMS, or UNIX. In the level-stripping procedure, no internally deleted graphic elements are transferred to the new file; in effect, this performs an automatic compress. File size will determine whether files must be translated individually (facet by facet), or whether files can be merged prior to translation. Merging facets creates larger files, but can simplify the conversion from IGDS to *dig*, which might otherwise take 10 or 20 individual conversions and patches, sometimes to a single step. One must always consider when to make file backups. You must balance data being worked on so as not to bottleneck the translation process by not having enough storage for processing. It is best to backup data at important stages of the preparation and conversion process. When to backup is usually determined by evaluating how much work has been invested in specific file(s). Other possible manipulations and graphic changes to IGDS data before translation into GRASS are:

1. Search for missing data from a level by turning levels on and off and perhaps displaying adjacent files as references. Use display controls to magnify data, to pan over data, or to create multiple viewports.

2. Change the color of data or symbology to enhance understanding. Change the element symbology of a group of graphics on a design file level to give the graphics a separate feature code when running *DLGOUT*. *DLGOUT* uses element symbology to distinguish graphics when assigning graphics a feature code (FC=category). Refer to Chapter 9, "How to Run Intergraph VAX *DLGOUT*" (p 29). *DLGOUT* changes all IGDS styles to the solid style during translation.

3. Move data to a different level in either the graphics display environment or EDG.

4. Move data on a level(s) to a new file using either the graphics display environment or EDG. This allows the user to translate a file containing only one or two levels without having to translate the data on the unwanted levels. The levels are separated out of the *dxf* file when it is imported into GRASS during the GRASS import program *v.in.dxf*. The program *v.in.dxf* creates a GRASS vector file for each

level in the *dxf* file. At this point, unwanted vector files can be removed, but translating unwanted levels results in many *dxf* files that take up storage and may be unnecessary.

5. Change an element type (e.g., complex to primary) curve string to a line string, cell to individual elements.

6. Delete data, such as duplicate linework, or graphics standards for topologic integrity. Join broken lines, trim line overshoots, or correct crossing contour lines.

7. Change display visibility parameters to show patterned elements, such as railroads or perhaps installation borders.

8. Detach reference files. It may be useful to attach reference files for analysis and editing.

9. Compress a file.

10. Change the global origin to move data to correct coordinate location.

11. Strip out data from one level to a new level or file using EDG to differentiate data by color, linestyle, lineweight, symbology, or element type. If two types of data reside on the same level, such as roads and streams, you may want to move the streams to a separate level, or to a separate design file. Both of these solutions will enable the two thematic types to be translated into two separate GRASS vector files. Refer to Chapter 20, "Exporting *DLG* and *DXF* Files From GRASS" (p 51)

12. Repair a corrupt file using EDG.

13. Merge two (or more) files; split a file into two (or more) pieces.

14. Change the color of a group of graphics to distinguish the group from another group of graphics on the same design file level. This would be applicable only if *DLGOUT* was to be run on the data after editing. *DLGOUT* is the Intergraph program that translates IGDS files to USGS Digital Line Graph (*DLG*) files with the option to assign the line color, line weight, or line style on an IGDS level to a *DLG* feature code (FC=category). Refer to "When To Use Intergraph *DLGOUT*" (p 26) and "How To Run Intergraph VAX DLGOUT" (p 29).

All of these changes can be made within the IGDS display environment. Graphic editing may also be accomplished using (see Edit Display Graphics (p 20). Editing in the display environment rather than EDG, however, is safer for new users since major mistakes are easier to make using EDG. EDG is an excellent tool to locate and identify data and data levels.

Editing GRASS Data

Sometimes data must be edited after translation from IGDS to GRASS. Some examples are:

1. The joining of discontinuous contour lines that were broken to place elevation text labels. Locating the text spaces in contours is useful after translation to help identify where the text label boxes for the contours, created by the GRASS import program *v.in.dxf*, should be when the box vector file is overlaid in the GRASS program digit. Occasionally there may be text (boxes) in a contour IGDS file identifying features other than the index contours. These text boxes are needed to check for mislabeled

contours within GRASS digit (refer to "v.cadlabel" [p 42] for further explanation regarding the creation of boxes around IGDS text data).

2. Screen digitizing of small line breaks is made easier in the GRASS digit program because the breaks are automatically highlighted with colored nodes. However, this advantage must be weighed against program performance with large volumes of data. Microstation handles large volumes of data more quickly during in-house tests. In addition, Microstation has powerful panning and multiple viewport display capabilities. Both programs have adjustable snapping thresholds; however, the Microstation program has an *undo* feature that tolerates erasure mistakes, whereas the digit program does not.

Automation Possibilities

Automation of the IGDS editing process is helped by a well-organized database that strictly adheres to the level structuring of data themes. If the level structuring of data themes varies from file to file (facet to facet), then individual modifications must be repeated for many or all of the facets (IGDS files) in the database. For example, a modification might be necessary when moving levels 50 and 55 for all files in the database to a single new file prior to translation. Automation will not be successful if many of the files have level 50 data themes on the wrong level. Automation can decrease translation time, file size, and aid in troubleshooting translation problems. When facets are translated separately, for example, primary roads on level 50 and secondary roads on level 55, the *level stripping* process would have to be repeated for each facet. Automation or *bulk editing* allows the user to run the editing sequence once and, in this case, strip levels 50 and 55 from any number of files without operator attention. This kind of automation is accomplished using editing software that runs outside the Intergraph graphics interface. The Edit Graphics Utility (refer to "Editing IGDS Data," p 23) provides this capability. A program called *Programmable EDG* is available to automate the editing of files.[8]

Check Plots

Hardcopy paper plots may be useful as references for targeting and editing IGDS data for conversion. These can be produced from the IGDS database if one has access to a pen plotter. Another approach would be to obtain blueprints of the map products derived from the IGDS mapping database. It can be difficult to maintain a perspective when one is zooming in and out of a complex file that takes several minutes to replot on the graphics screen. Referring to a check plot can be a useful "road map" to help you keep focused on the data of interest and to keep tabs on where you are in the editing process. After conversion to GRASS, another vector plot may be created to verify an accurate translation.

[8] *Programmable EDG* is distributed by Axiom Software, P.O. Box 210655, San Francisco, CA 94121-0655, tel. 415/751-8404.

7 INTERGRAPH TO GRASS VECTOR TRANSLATION PATHS

There are presently two possible vector translation paths from Intergraph's IGDS vector format to the GRASS vector *dig* format. One path uses the USGS *DLG* format as an intermediary format, and the other path uses the CAD *DXF* format as an intermediary format. The *DXF* or *DLG* file is then imported into GRASS using GRASS capabilities. Figure 3 shows the two translation paths.

```
IGDS ---------------------> DLG ----------------> GRASS vect
            (DLGOUT)              (v.import)

IGDS ---------------------> DXF ----------------> GRASS vect
            (dxfout)              (v.in.dxf)
```

Figure 3. Intergraph-to-GRASS Vector Translation Paths.

The current GRASS vector import capabilities are:

1. DLG-3 Optional Format
2. DXF
3. TTD (experimental DMA data)
4. ADRG (experimental DMA data).

The software programs that were evaluated are the two Intergraph IGDS export programs *DLGOUT* and *dxfout* and the two GRASS import programs *v.import* and *v.in.dxf*. *DLGOUT* runs on the Intergraph VAX platform only, and *dxfout* is an Intergraph Microstation 32 and PC Microstation-based product. The program *dxfout* is distributed with and comes packaged in the Microstation software. The two GRASS import programs, *v.import* and *v.in.dxf*, are executed within GRASS, and therefore run on all of the GRASS hardware platforms, such as the Masscomp, Sun, and Intergraph Interpro workstations.[9]

[9] See the *GRASS Hardware Configuration Guide*.

8 CHOOSING *DLG* OR *DXF* AS AN INTERMEDIARY VECTOR FORMAT

Both translation paths have produced successful results. The discussion that follows assumes you have the two alternative translation paths at your disposal. The additional cost of purchasing *DLGOUT* may not be warranted, especially when *dxfout* comes packaged with Microstation 32 and Microstation PC. *DLGOUT* offers advantages to databases prepared with its use in mind, i.e., when the data is clean and the *DLGOUT* feature coding capabilities can be used. However, most of the mapping databases we have evaluated have been produced with a paper map product in mind, not with *DLGOUT*. *dxfout* is easy to run and simplifies the complexity of translating different IGDS element types into lines and polylines. A translation parameter file exists that may be used to screen cell symbols, to eliminate them from the translation process. File size and contour labeling issues can also be considered when choosing between methods. These factors and the basic differences between *DLGOUT* and *dxfout*, as well as some important aspects of *v.in.dxf* that relate to translation decisions, will be covered in this chapter.

When To Use Intergraph *DLGOUT*

DLGOUT is the Intergraph program that converts Intergraph IGDS vector format to USGS Digital Line Graph format.[10] The difference between this program and the Intergraph dxfout translator is that element symbology in the IGDS file can be assigned a *DLG* major and minor feature code in *DLGOUT*, and then when the *DLG* file is converted to *dig*, the *dig* file will have category labels. Each category label in the *dig* file will have the same label number as the label number (feature code) assigned to the lineweight, linestyle, or linecolor in the IGDS file during *DLGOUT*. All IGDS linestyles (dashed and dotted lines, etc.) will become solid lines in *DLG* format and therefore in GRASS dig format.

Another advantage of *DLGOUT* is that it does not increase the intermediate vector file size quite as much as *dxfout*, although both translators may increase the file size (Table 3). The final GRASS dig file may be smaller or larger than the IGDS file, but is usually significantly smaller than the intermediate vector file.

There are two Intergraph versions of *DLGOUT*. One resides on the VAX and the other is packaged in Microstation GIS Translator (MGT), which is a modular part of Microstation GIS. The VAX version, at the time of this printing, is not able to retain the elevation values internally tagged to the IGDS contour lines, if the contour lines were digitized as Intergraph curve elements. During *DLGOUT*, the program "strokes" curve elements, converting them to Intergraph line elements, and in the process does not save the tagged elevation values. Null elevation values are put in their place. When *DLGOUT* encounters a null value, it quits. A logical solution to this problem would be to run the Intergraph curve-to-line conversion program[11] that changes curve elements to line elements before running *DLGOUT*; however, this program also strokes the curves in the process of converting them to lines, and replaces the elevation values with null values. If the data theme of the IGDS file to be translated into GRASS vector format is elevation contours, it is advised to use the *DLGOUT* version that runs on Microstation 32, or *dxfout*. If some of the contour lines in an IGDS file are tagged with elevation values and some are tagged with null values, *DLGOUT* will not run. If all of the contours are tagged, or if the contours are not tagged at all (with elevation values or with null values), *DLGOUT* will run successfully. The advantage of using *DLGOUT* is that, if labels or contour elevation labels are able to be translated, little or no labeling will have to be done in GRASS *v.digit* after translation.

---

[10] *Intergraph DLGOUT User's Guide*, DMAP145 (Intergraph Corp., 1986).

[11] See the *Intergraph Spatial Editor Users Guide*.

Table 3

Target and Intermediate File sizes in Bytes

IGDS	*DXF*	ASCII *DLG*	ASCII *DIG*	Binary *DIG*	CELL
393,216	2,174,036	1,536,894	943,951	564,907	2,000,000
124,928	274,262	83,430		30,379	52,127
2,564,096	21,668,253			41,416,011	

Intergraph *dxfout*

dxfout is the Intergraph translator that converts IGDS vector data to *DXF* format. It runs in the Intergraph Microstation environment in either Microstation 32 or PC Microstation. It is useful for translating all thematic types of IGDS data (roads, streams, contours, flood plains, etc.). One important aspect of using *dxfout* is that, if IGDS contours (usually index contours) are labeled with graphic (visible) text, these can be translated into the *DXF* file *(dxfout)*, imported into GRASS as a vector file (*v.in.dxf*), read using the GRASS program *v.cadlabel,* and assigned as category labels to the nearest index contour. All contours that have text labels, will then have GRASS category labels (elevations) after *v.cadlabel* and will not have to be labeled in GRASS *v.digit.* The individual performing translation is again urged to evaluate the complete contour conversion path to DEM product before extensive editing, including the use of *r.surf.contour* or *r.surf.idw2.*

Based on the discussion above and in "How to Run Intergraph *DLGOUT*" (p 29), *dxfout* is best used on elevation contour data that meet the following conditions:

1. The contour data do not have internally tagged elevation values.

2. The contour data, digitized as curve elements, do have internally tagged elevation values, but the Microstation 32 version of *DLGOUT* (packaged in MGT) is not available.

3. MGT is available, but some of the contour lines in the elevation contour data are tagged with elevation (z) values and some are tagged with null values. (The user should check to see if the current version of *DLGOUT* in MGT will accept null z values.)

4. Only the index contours labeled with graphic (visual) text will be translated into GRASS.

5. The user prefers to use *dxfout* for its ease of execution, and does not mind labeling some (usually intermediate) contours after translation into GRASS. Labeling of all contour lines is required if the GRASS vector contour file is to be converted into a raster digital elevation model. You must label at least the index contours if the GRASS vector contour data will be displayed as a vector contour file.

dxfout is easier to use than *DLGOUT* and can be followed by the user-friendly GRASS contour text label extraction program *v.cadlabel* and a GRASS bulk contour labeling capability in *v.digit.* It will translate the IGDS line style (dashed, dash dot, etc.) as a solid line. A dashed line in IGDS will become a solid line in GRASS *dig.* To avoid problems, the individual performing translation must check graphic elements to verify whether lines are short and separate, or connected lines that are merely being displayed as a line style or pattern. These conditions can be further controlled with a *dxfout* environmental parameter file that is referenced during *dxfout* execution to define *dxfout* results.

To fully describe the capabilities of *dxfout*, it is necessary to describe the GRASS programs that increase the efficiency and ease of labeling files that have been translated using *dxfout*. (Refer to "How to Run Intergraph *dxfout*," p 32, and "Summary of the IGDS-DXF-GRASS Translation Sequence," p 37.)

Chapter Summary

DLGOUT would be a preferred translation method when contour lines have been digitized with continuous lines (no text index label breaks) that have been attributed with elevation values or when linestyle symbology can be used to derive category values. However, these conditions are most often true when the data were created with the use of *DLGOUT* in mind. Few of the databases encountered so far have these characteristics. Most map databases have been created with the focus of producing paper map products, so that housekeeping on the digital database is often in need of attention. Table 4 summarizes the material presented in this chapter.

Table 4

Advantages/Disadvantages of *DLGOUT* and *dxfout*

DLGOUT		*dxfout*	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
• Assigns major and minor *DLG* feature code to IGDS element symbology. • Converts IGDS line style to solid line style. • Ignores text as default. • Usually creates files smaller than *DXF*.	• Does not retain tagged elevation values for curve elements. • Will not run if tagged null z values exist in the file. • Creates null z values for z values of curve elements.	• Ease of use; free with Microstation products. • Translation Table for controlling conversions. • Simplifies complex data elements. • Translates text if present. • GRASS translation utilities can convert *DXF* text to GRASS labels. (Contour labelling)	• Creates large intermediate files. • No translation utility to convert z value attributes of contour lines. • No utility to assign attributes to graphic symbology.

9 EXPORTING *DLG* AND *DXF* FILES FROM INTERGRAPH

How To Run Intergraph VAX *DLGOUT*

Two files are required to run VAX *DLGOUT* on an IGDS design file. Both files are created by the user using a screen text editor and are stored in a flat ASCII file. The first file is a header file, given a filename with a prefix that matches the name of the design file, and the extension *.hdr*. For example, if the design file is named *contours.dgn;1* the header file should be given the name *contours.hdr*. The VMS name will become *contours.hdr;1*. This allows the *DLGOUT* program to find the header file without prompting the user for the filename.

The second file is a *DLGOUT* parameter file. It can be given any name by the user; for example, *contours.param*. The full name in VAX VMS will then become *contours.param.;1*. Underbars should not be used in the filename because this may cause confusion in the VMS operating system. The user will be prompted for the parameter filename when running *DLGOUT*, and should type the full VMS filename, including the semicolon followed by the version number.

DLGOUT also has the capability to translate IGDS attributes stored in the Intergraph VAX-based DMRS. This option of *DLGOUT* was not evaluated in this study, however, and will not be included in this report.

The Header File

An example of the *DLGOUT* header file is:

BAN= FT. SILL TRANSPORTATION - State Plane, Feet

The only required characters of the statement are "BAN=". Refer to the *Intergraph DLGOUT* User's Guide for more information. This example header file is for a Fort Sill, Oklahoma transportation, or roads, design file. The user decided to include the coordinate system (state plane) and the units (feet). There is one space after the equals sign in the header file. The other special characters, the dash, comma, and period, are optional.

The Parameter File

An example *DLGOUT* parameter file is shown below:

```
OPT:  S
OVR:  Contour Facet 1
MLT:  1000
CAS:  Network
NOD:  1
REC:  ID=AREA,ET=TEXT,LV=60,FC=000-000
REC:  ID=AREA,ET=TEXT,LV=60,CO=4
REC:  ID=LINE,ET=LINE,LV=1,CO=6,FC=020-0001
REC:  ID=LINE,ET=LINE,LV=5,CO=3,FC=020-0002
```

There are one or two spaces after the colon in the parameter file, but there are no spaces on either side of the commas or the equals signs. The purpose of the parameter file is to identify the kinds of elements (ID= line, area, or node) that are to be translated, what level they reside on (LV= level_number), and what *DLG* feature code they are to be assigned (FC= number).

The example parameter file is a contour design file of one facet of Fort Drum, GA. The minimum requirements for a parameter file are: one OPT statement, one OVR statement, one MLT statement, one CAS statement, one NOD statement, and at least three REC statements (one for the *DLG* inside area, one for the *DLG* outside area, and at least one for the level of data being translated).

The argument following the keyword OPT is either an "S" or an "M". "S" indicates that only one contour file is being described by the parameter file (singular mode), and "M" indicates more than one design file will be described in the parameter file (multiple mode).[12] This document will cover only the singular mode. The line below the OPT statement is left empty.

The next entry is the OVR statement. This keyword is followed by a map name chosen by the user. The name cannot exceed 20 ASCII characters. In the Fort Drum example parameter file, the name is "Contour Facet 1".

The MLT statement is typed next. The argument within this statement is a number that equals the design file Sub Units (SU) multiplied by the Positional Units (PU). The SU and PU for each IGDS design file can be found by checking the *Design Options* menu. In the Fort Drum example, the SU is 10 and the PU 100, so the MLT value is 1000. New versions of VAX *DLGOUT* (1989 and up) have no limit on the MLT value.

The CAS statement has two options, Network or Area. Network applies to linear data, and Area applies to data having many closed polygons. If this statement is omitted, the CAS defaults to Area.[13]

The NOD statement allows the user to direct the program to place nodes throughout the design file if this was not done during digitizing. The NOD options are:

- 0 - Nodes are already present
- 1 - Place nodes throughout the design file
- 2 - Place nodes throughout the design file and tag them with identifications.

If this statement is omitted, the NOD defaults to "0".

The next entry is the first REC statement, which must be the REC statement for the inside area of the *DLG* file. The inside and outside areas are user-defined by drawing a box around the contour data, and by using the IGDS *place block* command in the IGDS graphic environment (*ice*). The box must be drawn on level 60 of the design file. For the inside area, the ID= is always AREA. The inside area is identified by placing an element type (ET=) text node inside the area after it is drawn using the IGDS command *place text node*. The argument following ET= is therefore TEXT. The feature code (FC=), or category label for the *DLG* inside area, is always 000-0000. The first three zeros of this number are the *DLG* major code and the last four zeros are the *DLG* minor code. Refer to *Digital Line Graphs from 1:24000 Scale Maps*, National Cartographic Information Center, Reston, VA. The entire first REC statement will always appear as it does in the Fort Drum example.

---

[12] For further information refer to pages B-3 through B-5 in the *Intergraph DLGOUT User's Guide*.

[13] Refer to *Intergraph DLGOUT User's Guide* for further information.

The second REC statement always defines the outside area of the *DLG* file. This is user-defined by placing a text node outside the box that was drawn around the data. To ensure that this text node is distinguished from the first text node, it is best to give it a different color using the *change color* option under *element manipulations* in the IGDS graphics environment. The color is then indicated in the REC statement by CO=color_number. The outside area should not be given a *DLG* feature code (FC=).

The next REC statements identify design file levels that contain data to be translated. Each level or each feature code per level must be identified using a separate REC statement. Feature codes on the same level or on separate levels, can be distinguished by linestyle (ST=), lineweight (WT=), and linecolor (CO=). Refer to the *Intergraph DLGOUT* User's Guide for other REC statement keywords.

Only the minor feature code will be translated by the GRASS *v.import* program, which imports *DLG* data into GRASS. After importing the *DLG* data, the GRASS vector lines will have GRASS category labels corresponding to the *DLG* minor feature code. The labels will be displayable in *v.digit* after importing the data. For example, if the *DLG* feature code (FC=) is 020-0003, the GRASS vector category label will be 3.

The number of feature codes in an IGDS design file that can be distinguished using IGDS levels, line styles, line colors, and line weights are limited to the number of design file levels, and the IGDS line symbology. There are seven line styles, seven line colors, seven line weights, and 63 design file levels (of which 62 are available). The maximum number of feature codes that can be converted from IGDS to *DLG* using the VAX *DLGOUT* translator is (7×3)×62 or approximately 1302 feature codes.

In the Fort Drum example contour file, there are two design file levels that contain data to be translated (although other levels may contain data of no interest). The index contours, with a contour interval of 25 ft and the line color yellow (CO=6), reside on level 1. They are digitized as lines (versus areas or nodes) and are therefore identified as ID=LINE. If the contour lines are digitized as Intergraph curve elements, they are still assigned the identification ID=LINE in the parameter file. The options for ID= are only LINE, AREA, and NODE. The intermediate contours, with a contour interval of 5 ft and the line color of red, reside on level 5. This parameter file would translate only lines of color 6 on level 1, and assign them the GRASS category number 1. It would also translate only lines of color 3 on level 5, and assign them GRASS category number 2. It will not translate the box on level 60.

Graphic Preparation

As mentioned in "The Parameter File" (p 29), a box and two text nodes must be placed on level 60 in the IGDS design file. Four text strings must also be placed by the user on level 60. One text string should be placed at each of the four corners of the box defining the *DLG* universe. The text strings do not have to be located at exactly the box corners, but can be placed anywhere approximately near the corner and outside of the box. Each text string should contain the latitude and longitude of the border of the whole map area. For example, the approximate latitude and longitude boundaries of the entire Fort Drum installation may be placed around each translated IGDS facet of Fort Drum. The latitude and longitude, then, do not correspond to the four corners of each design file, but to the whole geographic area. These coordinates are read by the *DGLOUT* translator and placed in the *DLG* header. The text strings should be typed in the following format:

SW LONG= -100.00 LAT= 44.55

DLGOUT Execution

After logging onto a VAX or microVAX, at the VAX prompt type: "@pro_dd_dlgout:dlgout", and then follow the interactive prompts described in the Intergraph *DLGOUT User's Guide*, pages 3-3 to 3-10.

How To Run Intergraph *dxfout*

dxfout is an Intergraph Microstation-based program that converts IGDS data to the *DXF* standard file format. *dxfout* comes packaged with Microstation PC and Microstation 32. For a Microstation PC, the *dxfout* program is initiated using one of these procedures:

1. Select the "DXF Translations" found under the *Utilities, Transfer/Translate* options in the *Microstation Command Environment* (*MCE*) menu. This begins the *DXF* configuration menu program, allowing the review and editing of the following tables that contain translation parameters to control various aspects of data and file structure:

   a. Cellname translation table
   b. Colors translation table
   c. Font translation table
   d. Level translation table
   e. Line code translation table
   f. Weight translation table
   g. Line style definition table.

These tables facilitate the control of conversion of different data elements and layer structure to the *dxf* file format.

In addition to allowing control over these filenames, their directory location, and contents to customize the translation, the *DXF* configuration program also prompts for the directory to create the output *dxf* file. Defaults established during installation are adequate for simple conversions. Completion of review and editing returns the user to the main *DXF* configuration menu.

2. Select the "*Run Microstation DXF-OUT*" option. This prompts the user for the name of a *.dgn* file for conversion, and the name of the *dxf* file to create. A final prompt requests whether the *dxf* file is to be scaled to the master or subunit coordinates defined within the microstation database seed file. Master units are converted to correct coordinate values for State Plane coordinate systems.

Once this final prompt has been answered, *dxfout* begins processing. *dxfout* makes three passes through the design file, first to calculate drawing extents of the file, then to create blocks from cells, and lastly to process the individual elements.

For a Microstation 32, the *dxfout* program is initiated using one of these procedures:

1. Access the SUPPLEMENT prompt through MCE UTILITIES by keying in the following:

   GRAPHICS: util
   UTILITIES: supp

2. Key in the following at the SUPPLEMENT prompt:

   SUPPLEMENT: *dxfout*

3. Follow the prompts that appear and key in the appropriate names for the design filename to be converted and the *dxf* filename to be output, as illustrated for the Microstation PC *dxfout* procedure. Translation table parameter files are identical to those shown in the Microstation-PC procedure above.

Note that in step 3 of either the Microstation PC or Microstation 32 procedure, you can force the *dxf* file to be created in other than the default directory by specifying a full pathname.

10 IMPORTING *DLG* AND *DXF* FILES INTO GRASS

How To Import *DLG* Files Into GRASS

Once the IGDS file is translated into *DLG*, the next step is to import the *DLG* file into GRASS. The output DLG file from the Intergraph VAX *DLGOUT* program is an ASCII *DLG* file. It can be viewed on the screen in VMS using the *type* command, or viewed in UNIX using the *vi, view, more,* or *cat* command. It will be necessary to network, or transfer using magnetic media, the ASCII *DLG* file to a GRASS UNIX machine. Once the file is accessible to GRASS, the user need only start GRASS by typing the appropriate command (*GRASS 4.0*), and then run the GRASS command *v.import*. This GRASS program converts ASCII *DLG* files, binary *DLG* files, and ASCII *dig* files to binary *dig* files, which are the GRASS binary vector files having GRASS vector format. To run v.*import*, refer to the GRASS *Reference Manual.*

When *v.import* is complete, the user can display the binary vector file in GRASS using either the commands *d.vect* or *v.digit*. When using the command *v.digit*, the user has the capability to display the line labels that originated as *DLG* feature codes in the *DLG* file (p 29) and were translated into the GRASS labels. The line labels are displayed in v.*digit* on the lines they identify.

At this point following translation and accuracy of the translated file may be checked (see Chapter 12, "Translation Accuracy Assessment," p 40) and editing or labeling in GRASS *v.digit* may be desired.

Summary of the IGDS-*DLG*-GRASS Translation Sequence

The procedure for translating IGDS-vector data into GRASS binary vector format using the VAX-based Intergraph *DLGOUT* program to produce a *DLG* vector file as an intermediary vector format is summarized below:

1. Display the IGDS file in the IGDS graphics environment to identify data levels, line symbology, element types, global origin, and the coordinate system.

2. Conduct tabular queries of the graphic data using the IGDS Supplemental Libraries or the "Edit Display Graphics" (p 20).

3. Edit in the IGDS graphics environment to repair digitizing inaccuracies. e.g., line overshoots, undershoots, duplicate lines, etc., or to move data to a different level. These repairs are not required to run *DLGOUT* but may be required for accurate map analysis in the GIS environment. Vector editing can also be done in the GRASS *v.digit* program after translation. The GRASS vector file, however, will not have internal levels, so manipulation of IGDS data to different IGDS design file levels must be accomplished in the IGDS environment.

4. On IGDS level 60, place a box around the *DLG* data, using IGDS graphics commands. Place a text node inside and outside the box, and a latitude and longitude text string at each of the four corners of the box.

5. Prepare a header file for the *DLGOUT* program using a screen text editor.

6. Prepare a parameter file for the *DLGOUT* program using a screen text editor. This file identifies the inside area of the *DLG* file, the outside area, and the level and feature code (optional) of all data to be translated.

7. Execute *DLGOUT.*

8. Transfer the ASCII *DLG* file from the VAX VMS operating system to a UNIX operating system accessible by GRASS.

9. Convert the ASCII *DLG* file to a GRASS binary *dig* file by running the GRASS program *v.import.*

10. Convert the GRASS binary vector *dig* file to the UTM coordinate system if desired, using the GRASS program *v.transform.* The binary *dig* file must first be converted to a GRASS ASCII vector *dig_ascii* file by running the GRASS program *v.out.ascii.* If you have compiled the MAPGEN plotting programs which are distributed with GRASS, you can use a program called proj to find a UTM coordinate which corresponds to a given STATE PLANE coordinate. The procedure would involve first obtaining four to 10 STATE PLANE coordinates by querying in the IGDS environment. You would use the MAPGEN program proj to determine the corresponding UTM coordinate for each STATE PLANE coordinate. You create an ascii pointsfile of these coordinate pairs for use with the GRASS program v.transform. v.transform uses this ascii file to transform all STATE PLANE coordinates in the dig_ascii file which was created with v.in.dxf to correct UTM coordinate position. You complete the conversion by running the GRASS program v.in.ascii, followed by v.support.

11. Convert the *dig_ascii* UTM file (or, if coordinate transformation was not done, the State Plane file) to GRASS binary vector *dig* format by running the GRASS program *v.import.*

12. Check registration accuracy in GRASS, and label and edit the file in GRASS *v.digit* if necessary.

How To Import *DXF* Files Into GRASS

1. Create a *DXF* file with the Microstation *dxfout* program or any CAD program that generates a standard *DXF* file. Note that *DXF* file translations do not guarantee 100 percent data conversions. Problems more often occur in translating 3D data across programs. However, for 2D mapping databases, translation results are usually satisfactory. Read the *dxfout* notes accompanying the microstation package for the current description of capabilities and caveats.

2. Transfer the ASCII *DXF* file from its directory to a GRASS directory named *dxf* on either the same machine or a different machine.

3. Convert the ASCII *DXF* file to a GRASS ASCII vector file, *dig_ascii,* by running the GRASS program *v.in.dxf.*

The v.in.dxf conversion program generates GRASS *dig_ascii* and *dig_att* layers from a *DXF* formatted file. Each level in the *DXF* file is converted to a separate *dig_ascii* layer. For each *DXF* level containing text, a *dig_att* file is also generated. These output files are placed in the GRASS *dig_ascii* and *dig_att* directories. The *v.in.dxf* program will only recognize points, lines, polylines, and text in the *DXF* format, and will translate these to the GRASS vector format; other types of data are ignored. The following command line variations may be used:

v.in.dxf [-a] dxf=name [lines=name[,name,...]] [labels=name[,name,,,]] [prefix=name]

Flags: -a = output to an ASCII vector file [default:binary]

Parameters:

dxf = name of the *DXF* file, including its full pathname
lines = *DXF* levels where the line data reside.
labels = *DXF* levels where the label data reside
prefix = prefix for *dig*or *dig-ascii* and *dig-att* output files

Examples of line options are:

- lines=15 Line data on *DXF* level 15
- lines=15,16 Line data on *DXF* levels 15 and 16
- lines=15:16 Any line data on *DXF* level 15 should be placed in the *dig* or *dig_ascii* file for *DXF* level 16
- labels=name specifies the *DXF* levels where the text data reside. Options remain the same as for lines
- prefix=name specifies that the name of the output *dig* (*dig_ascii*) and *dig_att* files are formatted as *prefix.extension*, where prefix is the prefix of the *DXF* filename and extension is the *DXF* level number. For example, for the *DXF* file named *streams.dxf*, that has line data on level 15, the output *dig_ascii* file would be named streams.15.

The prefix of the output filename can be changed with this prefix option by typing prefix=new_prefix on the command line. For example, given a *DXF* file named *cont.dxf* that contains contour lines and contour line labels on the following levels:

- index contour lines on level 9
- intermediate contour lines on level 11
- index labels and intermediate contour lines on level 12.

Using the default options for v.in.dxf:

v.in.dxf cont.dxf dxf=cont.dxf

is the same as:

v.in.dxf dxf=cont.dxf lines=9,11,12 labels=12

and generates the following GRASS dig dig and dig_att files:

dig: cont.9, cont.11, cont.12
dig_att: cont.12

However, the *cont.12* file contains intermediate contour lines that should be in the *dig_ascii* file *cont.11*. If the user would also like to change the prefix on the resulting files, keying in the following command:

v.in.dxf dxf=cont.dxf lines=9,11,12:11 labels=12 prefix=contour

will generate the following GRASS files, in which there are no contour lines in the text vector file:

dig: contour.9, contour.11, contour.12
dig_att: contour.12

4. If you wish to convert the STATE PLANE coordinates of the dig_ascii file to UTM coordinates, obtain four to 10 STATE PLANE coordinates by querying in the IGDS environment. Determine the corresponding UTM coordinate positions for these points using the MAPGEN program, proj, and set up the required ascii pointsfile required by the GRASS program v.transform. Run v.transform to move the STATE PLANE coordinates to correct UTM position. If you have to, use the UNIX *MX* command to move the resulting UTM dig_ascii file(s) to a GRASS LOCATION with a UTM coordinate system.

5. The vector files can be displayed at this point using the *d.vect* command, but *v.support* must first be run on the *dig* files before they can be edited in *v.digit*. It is likely that the file will contain unsnapped nodes, overshoots, gaps, and replicated lines. The translation program does not contain any of the quality control functions available in digit that will prevent improper data from getting into GRASS. If present, *DXF* entities are placed in output file(s) corresponding to the levels on which they occurred in the *DXF* input file. The header information (such as owner's name, map's name, date, and scale, and UTM zone), for the GRASS vector files will also need to be edited in *v.digit*.

6. The *v.in.dxf* program attaches attributes only to *DXF* text data that is converted to GRASS vector data (such as contour line labels). Attributes are not attached to converted *DXF* line data. For each level of text data in the *DXF* design file, *v.in.dxf* generates a vector file consisting of rectangular boxes (lines) that are drawn around the *DXF* text data, and uses the text values to create a GRASS attribute file for the boxes. The vector and attribute files can then be used to label contour lines with the *v.cadlabel* program. Refer to Chapters 13 and 14 (pp 42,43) for details regarding the *v.cadlabel* program.

Summary of the IGDS-*DXF*-GRASS Translation Sequence

The procedure for translating IGDS vector data into GRASS binary vector format using the Intergraph MicroStation *dxfout* program to produce a *DXF* vector file as an intermediary vector format is as follows:

1. Display the IGDS file in the IGDS graphics environment to identify data levels, line symbology and data theme per level, global origin, and the coordinate system.

2. Edit in the IGDS graphics environment to repair digitizing inaccuracies: line overshoots, undershoots, duplicate lines, etc., or to move data to a different level. It may be desirable to move data, or whole levels, to a new IGDS file to isolate the data of interest and reduce the size of the IGDS file being translated into *DXF*. The vector repairs mentioned are not required to run *dxfout* but are required in the GIS environment. Vector editing can also be accomplished in the GRASS *v.digit* program after translation. The GRASS vector file, however, will not have internal levels, so manipulation of IGDS data to different IGDS design file levels must be accomplished in the IGDS environment.

3. Record the data theme(s) that reside on each level, and the level number. This way when the GRASS program *v.in.dxf* is used to import the *DXF* file into GRASS, the separate vector files it produces for each *DXF* (and, therefore, IGDS) level can be identified in GRASS.

4. Execute *dxfout.*

5. Transfer the ASCII *DXF* file from its directory in the MicroStation 32 UNIX operating system to a GRASS directory named *dxf* on either the same or a different machine.

6. Convert the ASCII *DXF* file to a GRASS binary vector file, *dig*, or an ASCII vector file, *dig_ascii*, by running the GRASS program *v.in.dxf.* If you would like to convert the file from one coordinate system to another you must then run the GRASS program *v.transform* using four to 10 coordinate pairs as registration points.

7. To convert the STATE PLANE coordinates of the dig_ascii file to UTM coordinates, obtain four to 10 STATE PLANE coordinates by querying in the IGDS environment. You must then run the MAPGEN program proj to determine their corresponding UTM coordinates. Then run the GRASS program v.transform using the four to 10 coordinate pairs in a pointsfile.

8. Move, using the UNIX *mv* command, the UTM *dig_ascii* file(s) to a GRASS LOCATION with a UTM coordinate system.

9. Convert the *dig_ascii* UTM file (or, if coordinate transformation was not done, the State Plane file) to GRASS binary vector *dig* format by running the GRASS program *v.in.asii.* Then run *v.support.*

10. Check registration accuracy of the map in GRASS. If necessary, edit the binary file in GRASS *v.digit.*

11. If the data is contour data, the user may want to run the GRASS program *v.cadlabel*, which converts IGDS elevation text labels that were originally located on an IGDS design file level, to GRASS attributes. This program also attaches the attributes to the nearest GRASS contour line. *v.cadlabel* must be followed by *v.support.*

12. Use the GRASS command *v.patch* to patch together the desired GRASS vector files that represent the different levels of the IGDS and *DXF* file. *v.patch* must be followed by *v.support.*

13. Label Contour lines that are not labeled in *v.cadlabel* and that require elevation category values, in GRASS *v.digit.* GRASS *digit* has a bulk contour labeling capability. The *v.support* program must be run on maps after they are labeled.

11 COORDINATE CONVERSION IN GRASS

There are two coordinate conversion programs distributed with GRASS that are useful for converting GRASS vector files between the STATE PLANE coordinates system and the UTM coordinate system. It is important to note, however, that GRASS LOCATIONs can be registered with the State Plane coordinate system, the UTM coordinate system, the x,y coordinate system, or the latitude/longitude coordinate system. Conversion from the State Plane coordinate system to the UTM coordinate system would be desirable if a majority of other GRASS map layers for a geographic area were in the UTM coordinate system, and if the GRASS map layers converted from IGDS to GRASS were in the State Plane coordinate system. Having all of the map layers in the same coordinate system would allow the user to analyze them together in the same GRASS LOCATION.

The two GRASS conversion programs are:

1. v.transform (a GRASS program)
2. proj (distributed under the MAPGEN directory).

v.transform transforms an entire vector map layer from one coordinate system to another (i.e. State Plane ↔ UTM). In order to run v.transform, the user must know the coordinate position of four to 10 points in both the source coordinate system and the target coordinate system. The user typically queries in the source data display environment (IGDS or GRASS) to obtain four to 10 coordinate points. The Specific location of coordinates selected is unimportant. However, the user should select points that are distributed across the entire geographic region of the source data set. To determine target destination coordinates for the source coordinates, the user must run the MAPGEN program proj. proj is a coordinate projection conversion program that will echo the respective position of a State Plane coordinate pair in the UTM system and vice-versa. The user inputs these source and target coordinate pairs to v.transform either interactively or via a points file. The interactive version of v.transform produces registration coordinate residuals that can be checked by the user to estimate transformation accuracy. When the registration coordinate residuals are acceptable good point values can be placed in a points file to be used in the command line version of v.transform to convert data in a production mode. Source data in dig_ascii format is transformed to target coordinate system position and output in a dig_ascii and associated dig_att file format. The user then runs v.in.ascii on the dig_ascii vector file to create a binary vector file (dig file) and then v.support to create the associated topology file (dig_plus).

In a reverse fashion, the MAPGEN program proj can be used to check the accuracy of a translated and transformed map (see Chapter 13, p 42). It can also be used when converting a GRASS vector map from the UTM coordinate system to the STATE PLANE coordinate system. For further information on these programs, consult the GRASS *Reference Manual*.

12 TRANSLATION ACCURACY ASSESSMENT

When a vector map is successfully translated from IGDS to GRASS, one of the first concerns is to check the coordinate accuracy of the map. This can be done in several ways. One way is to display the map in GRASS and overlay a GRASS vector map of known accuracy. If the two maps coincide graphically, the translated map may be of acceptable accuracy. A further test is to print the translated vector map on mylar or some other transparent material, at a specific scale, and overlay the printed map on a USGS map of equivalent scale, for example, 1:24000. If the maps register closely, the translated map may be acceptable. This method, however, may reveal some distortion from the output device. The GRASS command *d.where* can be used to echo coordinates to the screen. The location of these coordinates can be compared to those on a reference map.

The MAPGEN program proj can be used to confirm the coordinates of a translated and/or transformed vector map. For example, State Plane coordinates in the IGDS map can be queried in the IGDS environment. The program proj can be run to obtain the equivalent UTM coordinates for the queried IGDS coordinates. It can also be used to verify translations between the coordinate systems.

A plot of the translated vector map can be made using the GRASS-MAPGEN interface and compared to original CADD check plots of the IGDS files. This procedure would confirm whether the original precision and accuracy of the translated map has been retained.

The plotting capabilities of MAPGEN can be used to assess map accuracy, also.

13 LABELING VECTOR FILES IN GRASS

The capabilities to label vector files in GRASS reside in the v.digitizing, editing, and labeling program *v.digit* (refer to the GRASS *Reference Manual*). *DXF* files that have been imported into GRASS, using the GRASS program *v.in.dxf* can also be labeled by running the GRASS program *v.cadlabel* (refer to the GRASS *Reference Manual*). A description of the labeling capabilities of these two programs is the subject of this chapter.

v.digit

v.digit is invoked by typing the command *v.digit* at the GRASS prompt. After selecting the digitizing device, inputting the map title, coordinating information, etc., the user is presented with the *v.digit* main menu. Among the main menu choices is the Label Menu, which is accessed by typing an upper case "L". Figure 4 shows the Label Menu.

To label digitized lines, such as roads or streams, option *l* is chosen. To label digitized areas, such as lakes, ponds, or buildings, option *a* is chosen; to give all remaining lines in a map the same label, *option B* is chosen; and to give connected lines the same label, option *m* is chosen. To increase the speed with which contour files are labeled, option *c*, the *Label Contours* option (Figure 5) can be selected. screen.

The left, middle, and right mouse buttons are used to choose and accept a contour line with an elevation label. The program then prompts the user to select a second labeled line. Both the first line selected and the second line selected must already be labeled. When the second line is chosen, a rubber band line is created on the screen between the two contour lines. When the second contour line is

GRASS-DIGIT VERSION # 4.0 **Label Menu**

Label Options:

a - Label Areas	m - Label Multiple Lines
l - Label Lines	M - Unlabel Multiple Lines
s - Label Sites	c - Label Contours
A - Unlabel Areas	i - Contour Interval <5>
L - Unlabel Lines	
S - Unlabel Sites	
B - Bulk Label Remaining Lines	
h - Highlight Lines of Category #	
d - Display Areas of Category #	
q - Return to Main Menu	

Digitize Edit Customize Toolbox Window Help Zoom * ! ^
GLOBAL MENU: Press first letter (upper case).

Figure 4. Label Menu.

Buttons:
Left: Choose line
Middle: Abort
Right: Accept chosen line

Figure 5. Label Contours Option.

accepted, all of the contour lines between the two accepted contour lines are automatically labeled using the designated contour interval. This will happen if all program criteria are met (i.e., the sum of the intermediate contours must equal the labeled value of the first contour minus the labeled value of the second, or the labeled value of the second contour minus the labeled value of the first, at the designated contour interval). The contour interval *i* is displayed on the *Label Menu*, and can be set directly from that menu prior to selecting the option c. For example, when the contour interval is 5, and the first accepted contour line has a label of 75, and the second accepted contour line has a label of 100, then all of the intermediate contours will be labeled in 5-unit increments: 80, 85, 90, 95. This increases the speed of labeling contours and also decreases the number of label assignment judgments that have to be made by the user. The entire length of a contour line will be labeled if there is no break in the line. The contour labeling option will label a line until it comes to a starting node or an ending node. This is another advantage of digitizing contour lines without creating breaks in the lines (see Chapter 6, "*Targeting and Editing Vector Data,*" p 22).

v.cadlabel

Background

The GRASS vector files that *v.cadlabel* can be applied to are only the output dig files of the GRASS program *v.in.dxf*. The *v.in.dxf* program (p 35) imports *DXF* ASCII files into GRASS *dig_ascii* format and creates a separate GRASS *dig_ascii* file for every level in the *DXF* file. Each level in the *DXF* file corresponds to the same level in the IGDS file, if the file was translated from IGDS using the Intergraph MicroStation program *dxfout*. The GRASS dig_ascii files are given a filename extension matching the *DXF* level number. For example, if the *v.in.dxf* output filename chosen by the user is *contour_facet1*, and *DXF* levels 5, 7, and 9 contain data that will be translated, then the GRASS *dig_ascii* files that are created will be named *contour_facet1.5*, *contour_facet1.7*, and *contour_facet1.9*.

When most Intergraph IGDS contour files are digitized, the contours that have text labels are digitized on one level (these are often called index contours), intermediate contours without text labels are digitized on another level, the elevation text is digitized (or in IGDS terms placed) on a third level, and the benchmark symbols are digitized on a fourth level. When *v.in.dxf* is run, each of the *dig_ascii* output files will correspond to a *DXF* level. In this case, there will be a *dig_ascii vector* file containing only index contours, one containing only intermediate contours, one containing only benchmarks, and one containing only text. During the *v.in.dxf* program, a graphics box is drawn around each elevation text string in the GRASS text *dig* file. When *v.in.dxf* is complete, the user should run *v.import* to convert each *dig_ascii* file to a binary *dig* file, and then run *v.support* before running *v.cadlabel.*

The v.cadlabel Program

The graphics box created in *v.in.dxf* is used by the *v.cadlabel* program to label the index contours in the *dig* index contour file. *v.cadlabel* uses the box to locate the nearest line to the text and therefore the contour line that should be assigned the elevation value of the text. The nearest contour is then assigned the GRASS category value of the text. For example, if the text within the box is 1250, this value would be assigned to the nearest line, which would denote an elevation of 1250 ft. The graphic boxes remain in their original *dig* file (they are not transferred to the index contour *dig* file) and the GRASS category number (1250) is stored in the associated *dig_att* file. All index contours that have text labels will be assigned GRASS category labels in *v.cadlabel.* When *v.cadlabel* is complete, the user should run *v.support*, which will create the GRASS topology file (*dig_plus*).

The labeled vector file can then be displayed in *v.digit.* The user can see the labels on the index contours by using the *Display Line Labels* option in the Display Menu of *v.digit.* The display Menu is reached from the Main Menu by entering the *Customize* Menu and then selecting the *D* option for *Enter Display Options* Menu. The dig file containing the graphic boxes can be displayed as an Overlay Map in *v.digit* to use as a reference. The Overlay Map option is located in the Customize Menu.

When the data is contour data and the advantages of *v.cadlabel* are warranted, the user will ordinarily want to run *v.cadlabel* first, to label the file, then use the labeling capabilities in digit if some lines remain unlabeled or are labeled incorrectly. If the contour file is eventually to contain the intermediate contour lines as well, the labeled *dig* file from *v.cadlabel* should be patched to the intermediate contour *dig* file using the GRASS program *v.patch*, and then the intermediate contours can be labeled in *v.digit.* This sequence of running programs will make checking the labels of the index contours easier, when the intermediate contours are not patched in yet, and it will make the labeling of the intermediate contours in *v.digit* easier when the already labeled index contours are present. The labeled index contours are used to label the intermediate contours in the *Contour Labeling* facility of *v.digit.* For information on how to run *v.cadlabel*, refer to the GRASS *Reference Manual.*

14 GUIDELINES FOR DIGITIZING MULTIPURPOSE IGDS DATA

Multipurpose IGDS data is defined as Intergraph IGDS CAD vector data that will be translated into the GRASS vector format. The specifications presented here are the result of the evaluation of the translation programs that convert IGDS data to either *DXF* Standard ASCII format or the USGS vector *DLG* format. These two translation programs are the VAX based Intergraph program named *DLGOUT* and the MicroStation based Intergraph program named *dxfout*. GRASS has the ability to import both *DXF* formatted data and *DLG* Optional 3 formatted data.

The following digitizing specifications may be applied to IGDS data when a direct translation program is used, e.g., one converting IGDS data directly to GRASS *dig* format. Even if a translation program provides some options for graphic correction, such as node snapping, all the the requirements outlined below are not likely to be provided by a post-digitizing translation program. At present, a direct IGDS to *dig* translator is not available but may be written in the future. IGDS data specifications are:

1. Digitized IGDS data should be accompanied by written documentation stating the date of creation, the coordinate system the data were digitized in, the outside coordinate boundaries (for *DLGOUT*), the working units, sources of data, facet diagram and file nomencalature, linestyle, lineweight and linecolor legend, active design levels, contents of design levels, Statement of Work, etc., Refer to Chapter 16, "Documentation of IGDS Data Intended for Translation" (p 49).

2. The coordinate system must be the UTM or State Plane coordinate system. The digitized coordinates must match the coordinates for the system that occur on the ground for the geographic area being digitized.

3. The number of digits to the left of the decimal point in the coordinates representing the graphic locations must be the same as the number of digits to the left of the decimal point required by the coordinate system used: State Plane or UTM. In other words, do not abbreviate the coordinates that have been found in some mapping databases.

4. The database, if partitioned, should be digitized so that each file is one rectangular or square facet composing a grid that covers the entire geographic area under consideration. (Outer edges of design files should not be irregular.)

5. Design file size should be kept as small as possible while still retaining the integrity of the graphic lines. File size can be reduced by digitizing files by topic, for example; a contour file, a transportation file, and a streams file.

6. The graphic elements selected during digitizing to represent the geographic data (i.e., road lines, building or sidewalk outlines, stream courses, water body boundaries, contour lines, etc.) should be primary elements. Complex elements should not be used in geographic data that will be translated.

7. Patterns should not be used to depict linear elements, such as streams, railroads, or roads. In place of patterns, linestyle should be used to depict the different stream categories, the presence of a railroad, or the outside lines of a double-lined road. A separate linestyle should be used to depict the double-lined road's center line.

8. Geographic data that represent a single theme should be digitized on separate design file levels, or in separate design files. For example, roads represented by one line should be digitized on one level, roads represented by two graphic lines should be digitized on another level. Sidewalks should be

digitized on a third level, parking lots on a fourth, streams on a fifth, and buildings on a sixth level. Single-lined roads must be separated from double-lined roads. A raster GIS can use double-lined roads in analysis if each subcategory of the double-lined roads is digitized as a closed vector polygon and differentiated by line style, line color and/or line weight. The center lines for double-lined roads should be digitized on a separate level from all other data.

9. Text should always be digitized on a different level than the graphic data.

10. Subcategories within a geographic theme should be given unique identifiers using IGDS line style, line weight, and/or line color. For example, roads could be subdivided into primary roads, secondary roads, and trails, or other appropriate categories. In this case, the primary roads would have a unique line style, secondary roads a different line style, and trails a third line style. Or each of the road subcategories could be digitized on a separate level, with or without unique identifiers. For example, primary roads could be digitized on level 5 with linestyle 1, and secondary roads could be digitized on level 6, also with linestyle 1. IGDS to *DLG* or *DXF-dig* translators can assign GRASS category numbers to translated data only if it is either separated by linecolor, lineweight, linestyle, or the design file level.

11. Streams and water bodies should be subcategorized. For example, streams/rivers should be categorized into primary streams, secondary streams, tertiary streams, etc. Water bodies should be divided into natural lakes, natural ponds, manmade lakes or ponds, manmade canals, etc.

12. Graphically, contour lines are represented as both linear elements and closed polygons (i.e., depressions or ridgetops). These contour lines should be digitized as continuous vector lines and should not be broken. If text labels are to be created for the index contours, the text should be placed on a separate design file level. Breaks in the contour lines at the location of the text label should not be made.

13. Primary contour lines and secondary contour lines should be identified by a unique linestyle, linecolor, and/or lineweight, and should be digitized on separate levels. Primary contours occur at a maximum interval, for example at a 25-ft interval: elevations 25, 50, 75, 100, and 125. Secondary contours occur at a minimum interval. In the preceding example, secondary contours would occur every 5 ft between the primary contours. Secondary contour lines should not be broken. All discontinuous secondary contour lines, however, could be digitized on a separate level. This level would not be translated into GRASS.

14. All contour lines should be tagged with Intergraph low range z values. These are the elevations of each contour. Some translation programs will not run even if one contour remains untagged or if one contour is tagged with a null z value (example: Intergraph Terrain Modeling System software).

15. The graphic lines in contour files that represent elevation depressions should be digitized on a separate design file level.

16. Contour lines should be digitized as line elements, not Intergraph curve elements.

17. During digitizing, undershoots, overshoots, and lines digitized twice (identical duplicate lines and unidentical duplicate lines) should be avoided. When lines are intended to meet at a point, they should be digitized to meet exactly. Lines should not stop short or overshoot the intended termination point. A line digitized once should not be digitized again or duplicated in exactly the same location using CAD software.

18. Reference files should be detached after completing the digitizing of each IGDS facet, and all files should be compressed.

15 GRASS-INTERGRAPH DATABASE UPDATING

After digital map data is translated between the Intergraph CAD system and the GRASS GIS, it is relevant to ask whether identical map layers on two systems both need to be updated. The answer to this question is best supplied by contrasting the thematic map layers characteristic to each system (Table 5).

The mapping database usually consists of a planimetric (roads, sidewalks, buildings), contour and utility set of files. A GIS usually contains vector maps for roads, streams, elevation, soils, geology, land use, land cover, wetlands, archaeological areas, and wildlife habitat. Of the CADD maps listed, only roads, streams, and contours are commonly used in a GIS. Of these, only the roads map requires updating on a continual basis. Among the map layers unique to a GIS, the soils map holds the most obvious potential for use in a CADD system, and soils maps are rarely updated. The time required to update shared map layers, then, can be minimized.

In summary, it is necessary to update only frequently changing files. Updates should be done in only one system, and then transferred to the other. Table 6 outlines updating capabilities for each system.

Table 5

Vector Data Themes

Layers	GRASS	IGDS CADD
Contours	Used to derive raster elevation model 100/500 yr. flood limits text (100 & 400 scale)	(5-, 10-, 25-ft) spot elevations 100/500-yr. flood limits
Planimetric	Boundaries Survey monuments Buildings Land cover Land uses & management Safety zones	Boundaries Survey monuments Roads, sidewalks buildings streams trees (all misc. ground features i.e., golf course, helipads) Safety zones Text (100 & 400 scale)
Utility systems	Vectors only for reference (infrequent)	Valves, hydrants, poles Text (100 & 400 scale)
Natural resources	Streams & watersheds Soils Wetlands Wildlife habitats	Streams Soils (infrequent) Wetlands Wildlife habitat (infrequent)
Cultural resources	Archaeology Sites and areas and buildings	Sites, areas, buildings
Predictive models	Erosion	No predictive modeling

Table 6

Updating Capabilities

GRASS	IGDS CADD
Vector digitizing/editing	Vector digitizing/editing
Vector translation	Vector translation
Raster digitizing	
Raster to vector	Raster to vector
Vector to raster	Vector to raster
Raster to display	Raster to display (terrain)

16 DOCUMENTATION OF IGDS DATA INTENDED FOR TRANSLATION

Specific documentation of Intergraph vector data (IGDS) is necessary to convert IGDS files to the GRASS vector format (*dig*). The documentation specifications are:

1. If the IGDS data is being relocated to a GRASS machine via magnetic media or optical media, a printout listing the tape or disk contents is suggested. If the files are written in VAX VMS backup format, this printout should have the *saveset* name(s) on it as well as the tape *label*. IGDS data may also be transferred to a GRASS machine using network software, or may already reside on a GRASS machine when Intergraph Interpro hardware is in use. In these instances, listing the contents of the UNIX directory is sufficient to view the file names and sizes.

2. The naming convention for the IGDS files should be documented together with a diagram showing file geographic coverage to help identify the files.

3. The kinds of thematic data represented in each file should also be indicated by: (1) listing the graphic themes (for example, primary roads, secondary roads, bridges, and trails), and (2) using a digital or hard copy legend to identify the line styles, symbols, and colors that represent the themes.

4. The design file level where each graphic theme resides should be documented per design file.

5. The map and photographic source(s) used in digitizing each IGDS map and the scale(s) of the sources and their dates should be identified.

6. The *monument* or *benchmark* symbology must be documented, if applicable.

7. The documentation must indicate the target hardcopy scale of the digital mapping, the coordinate system in which the data were digitized, and the units of the coordinate system.

8. The Intergraph *working units* the data were digitized in must be indicated: the master unit (mu), the subunits per master unit (su), and the positional units per subunit (pu).

9. The latitude and longitude of the SW, SE, NE, and NW corners of the entire map area must be indicated if using *DLGOUT*. For example, if the entire area includes 43 IGDS facets, the four corners of the entire map would be identified as a point outside the SW corner of the SW facet, a point outside the SE corner of the SE facet, a point outside the NE corner of the NE facet, and a point outside the NW corner of the NW facet. These corner coordinates should also be provided in the coordinate system in which the data were digitized (e.g., UTM or State Plane).

10. A hardcopy map showing the facetized map sheet grid in the scale in which the data were digitized (e.g., 1 in.= 400 ft, 1 in.= 100 ft) is required. The geographic boundary for the installation or map area and the location of town or cantonment areas should be depicted on this map.

11. If Intergraph VAX DMRS database files accompany the data, a .DDL file and a .DBS file must be delivered on the tape or disk containing the attribute data.

12. The IGDS *global origin* for the database must be documented.

13. A point of contact for the map project and the office of the contractor who generated the Intergraph data should be provided.

17 GRASS-TO-INTERGRAPH VECTOR TRANSLATION PATHS

There are currently two vector translation paths from the GRASS vector *dig* format to Intergraph's IGDS vector format, due to the export capabilities of GRASS. One path uses the USGS *DLG* format as an intermediary format, and the other path uses the CAD *DXF* format as an intermediary format. The *DXF* or *DLG* file is then imported into the IGDS environment using IGDS import capabilities. Figure 6 shows the two translation paths.

The current GRASS vector export capabilities are:

1. DLG-3 Optional Format
2. DXF
3. ARC
4. MOSS-AMS.

The GRASS-to-IGDS translation programs that have been evaluated are the two Intergraph IGDS import programs *DLGIN* and *dxfin*, and the two GRASS export programs *v.out.dlg* and *v.out.dxf*. *DLGIN* runs on the Intergraph VAX platform only, and *dxfin* is an Intergraph MicroStation 32 and PC MicroStation-based product. The two GRASS export programs, *v.out.dlg* and *v.out.dxf*, are executed within GRASS, and therefore run on all of the GRASS hardware platforms, such as the Masscomp, Sun, and Intergraph Interpro workstations (see the GRASS *Hardware Configuration Guide*).

Note that v.out.dxf has not been updated recently. The user should test v.out.dxf to ensure that it produces suitable dxf files.

```
GRASS vect -----------------> DLG -------------------> IGDS
              (v.out.dlg)              (DLGIN)

GRASS vect -----------------> DXF -------------------> IGDS
              (v.out.dxf                (dxfin)
```

Figure 6. GRASS-to-IGDS Translation Paths.

18 CHOOSING *DLG* OR *DXF* AS AN INTERMEDIARY VECTOR FORMAT

The decision to use either *DLG* or *DXF* as an intermediary format is based on the capabilities of the GRASS export programs *v.out.dxf* and *v.out.dlg*,and the capabilities of the Intergraph import programs VAX *DLGIN* and MicroStation *dxfin*.

GRASS Export Programs

The output file created by *v.out.dxf* can be either larger or smaller than the output file created by *v.out.dlg* for the same GRASS *dig_ascii* file (Table 3). Both GRASS programs output the files in ASCII format. (Chapter 20, "How To Export *DLG* and *DXF* Files From GRASS," p 51).

DLGIN

DLGIN is the Intergraph translator that translates USGS *DLG* data files to Intergraph Standard Interchange Format (ISIF) ASCII data files. The Standard *DLG*-3 distribution format and the Optional *DLG*-3 distribution format are supported. The translator runs only on *DLG*-3 format data with the scales of 1:2000000, 1:24000, or 1:00000. The *DLG* ASCII files can be translated into Standard IGDS design files using the Intergraph Standard Interchange Format (ISIF) software. (Refer to the *Intergraph USGS DLGIN* User's Guide). *DLGIN* has the option to translate up to 100 *DLG* files at a time, placing each in a separate design file. *DLGIN* also has the capability to create an Intergraph database DMRS.

The choice between vector translation paths when translating data from GRASS to IGDS depends on the available hardware and whether the user would like to identify the GRASS category attributes as feature codes in the *DLG*-3 file, thus assigning them to ISIF attributes, and IGDS attributes with the associated IGDS database (DMRS). The creation of two input parameter files are required by *DLGIN*, and a third parameter file if a DMRS database is desired.

dxfin

The Intergraph MicroStation translator *dxfin* translates *DXF* ASCII format to IGDS binary vector format. There is no option to assign the *DXF* levels, element symbology, or element type to an IGDS level, element symbology, or element type. *DXF* levels translate into the corresponding IGDS levels. This program is simple to use since no parameter files are required. This program is the best choice if translation of database attributes is not desired. (Refer to the *Intergraph MicroStation Reference Guide*.)

Chapter Summary

The output file created by *v.out.dxf* may be larger or smaller than the output file created by *v.out.dlg* (Table 6). GRASS export file size may not be a criterion for choosing the export program. The main difference between the Intergraph programs *DLGIN* and *dxfin*, for translation decisions, is *DLGIN*'s ability to translate GRASS category attributes in the *DLG*-3 file to the ISIF format. One important factor is that translating the ISIF vector data into IGDS vector data requires an additional Intergraph *ISIF* program.

DLGIN is recommended if VAX hardware and software are available and if the user wishes to retain the GRASS *dig* attributes during translation. *dxfin* is recommended in the absence of VAX hardware and software, and if translation of GRASS attributes is not a priority.

19 EXPORTING *DLG* AND *DXF* FILES FROM GRASS

How To Export *DLG* Files From GRASS

The GRASS program *v.out.dlg* translates GRASS binary vector files, called *dig* files, to the USGS *DLG* format called *DLG-3 Optional* format. The output *DLG-3 Optional* file is in ASCII format.

To run *v.out.dlg*, type "v.out.dlg" at the GRASS prompt. The user is prompted for the name of the GRASS binary vector file and the name of the resultant *DLG* file. When the program is complete, the new *DLG* file will reside under the directory "dlg" in the current GRASS LOCATION and MAPSET. (Refer to the GRASS *Reference Manual* for further details.)

Summary of the GRASS-*DLG*-IGDS Translation Sequence

The procedure for translating GRASS vector data into IGDS vector format using the GRASS *v.out.dlg* program and the Intergraph VAX *DLGIN* program, to produce an ISIF vector file and then an IGDS vector file is:

1. Use the GRASS program *v.transform* to translate the UTM *dig_ascii* file to a State Plane *dig_ascii* file (see Chapter 11, "Coordinate Conversion in GRASS," p 39).

2. Run the GRASS programs *v.in.ascii* and *v.support* on the output file of *v.transform* to create a binary *dig* file with topology.

3. Run the GRASS program *v.out.dlg* to convert the *dig* file into an *ASCII DLG*-3 Optional formatted file.

4. Transfer the ASCII *DLG*-3 file to an Intergraph VAX computer that has the Intergraph programs *DLGIN* and ISIF installed.

5. Execute *DLGIN* (refer to Chapter 19, "Importing *DLG* and *DXF* Files into IGDS," p 53). The output file will be in the ISIF.

6. Run the Intergraph VAX-based program named *ISIF* to convert the ISIF file to an IGDS file.

7. Display the IGDS file in the Intergraph VAX or MicroStation display graphics environment.

How To Export DXF Files From GRASS

The GRASS program *v.out.dxf* translates GRASS ASCII vector files, called *dig_ascii* files, to the ASCII CADD *DXF* format. The program is a command line program in which the user types the program name followed by the required arguments. The syntax for the program is:

```
GRASS> v.out.dxf input = name output = name
```

where:

input = vector ASCII input file
output = dxf output file.

The following example uses the GRASS Spearfish data base:

```
GRASS> v.out.dxf input  =  /data/foghorn/spearfish/PERMANENT/dig_ascii/roads.ascii
                 output =  roads.dxf
```

The output file (in this case roads.dxf) is placed in the current working directory. As with all GRASS programs, the syntax of the program can be displayed on the screen by typing the program name and *help* at the GRASS prompt and pressing *Return*. For further information consult the GRASS *Reference Manual.*

Summary of the GRASS-DXF-IGDS Translation Sequence

The procedure for translating GRASS vector data into IGDS vector format, using the GRASS *v.out.dxf* program and the Intergraph MicroStation *dxfin* program to produce an IGDS vector file is:

1. Use the GRASS program *v.transform* to translate the UTM *dig_ascii* file to a State Plane *dig_ascii* file (see Chapter 11, "Coordinate Conversion in GRASS," p 39).

2. Run the GRASS program *v.out.dxf* to convert the *dig_ascii* file to an ASCII *DXF* file (see "How to Export DXF Files From GRASS," p 51)

3. Transfer the ASCII *DXF* file to an Intergraph machine running MicroStation PC or MicroStation 32, if it is not already on one (GRASS runs on Intergraph Interpro 200, 300, 3000, and 6000 series workstations. These workstations also run MicroStation 32).

4. Execute *dxfin* (refer to Chapter 21, "Importing DLG and DXF Files into IGDS," p 53). The output file will be in the Intergraph Graphics Design Software (IGDS) format.

5. Display the IGDS file in the Intergraph VAX or MicroStation display graphics environment.

How To Run Intergraph *DLGIN*

The Intergraph translation program *DLGIN* runs on the Intergraph VAX platform. After exporting the GRASS *dig* file to an ASCII *DLG* file, the *DLG* file can be networked or transferred to the VAX via magnetic or optical medium and imported into IGDS using the *DLGIN* program (see Chapter 20, "Exporting *DLG* and *DXF* Files From GRASS," p 51). The full translation path using Intergraph *DLGIN* will be:

GRASS dig -----> *DLG* -----> ISIF -----> IGDS

Two files are required to run VAX *DLGIN* on a *DLG* file. Both files are created by the user using a screen text editor and are stored in a flat ASCII file. The first file is an ISIF Environment File. This is the control file which contains specifications for ISIF processing. The *Intergraph DLGIN* Translator User's Guide does not describe this file, but refers the reader to the *Standard Interchange Format Command Language Implementation User's Guide*[14] for details.

The second file is a *DLGIN* parameter file. It can be given any name by the user, for example, *dlg.param.* The full name in VAX VMS will then become *dlg.param.;1* Underbars should not be used in the filename because this may cause confusion in the VMS operating system. The user will be prompted for the parameter filename when running *DLGIN*, and should type the full VMS filename, including the semicolon followed by the version number.

The parameter file allows the user to specify the IGDS element type, level, color, weight, and linestyle to be associated with the *DLG* lines, areas, and nodes, and then to be placed in the IGDS design file. If *DLG* nodes, areas, or lines are not given IGDS element characteristics in the parameter file, defaults are assigned. For a detailed description of the parameter file and a list of *DLGIN* defaults, refer to the *DLGIN* Translator User's Guide.

DLGIN also has the capability to create IGDS attributes from a *DLG* file that is stored in the Intergraph VAX-based DMRS data base. This option of *DLGIN* was not evaluated in this study.

To execute the *DLGIN* translator, at the VAX prompt, type: "run pro_dd_dlgin:dlgin" and then follow the interactive prompts described in the *Intergraph DLGIN* User's Guide.

How To Run Intergraph *dxfin*

dxfin is the Intergraph MicroStation-based program that translates *DXF*-formatted vector data to IGDS vector format. *dxfin* is a program that comes packaged with Intergraph MicroStation PC and Intergraph MicroStation 32.

The first step before running dxfin is to create an IGDS design file. The design file should be given the same name as the *DXF* file. For example, if the *DXF* file is named *roads.dxf*, the design file should be given the name *roads.dgn*. If the design file is created on a VAX, create the file in the VAX home

[14] *Standard Interchange Format Command Language, Implementation Guide*, DTRN001 (Intergraph Corp., 1986).

directory. To create a new design file, either on a VAX or within MicroStation, type the following sequence of commands:

$ ice (VAX) or mce (MicroStation)
GRAPHICS: uti cre filename 2d 150
UTILITIES: g filename.dgn

These commands create a two-dimensional design file having the name *filename* with 150 blocks. IGDS working units (MU, SU, and PU) and the global origin (go=) should be set as well according to the scale and desired coordinate system (refer to the *IGDS User's Guide*).[15] To exit the design file, select the *Exit* interface option in MicroStation or the *Utilities* pop down menu on the VAX. Next, when on the VAX, select the "File Design" option and then press <control> z <return>. If the file was created on a VAX, transfer the file to a MicroStation platform through the network.

To execute *dxfin*, type "mce" at the $ prompt and enter the supplement library as follows:

$ mce
GRAPHICS: uti
UTILITIES: supp
SUPPLEMENT: dxfin

You will be prompted for the *DXF* file name. When the program is complete the design file can be displayed in either the MicroStation or VAX IGDS environment. (For additional information refer to the *Intergraph Microstation Reference Guide*).

[15] *IGDS User's Guide*, DNUC012 (Intergraph Corp., 1985).

REFERENCES

Bradshaw, Stuart, and Pam Thompson, *Options for Acquiring Elevation Data*, Technical Manuscript (TM) N-89/20/ADA2200934 (U.S. Army Construction Engineering Research Laboratory [USACERL], January 1989).

Brooks, Douglas, Michael Shapiro, and Mark Johnson, *GRASS Hardware Configuration Guide*, Automated Data Processing (ADP) Report N-89/21/ADA220954 (USACERL, March 1989).

EDG User's Guide, DSYS103 (Intergraph Corporation, 1987).

IGDS User's Guide, DNUC012 (Intergraph Corporation, 1985).

Intergraph DLGIN User's Guide, DMAP14410 (Intergraph Corporation, 1986).

Intergraph DLGOUT User's Guide, DMAP145 (Intergraph Corporation, 1986).

Intergraph MicroStation Reference Guide, DSA027110 (Intergraph Corporation, 1989).

Intergraph Spatial Editor User's Guide, DMAP16520 (Intergraph Corporation, 1988).

Messersmith, Jean, "Map Preparation," in *GRASS Reference Manual*, by James Westervelt, et al., ADP Report N-87/22 (USACERL, September 1988).

Shapiro, Michael, et al., *GRASS Programmer's Manual*, ADP Report N-89/14 (USACERL, September 1989).

Standard Interchange Format Command Language Implementation Guide, DTRN001 (Intergraph Corporation, 1986)

U.S. Department of the Interior, U.S. Geological Survey, *Digital Line Graphs from 1:24000-Scale Maps* (Department of the Interior, 1986).

Westervelt, James, et al., *GRASS Reference Manual*, ADP Report N-87/22 (USACERL, September 1988).

USACERL Distribution

Chief of Engineers
ATTN: CEHSC-IM-LH (2)
ATTN: CEHSC-IM-LP (2)
ATTN: CERD-L
ATTN: DAEN-ZCI-P (2)
ATTN: CECW-PF
ATTN: CECW-EP-S
ATTN: CECW-RE
ATTN: CEIM-P
ATTN: CECW-OR

US Army Engineer District
ATTN: Chief, Regulatory Functions
Buffalo 14207
Norfolk 23510
Huntington 25701
Wilmington 28402
Charleston 29402
St. Paul 55101
Chicago 60606
Little Rock 72203
Galveston 77550
Albuquerque 87103
Los Angeles 90053
San Francisco 94105
Sacramento 95814
Portland 97208
Seattle 99362
New England 02254
ATTN: CENED-OD-R
New York 10278
ATTN: CENAN
Philadelphia 19106
ATTN: CENAP
Baltimore 21203
ATTN: CENAB-OP-R
Savannah 31402
ATTN: CESAS-OP-FP
Jacksonville 32232
ATTN: CESAJ-R-RD
Mobile 36628
ATTN: CESAM-OP-S
Nashville 37202
ATTN: CEORN
Memphis 38103
ATTN: CELMM-CO-R
Vicksburg 39180
ATTN: CELMV
Louisville 40201
ATTN: CEORL
Detroit 48226
ATTN: CENCE-CO
Rock Island 61204
ATTN: CENCR
St. Louis 63101
ATTN: CELMS-OD-F
Kansas City 64106
ATTN: Chief, Permits Section
Omaha 68102
ATTN: CEMRO-OP
New Orleans 71060
ATTN: Chief, Permits Section
Tulsa 74121
ATTN: CEWST
Fort Worth 76102
ATTN: CESWF-OD-O

US Army Engr Divisions
North Central 60606
ATTN: CENCD-CO
Southwest 75242
ATTN: CESWD-CO-R

US Military Academy 10996
ATTN: Dept of Geo & Envr Engr (2)
ATTN: Natural Resources Branch

Aberdeen Proving Ground, MD 21005
ATTN: ISC-TECOM
ATTN: HSHB-CI

US EPA Research Lab 97333

Yakima Firing Center 98901

USATHAMA 21010
ATTN: CETHA-RM-I
ATTN: AMXTH-RM
ATTN: CETHA-IR-S

CEHSC 22060
ATTN: CEHSC-FN
ATTN: CEHSC-SP
ATTN: CEHSC-FM-A

Bureau of Land Management
WASH DC 20240
Fort Collins 80526
Denver 80225

US Dept of Commerce 20233

Army National Guard 20310
ATTN: NGB-AREC

US Army Concepts Analysis Agency 20814

FBI Academy 22135

USA Foreign Science Tech Ctr 22901

Naval Oceanographic Office 39522

Redstone Arsenal 35898
ATTN: AMSMI-RA-EH-MP-PC

CEWES
ATTN: CEWES-IM-DA 39181
ATTN: CADD Center 39180

Wright-Patterson AFB 45433
ATTN: NSBIT AL/OEBN

Michigan Dept of Military Affairs 48913

Twin Cities Army Ammo Plant (2) 55112

Camp Ripley 56345
ATTN: Ofc of Archeology & Engr (2)

5th Inf. Fort Polk 71459
ATTN: Envr & Nat Rsrcs Mgmt Div (2)

US Army Materiel Cmd 61299
ATTN: AMXEN-U (2)

US Army Europe
HQ USAREUR 09403
ATTN: AEAEN-FE-E (2)
V Corps 09079
ATTN: AETV-EHF-R

Texas Army Nat'l Guard 78763

Lone Star Army Ammo Plant 75505

Red River Army Depot
ATTN SDSRR-GB 75507

Dugway Proving Ground 84122
ATTN: DPG-EN-E (2)

Yuma Proving Ground 85365
ATTN: ATEYP-ES-E

White Sands Missile Range
ATTN: STEWS-ES-E 88002

Envr Response & Info Ctr
ATTN: ENVR-EP 20310

Nat'l Geophysical Data Ctr
ATTN: Code E-GCI 80303

Hohenfels Training Area 09173
ATTN: AETTH-DEH
ATTN: AETTH-DEH-ENV-APO

US Army Forts
Fort Belvoir, VA 22060
ATTN: CEETL-CL-GC (2)
ATTN: CETEC-CA-D
ATTN: AMSEL-RD-NV-VMD-TST
ATTN: Envr & Nat Res Div
Fort Monroe, VA 23651
ATTN: ATBO-GE
ATTN: ATEN-FN
Fort Drum 13603
ATTN: AFZS-EH-E
Fort Jackson 29207
ATTN: ATCJ-EHN
Fort Gillem 30050
ATTN: FCEN-CED-E
Fort Gordon 30905
ATTN: ATZH-DIE (2)
Fort Stewart 31314
ATTN: AFZP-DEN-W
Fort Benning 31905
ATTN: Nat. Resource Mgmt Div (2)
Fort McClellan 36205
ATTN: ATZN-FEE
Fort Rucker 36362
ATTN: ATZQ-EH
Fort Knox 40121
ATTN: ATZK-EHE
Fort Campbell 42223
ATTN: AFZH-DEH
Fort Benjamin Harrison 46216
ATTN: ATZI-ISP (2)
Fort McCoy 54656
ATTN: AFZR-DEN
Fort Riley 66442
ATTN: AFZN-DE-N (2)
Fort Chaffee 72905
ATTN: ATZR-ZFE (2)
Fort Sill 73503
ATTN: Fish & Wildlife Br (2)
Fort Leonard Wood 65473
ATTN: ATZT-DEH-EE
Fort Dix 08640
ATTN: ATZD-EHN
Fort Eustis 23604
ATTN: Ranges & Targets Dir
Fort Worth 76115
ATTN: Cartographic Ctr (2)
Fort Hood 76544
ATTN: AFZF-DE-ENV
Fort Bliss 79916
ATTN: ATZC-DEH-E
Fort Carson 80913
ATTN: AFZC-ECM-NR
Fort Huachuca 85613
ATTN: ATZS-EHB
Fort Irwin 92310
ATTN: AFZJ-EH
Fort Lewis 98433
ATTN: AFZH-DEQ
ATTN: ATZH-EHQ
Fort Richardson 99505
ATTN: DEH
Fort Bragg 28307
ATTN: DEH

National Weather Service 20910

US Geological Survey 22092

Pine Bluff Arsenal 17602
ATTN: SMCPB-EMB

US Army Cold Regions Research & Engr Lab
ATTN: CECRL-IS 03755

NASA/SSC/STL 39529

Defense Technical Info Center
ATTN: DTIC-FAB (2)

144
+17
4/92