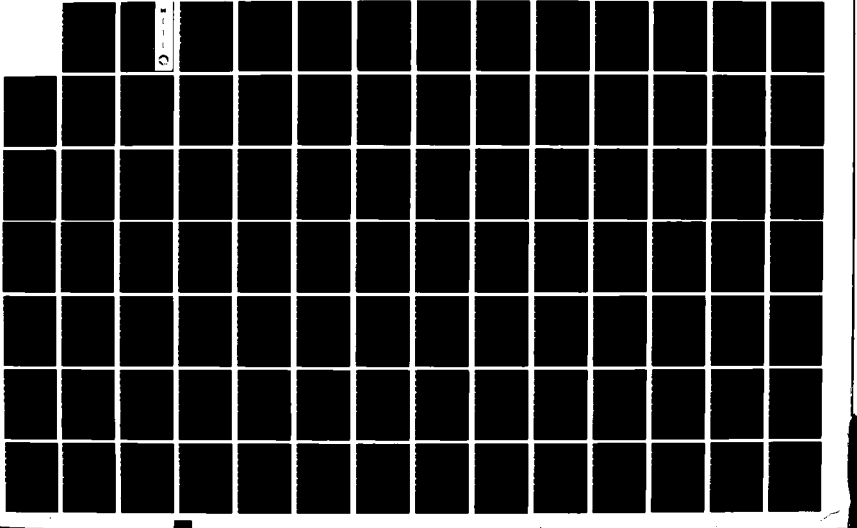
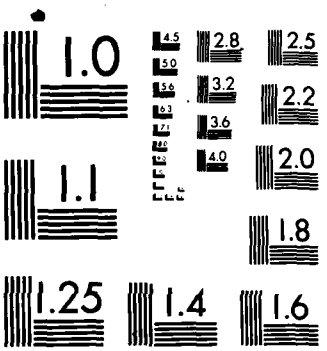
AD-A161 086 SOFTWARE CONVERSION OF STANDARD LINEAR FORMAT (SLF) TO STANDARD INTERCHANGE FORMAT (SIF)(U) MEASUREMENT CONCEPT CORP ROME NY K J SANDS ET AL. JUL 85 ETL-0394 1/2

UNCLASSIFIED DAAK70-83-C-0174 F/G 8/2 NL





ETL-0394

Software conversion of standard linear format(SLF) to standard interchange format(SIF)

Kathryn J. Sands

Measurement Concept Corporation
1721 Black River Boulevard
Rome, New York 13440

July 1985

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

E T L

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
ETL

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ETL-0394	2. GOVT ACCESSION NO.: AD-A161086
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): Software Conversion of Standard Linear Format (SLF) to Standard Interchange Format (SIF)	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report September 1983 - July 1985
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Kathryn J. Sands Kim D. Headlee	8. CONTRACT OR GRANT NUMBER(s): DAAK70-83-C-0174
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Measurement Concept Corporation 1721 Black River Blvd. Rome, NY 13440	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546	12. REPORT DATE: July 1985 13. NUMBER OF PAGES: 150
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Defense Mapping Agency Standard Linear Format
Intergraph Corporation Standard Interface Format
Graphic Symbolization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Software accepts a digital cartographic data set in DMA Standard Linear Format as input and generates graphics and attribute data base information in Intergraph Corporation Standard Interchange Format. Cartographic feature information is examined to derive graphic symbolization characteristics. Validation of inputs ensures the quality of the output. Additional software provides the capability to convert from Intergraph Standard Interchange Format to DMA Standard Linear Format.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document was prepared under contract DAAK70-83-C-0174 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, by Measurement Concept Corporation of Rome, New York. The Contracting Officer's Representative was Ms. Betty Mandel.

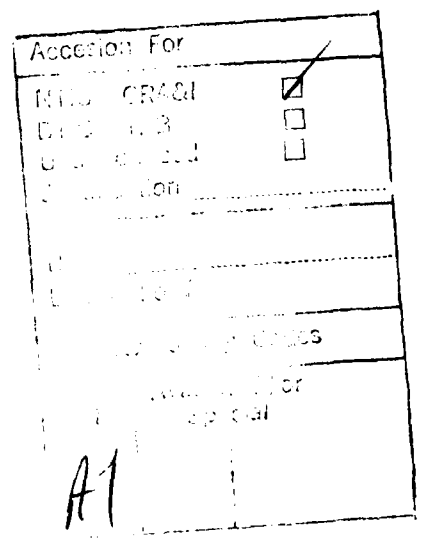


TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF FIGURES

SECTION 1
INTRODUCTION

1.1 Purpose of the Report

This Final Technical Report for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF), Contract Number DAAK70-83-C-0174 summarizes the objectives of the effort and presents a brief discussion of the software. Conclusions discuss limitations and recommendations for further research.

1.2 Purpose of the Conversion Software

The Conversion Software was developed by Measurement Concept Corporation (Mc^2) under contract to the US Army Engineer Topographic Laboratories (USAETL) for the Defense Mapping Agency (DMA). The software provides a basic reformatting capability and supports data transfer between systems as part of the Carto Data Validation Project research activity within the USAETL. To ensure symbolization and data verification capabilities of the software, the effort was extended to include Software Conversion from Intergraph Corporation Standard Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF).

1.3 References

The following references were used throughout the effort and in preparation of this report:

- Statement of Work for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF), ETL-TD-MA, 30 June 1983.

- Statement of Work for Software Conversion of Intergraph Corporation Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF), ETL-TD-MA, 21 March 1984.

- Defense Mapping Agency Product Specifications for 1:50,000 Scale Topographic Maps of Foreign Areas, PS/3AA/101, July 1980.

- Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog, July 1977.

- Defense Mapping Agency Standard Linear Format (SLF) for Digital Cartographic Feature Data, 16 May 1983.

- Intergraph Standard Interchange Format (SIF) Command Language Implementation, DIXD1300, 30 May 1982.

- Intergraph Standard Interchange Format (SIF) Users Guide, DIXD0500, 15 November 1980.

- Intergraph Interactive Graphics Design System (IGDS) Operating Manual (IGDS8), 79-067C, 13 August 1981.

- Intergraph DMRS Data Definition Language Compiler User Manual, 79-050C, 20 June 1981.

- Intergraph DMRS 8.3 Command Language Users Guide, 79-065C, 7 July 1981.

- SLF to SIF Software Conversion Working Papers, Measurement Concept Corporation, December 1983, April 1984, November 1984, February 1985.

- SLF to SIF Conversion Software Users Manual, Measurement Concept Corporation, July 1985.

- SLF to SIF Conversion Software Program Maintenance Manual, Measurement Concept Corporation, July 1985.

1.4 Terms and Abbreviations

The following terms and abbreviations are used in this report:

DMA	Defense Mapping Agency
DMRS	Data Management and Retrieval System
IGDS	Interactive Graphics Design System
Mc^2	Measurement Concept Corporation
SIF	Standard Interchange Format
SLF	Standard Linear Format
USAETL	US Army Engineer Topographic Laboratories

1.5 Summary

Section 2 of this report describes the capabilities, logical flow and components of the Conversion Software. Section 3 presents conclusions regarding performance and limitations and recommends areas for further investigation. The Users Manual, Program Maintenance Manual and Message Definition Document are included as Appendices.

The Conversion Software provides the capability to convert digital geographic/cartographic data in Defense Mapping Agency Standard Linear Format (SLF) to digital cartographic/graphic data in Intergraph Corporation Standard Interchange Format (SIF). Input SLF feature attributes are used to define graphic attributes. Full graphic definitions are provided for 60 features. The capability to convert data from Intergraph Corporation Standard Interface Format (SIF) to defense Mapping Agency Standard Linear Format (SLF) is also supported.

The software is written in VAX FORTRAN and executes on the Digital VAX Family of processors under the VMS Operating System. Structured techniques and current software engineering technologies were applied during the design and implementation of the Conversion Software. The result is a modular organization which will facilitate software maintenance and enhancement.

SECTION 2
SYSTEM DESCRIPTION

2.1 Capabilities

The Conversion Software accepts a digital cartographic data set in DMA Standard Linear Format as input and generates graphics and attribute data base information in Intergraph Corporation Standard Interchange Format. Optionally, cartographic feature information is examined to derive graphic symbolization characteristics. Validation of inputs ensures the quality of the output.

The software is written in VAX FORTRAN and executes on the Digital VAX family of processors under the VMS Operating System. DMA Standard Linear Format (May 1983) Digital Auto-Carto Data (DACD) and Digital Feature Analysis Data (DFAD) product types are supported for feature symbolization assignment. Mechanisms are provided to establish and modify a Symbology Correlation Library; alternate libraries may be specified. The Standard Interchange Format generated is compatible with the Interactive Graphics Design System (IGDS), Version 8.5, executing on the Digital PDP-11/70 under the RSX Operating System.

The extension to the software generates an SLF data set from cartographic data digitized on the Intergraph System. Data relationships, feature attributes and Data Set Identifier information is established in an Intergraph Data Management and Retrieval System (DMRS) data base and is extracted by the Conversion Software. Symbolization information present in the input SIF is ignored.

Both the conversion from SLF to SIF and the conversion from SIF to SLF are integrated into a single package and invoked via a common user interface. Both formats may be transferred between magnetic tape and disk. All conversion is performed against files resident on disk.

2.2 System Logical Flow

Processing steps for conversion from SLF to SIF are illustrated in Figure 2-1 and include input of SLF from tape, conversion from SLF to internal format, conversion from internal format to SIF and output of SIF to tape. The output SIF tape is input to the Intergraph system using the Intergraph SIF Processors; Binary Tape Input (BTI) is used to input the SIF file from tape and Translator-In (TRI) is used to convert the SIF to Intergraph internal IGDS/DMRS format.

A Symbol Correlation Library is used for conversion from SLF to SIF. The Library is constructed from a Symbol Correlation File source; a Library can be used for many conversion jobs. Multiple Symbol Correlation Libraries may be defined; the specific Library to be used is a run-time parameter. Each Library must contain symbolization information for standardized information, such as registration points, and the default and error symbology to be applied.

Processing steps for conversion from SIF to SLF are illustrated in Figure 2-2 and include input of SIF from tape, conversion from SIF to internal format, conversion from internal format to SLF and output of SLF to tape. The input data is constructed on the Intergraph system to create data in IGDS/DMRS format. The DMRS data base contains attribute information which enables the conversion software to derive SLF feature-segment relationships, as detailed in the Appendices. The Intergraph Translator-Out (TRO) and Binary Tape Output (BTO) SIF processors are used to convert the IGDS/DMRS data to SIF for input to the Conversion Software.

The same internal format is used for both conversion from SLF to SIF and from SIF to SLF. This enables conversion from SIF to SLF to SIF by performing the processing steps illustrated in Figure 2-3.

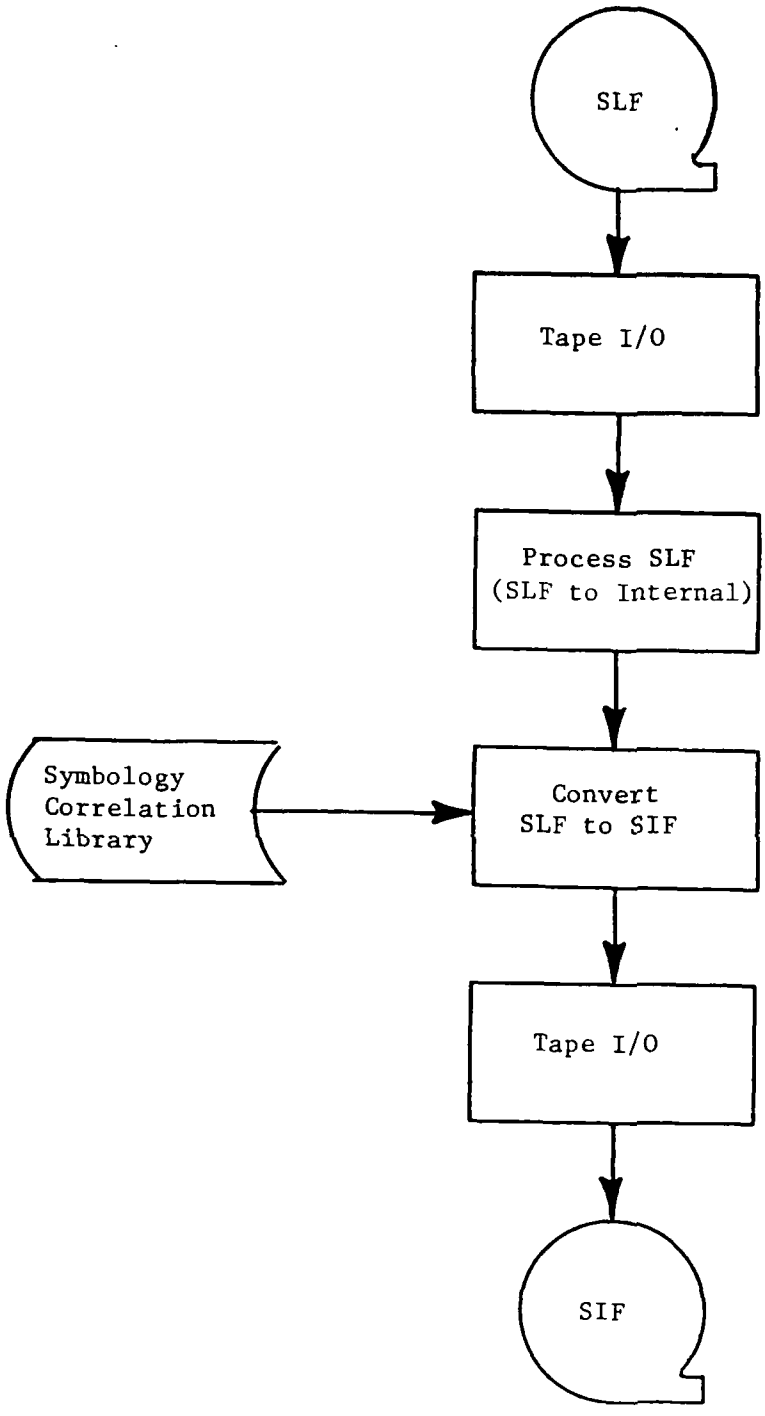


Figure 2-1 SLF to SIF Logical Flow

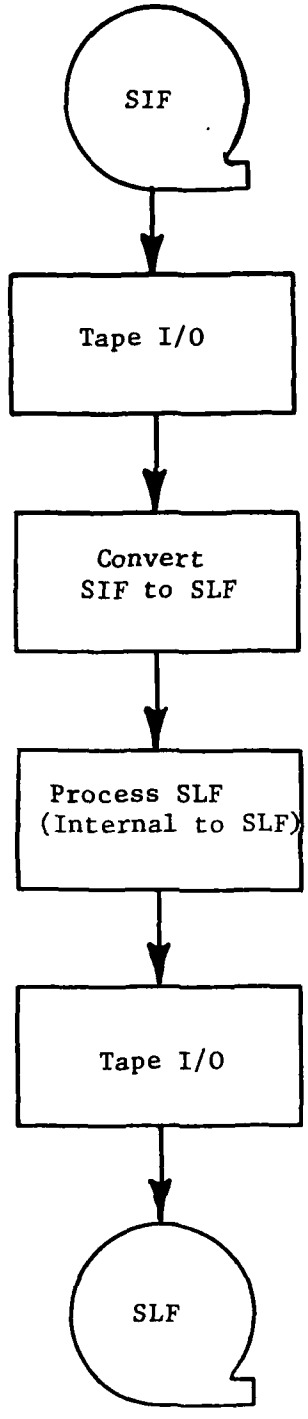


Figure 2-2 SIF to SLF Logical Flow

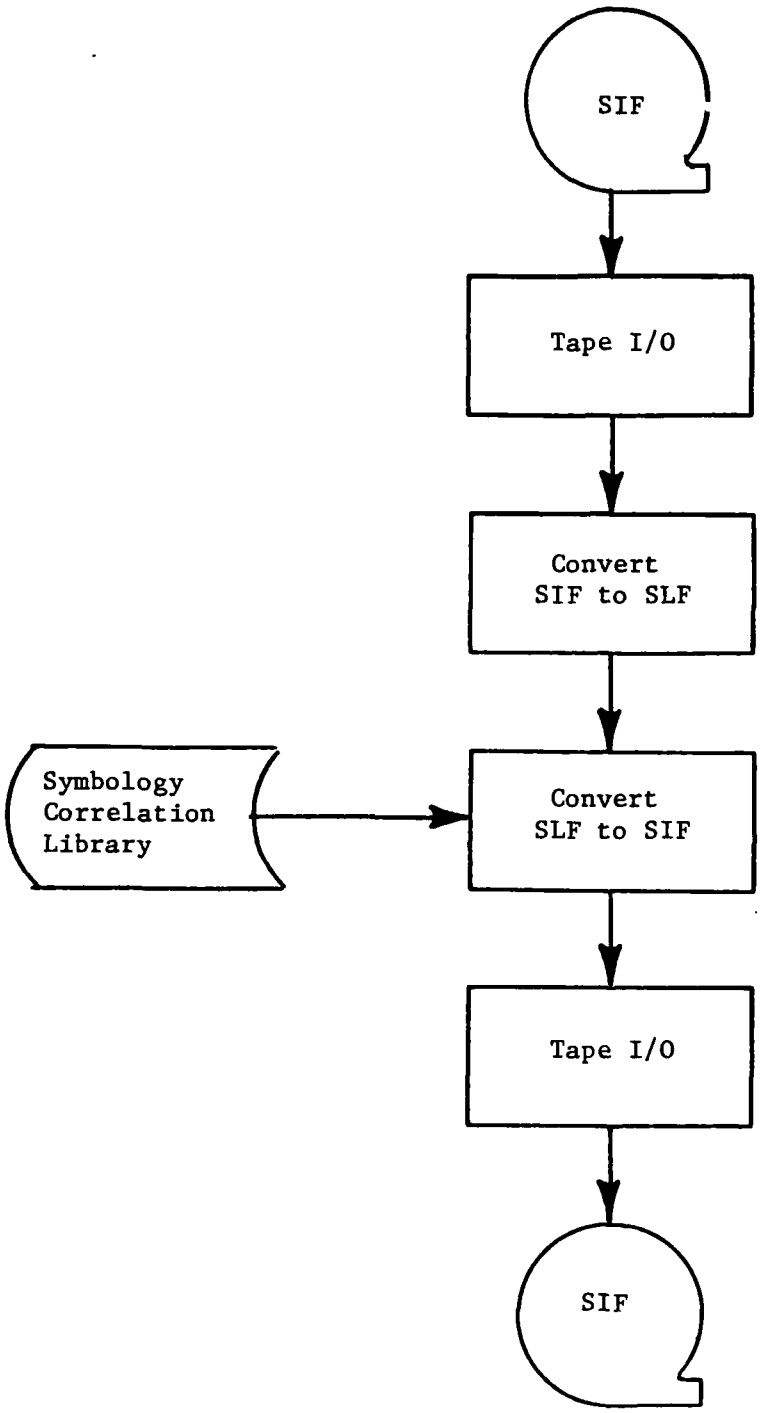


Figure 2-3 SIF to SLF to SIF Logical Flow

2.3 Components

2.3.1 Software

The Conversion Software consists of several major functions including:

- o Define/Submit Job
- o Build Symbol Library
- o Tape Input/Output
- o Process SLF
- o Convert SLF to SIF
- o Convert SIF to SLF

These processes are supported by generalized service routines for data management, magnetic tape functions and SIF command generation and output. The services are maintained in an object library and are available for general use. Most are written as functions and return a condition value status to the calling routine. The VMS Message Utility is used to define condition values for the Conversion Software which are identical in form and function to VMS condition values. A single service routine is used to report both Conversion Software and VMS condition values. Each service routine reports messages as errors are encountered and returns the appropriate condition value to the calling routine. This philosophy is followed throughout the Conversion Software to provide clean, straightforward error handling mechanisms.

Define/Submit Job is written as a VMS command procedure. It interfaces to the user to define a conversion job and to submit the job for execution. The user may select any of the conversion functions in any desired sequence. As a function is selected, appropriate job parameters are requested; default values are supplied as applicable. Once defined, the job is submitted to a VMS batch queue for execution. The user specified functions are executed in the defined sequence and a process summary report is generated either on the system printer or to a file.

Build Symbol Library converts an ASCII Symbology Correlation File into the Symbology Correlation Library format. The contents and format of the input is verified.

Tape Input/Output provides the capability to transfer data in either SLF or SIF between tape and disk. The disk file is an exact copy of the tape contents, excluding labels, end of file marks, etc. A data set may be input from an SLF tape which contains multiple data sets. Multi-reel volumes are not supported. The magnetic tape services are used to allocate a tape drive, logically mount the tape, and perform I/O to the tape. Mount request messages are generated on the operators console. Upon completion of the process, the tape is rewound and dismounted.

Process SLF converts a data set between SLF and internal format. The internal format maintains each record type (i.e., features, segments) in a separate file. Certain SLF information, such as feature ID, is retained internally as a numeric value; all SLF information is ASCII character data. The internal format includes additional information, such as the segment bounding rectangle, which is used by the conversion process. The Process SLF function will convert from SLF to internal format or from internal format to SLF. This conversion is performed sequentially against the input and includes record blocking/ deblocking, association of logical record types with separate files and conversion between numeric and character values.

Convert SLF to SIF converts a data set in internal format to SIF. The resultant SIF file consists of a series of SIF commands which instruct the Intergraph SIF Processors to construct data in IGDS/DMRS format. Commands are output to generate data set DSI, registration point descriptions and feature attributes as entity data in a DMRS data base. Commands to generate graphics are output for the DSI, each registration point and each segment. A DSI symbol is arbitrarily positioned at the last registration point. Input features are processed sequentially. For recognized SLF product types, the feature header information is extracted and used to

obtain symbolization information for the feature. SIF commands are output to establish graphic characteristics (e.g., color, line style) for the subsequent graphic elements. One or more graphic element generation commands are output for each segment owned by the feature. Data integrity is checked to ensure consistency of the output (e.g., closed areal features). An input segment owned by multiple features is output (as a series of graphic element commands) once for each feature to which it belongs, to enable generation of areal symbology. Segments in the internal format are stored in an indexed sequential file, allowing for random retrieval by segment ID.

Convert SIF to SLF converts input SIF to internal data set format. The input SIF file contains a series of SIF commands generated from IGDS/DMRS data by the Intergraph SIF processors. The input SIF is processed sequentially to generate the internal data set format. Each type of information input is recognized by attribute data followed by graphics data. The conversion software assumes that each segment is digitized only once and has associated attribute information for one to three features (i.e., to its left, to its right, and coincident). Segments are created as the graphics data (i.e., points) is input. Feature information is input once for each segment belonging to the feature. A temporary feature file is constructed as the input is processed; segments are recorded as part of a feature in the same sequence encountered in the input. Once the entire SIF file has been input, each temporary feature is processed to sequence the segments within the feature. Symbolization information and graphic data with no associated attribute information is ignored in the input SIF. Figure 2-4 lists the disposition for each type of input SIF command.

COMMAND	#	ACTION
Classification	22	Only elements of class 0 (primary) are processed
Identifier	30	Occurrence num = segment ID; Entity num = data type
Line String or Curve	40, 45	Segment points
Symbol - Cell	43	Origin = single point segment
Begin Complex String	50	Start of segment
End Complex String	51	End of segment
Begin Symbol	52	Ignore following elements
End Symbol	53	Stop ignoring elements
Continue	82	Continue current command
Associative Values	83	Build feature, reg point or DSI

All other SIF command types are ignored.

Figure 2-4 SIF Command Disposition

2.3.2 Data

The DMA Standard Linear Format (SLF) is designed as the vehicle for the exchange on magnetic tape of digital cartographic feature data, consisting of planimetric and/or hypsometric information created in the production of various DMA hydrographic, topographic and aeronautical products. The objectives of the SLF are:

- To store a string of data only once, no matter how many features it may be a part of.
- To provide a standard format for digital data exchange.
- To accommodate multiproduct and multiseries mapping and charting requirements.

Each SLF file represents a specific geographic area and is referred to as a data set. SLF data includes four basic types of information (i.e., logical record types): Data Set Identifier, Segments, Features and Free Text. The Data Set Identifier (DSI) occurs once per data set and is comprised of control information including identification, security, geographic parameters, history, registration, and accuracy information. Registration information consists of a series of control points described by geographic and Cartesian coordinates and elevation. Accuracy information is similarly a series of points which represent areas of accuracy. Free Text is ASCII format textual information optionally accompanying a data set.

Feature and Segment logical records comprise the remainder of a SLF data set. The SLF supports a many-to-many relationship between features and segments. A segment is a string of data points which describe a location on the earth's surface. One or more segments are logically grouped together to form a feature. A feature has associated attribute information which describes characteristics of the earth's surface as

applicable for that geographic area defined by its segments. Each feature consists of one or more segments. Each segment belongs to one or more features. A segment belonging to more than one feature is referred to as a common segment.

A data set includes one logical record for each segment and one logical record for each feature in the data set. The variable length segment record includes a unique segment number, a count of the number of features to which the segment belongs, the number of points and the data points. The data coordinates are stored as delta x,y (and optional z) values from the data set origin. The data may represent geographic or Cartesian reference frames. The variable length feature record contains a unique feature number, a type indicator (point, lineal, areal), a count of feature header records, a count of segments belonging to the feature, a variable number of header records, and an entry for each segment belonging to the feature. The format of the feature header varies by product type.

The Intergraph Standard Interchange Format (ISIF) provides a common mechanism for transmittal of graphics and associated data between systems. Magnetic tape serves as the medium for exchange of SIF files between systems. A SIF file consists of unformatted, sequential 1024 byte physical records. Each physical record is comprised of one or more logical records, or commands. A command ranges from 1 to 256 32-bit words in length and may continue from one physical record to the next. Each SIF command is identified by a numeric type and subtype.

SIF commands are grouped into four categories: graphic characteristics, graphic element generation, graphic text generation, and miscellaneous. Graphic characteristics commands are used to establish the level, color, line style, line weight, complex symbolization and DMRS attribute information for the graphic element definitions which follow. The characteristic(s) remain in effect until a subsequent graphic characteristic(s) command is issued.

Graphic element generation commands are used to output the many types of graphic elements supported by the IGDS, including cells, text, line strings and shapes. A line string may contain a maximum of 101 data points. Larger strings are stored as complex strings and may contain up to 32767 strings of 101 points each. A shape is a closed polygon with a maximum of 101 vertex points. A complex shape is comprised of up to 32767 line strings representing a closed polygon. A cell is basically a point symbol, and is a complex element which may include element types such as lines, line strings, circles, shapes, and text.

Several files are required on the Intergraph system for conversion from SIF to IGDS/DMRS format. A Seed Design File supplies default file header information and parameter settings (e.g., color table) and is used by the Intergraph software to create an IGDS design (i.e., graphics) file.

A Cell Library defines cell elements to be used for point symbology. Each cell output in the SIF data must be included in the Cell Library. A Pattern File describes patterns (i.e., complex symbology) which may be applied to lineal and areal features. Patterns are constructed from cell elements in the Cell Library. Each pattern referenced in the output SIF data must be present in the Pattern File.

The SIF Environment File contains parameter information for input to the SIF processors. Several parameters are static, such as byte storage information, and do not change from job to job. Other parameters, such as file names and Design File size, are dynamic and vary by individual job.

A Data Definition Language (DDL) file defines the structure and content of a DMRS data base, which exists as a Schema file and a series of entity files, one per record type. The generated DMRS data includes DSI, registration point and feature entities. Attributes for each feature entity occurrence (i.e., record) include feature ID, feature type, number of segments and feature header attribute values.

The format and contents of files internal to the Conversion Software are described in detail in the Program Maintenance Manual.

SECTION 3
CONCLUSIONS

3.1 Overview

The following conclusions summarize the performance and limitations of the SLF to SIF Conversion Software. Recommendations address applicability as a tool for alternate applications and as a software basis for future development as well as suggesting enhancements to the software.

3.2 System Performance

The following describes functional capabilities and throughput characteristics of the Conversion Software.

Prototype R&D software provides USAETL the capability to convert digital geographic/cartographic data in Defense Mapping Agency Standard Linear Format (SLF) to digital cartographic/graphic data in Intergraph Corporation Standard Interchange Format (SIF). SIF data is generated on disk and tape which is identical in form and function to SIF produced by Intergraph Corporation SIF processors and is fully capable of access and processing by Intergraph Corporation SIF processors. Input SLF Digital Cartographic Feature Data attributes, based on the indexing system defined in the Defense Mapping Agency Standard Cartographic Feature Digital Identification Catalog, are used to access the Symbol Correlation Library which defines graphic attributes. Full graphic definitions, providing draft symbolization in accordance with Defense Mapping Agency Product Specifications for 1:50000 Scale Topographic Maps of Foreign Areas, PS/3AA/101, are provided for 20 point, 20 line and 20 areal SLF features, as listed in Figure 3-1.

NUM	FEATURE	SYMBOL
201	Divided Highway	Solid red 1D
203	Hard Surface Road	Solid red 1
205	Loose Surface Road	Solid red 2
208	Track	Solid red 6
212	Rd (Under Construction)	Solid red ? C
225	Interchange	Solid red
227	Bridge w/superstruct	Cased solid white
235	Highway Tunnel	Cased dashed white
242	Single Track RR	Solid ticked white
243	Double Track RR	Solid double ticked white
248	Non-Operating RR	Dashed ticked white ABANDONED
267	RR Tunnel	Cased dashed white
301	Built-up Area	Solid white
304	Shanty Town	Solid white SHANTY TOWN
305	Building	Square white
306	Church	Church white
320	School	School white
321	Hospital - Large	Large hospital white
322	Hospital - Small	Circle with cross white
329	Destroyed building	Open square red DESTROYED
401	Misc Tower	Circled dot white TOWER
402	Chimney	Chimney white
403	Lighthouse	Star dot white
405	Windmill	Windmill over dot white
408	Radio Mast	Mast over dot white
416	Elevated Tank	Tank over dot white
419	Power Line	Long dash with pylon blue
424	Barbed Wire Fence	Dash with X white
433	Airfield	Plane in circle green
434	Aero Obstruction	Tower over dot green
438	Seaplane Base	Anchor in circle green
440	International	Longdash dash dash white
441	1st Admin Boundary	Longdash dash white
447	Military Reservation	Dash dot white
502	Control Point	Dot in triangle white
503	Bench Mark	Dot in triangle white BM
508	Spot Elevation	Dot 00000 white
512	Index Contour	Solid brown
513	Intermediate Contour	Solid brown
520	Columnar Rock	Solid with halftick brown
533	Sand	Solid yellow SAND
543	Snowfield	Dashed; dots greeen
544	Glacier	Dashed; dashed hachure green
552	Moraine	Solid yellow AP99
553	Pack Ice	Dashed green PACK ICE
525	Crevasse	Solid hachure brown

Figure 3-1 Symbology (Page 1 of 2)

NUM	FEATURE	SYMBOL
604	Lake	Solid green W
605	Pond	Dashed; solid hachure green
645	Salt Evaporator	Solid; dots green
658	Swamp	Solid; swamp grasses yellow
659	Peat Bog	Same as swamp with PEAT BOG
662	Rice Fields	Solid; rice symbols yellow
664	Land Subj to Innundation	Dashed; dashed hachure green
670	Wet Sand	Dashed; dots green WET SAND
701	Coniferous Trees	Solid; triangle trees white T
703	Mixed Wood Trees	Solid; triangle and dot trees white T
706	Scrub	Solid white S
807	Rocks awash	Solid white ROCKS AWASH
808	Exposed Wreck	Wreck white
813	Oil/Gas Rig	Tower over dot white OIL

Figure 3-1 Symbology (Page 2 of 2)

Additional software provides the capability to convert data from Intergraph Corporation Standard Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF). This capability was developed to provide a mechanism for generation of SLF data including appropriate feature classifications and test data cases for the SLF to SIF Conversion Software. Input SIF is created from Intergraph IGDS/DMRS format data and contains sufficient information to derive SLF feature-segment relationships and feature attributes.

DMA Standard Linear Format (May 1983) data for any product type can be processed by the SLF to SIF Conversion Software; default symbology is assigned to features for product types not recognized by the software.

A SLF data set can selectively be input from a magnetic tape containing multiple data sets. Optional z/elevation data in the input SLF is not output by the Conversion Software. A scale factor may be applied against the input SLF. This provides the capability to edit the resultant graphics on the Intergraph system at a finer resolution than that of the input SLF. This capability is intended to support more precise positioning of symbolization information (e.g., text labels) prior to output as a product.

The DMA SLF allows for data in geographic and Cartesian reference frames. The Conversion Software accepts data in any reference frame but does not perform any projection transformations. Numerous DMA systems include capabilities to perform rigorous projection transformations. Additionally, the Intergraph World Mapping System product performs projection transformations against IGDS design files.

The Conversion Software correlates feature classification information with symbol specifications to define the graphic characteristics to be produced. The actual symbolization function (i.e., generation and modification of data points) is performed by the Intergraph software.

Basic symbolization is present for all IGDS design file elements and includes color, line style, line weight, etc. More complex symbolization, such as railroad ticking and swamp area symbols, is created via Intergraph patterning. Single point symbols are represented by Intergraph cell type elements. As part of the development activity, an Intergraph Cell Library and Pattern File were digitized to provide symbolization as per Figures 3-2 and 3-3.

The Conversion Software was developed on a Digital Equipment Corporation VAX-11/780 processor under Version 3 of the VMS Operating System. Integration testing was performed on a VAX-11/750 under VMS Version 3 and the software was delivered to the VAX-11/750 under VMS Version 4. An initialization command procedure defines site-specific information (e.g., disk and tape device names) so that no software modifications are necessary to transport the software between VAX systems or Versions 3 and 4 of VMS. Resource utilization statistics were gathered during testing of the Conversion Software in both the 780/Version 3 and 750/Version 4 environments. Identical test data and test sequences were used. For the 750/Version 4 testing, FORTRAN compiler array bounds checks were disabled. Based on this, and Digital announcements describing significant performance enhancements resulting from improved FORTRAN compiler optimization under Version 4, shorter processing times were anticipated for the 750/Version 4 environment. Actual measurements, however, indicated that CPU utilization and throughput remained relatively constant and that the most impacting factor was overall load on the VAX. Unfortunately, it was not possible to perform the testing on a quiescent system. At both sites, a typical conversion from SLF on tape to SIF on tape consumed 30 - 90 seconds of CPU time and took 2 - 5 minutes elapsed time. The effect of system load and the impact of VMS priorities was evident on the 750/Version 4 system where a test requiring 5 minutes on a quiet system could consume up to 30 minutes during peak system loads.

Figure 3-2 Cell Library

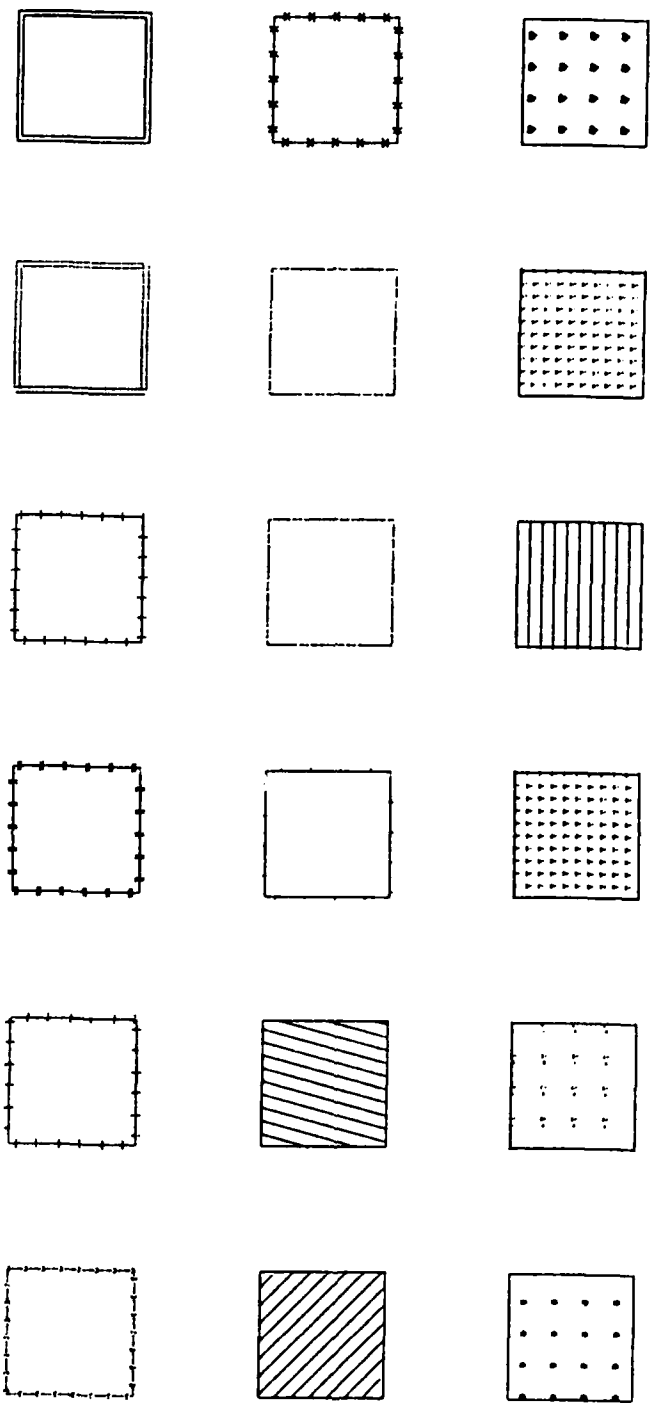


Figure 3-3 Pattern Library

Processing times for the Intergraph SIF processors on the USAETL PDP-11/70 were not measured. Tests were performed which executed the Intergraph Translator-In SIF processor against identical input SIF, both with and without generation of an associated DMRS data base. Creation of the DMRS data base appears to be the most significant factor affecting processing time for conversion from SIF to Intergraph formats. Pattern symbology is created by executing the Intergraph PDV task against an IGDS design file. Processing time requirements for PDV are highly dependant upon the characteristics of the input data and the density of the symbolization information. Performance considerations are addressed in the following sections.

3.3 Limitations

The Conversion Software has been designed to execute in the VAX/VMS environment. This has allowed for use of logical names to ensure device independence, adoption of coding standards which make use of VAX extensions to FORTRAN-77 (such as the IMPLICIT NONE statement), and software design to exploit indexed file capabilities of the VAX Record Management Services for enhanced performance. It is not anticipated that this limitation will impose significant constraints since USAETL and DMA have several VAX-based systems and Intergraph systems are offered primarily for the VAX family of processors.

Symbolization which can be generated is ultimately limited by the capabilities of the Intergraph system and the information provided in the input SLF. Based upon a review of Intergraph capabilities, the Conversion Software was designed to generate drafting symbols, as per the Defense Mapping Agency Product Specifications for 1:50000 Scale Topographic Maps of Foreign Areas, PS/3AA/101; it was decided that a mixture of draft and compilation symbology would be confusing to the user. An approximation is supplied for certain symbology (e.g., glacial form lines are not present in the input and are approximated by a hachure). The goal of

symbolization was to create a symbolized interim product of sufficient quality to support review and minor refinement prior to final product creation. The Conversion Software generates automatically positioned textual annotations. The resulting text positions appeared acceptable, although this may be an effect of the characteristics of the input SLF.

The SLF data structure supports a many-to-many relationship between segments (i.e., coordinates) and features (i.e., attributes). Common boundaries between areas are represented by a single string of points. Discrete point features (e.g., benchmark) are represented by a single coordinate. The IGDS/DMRS (and SIF) data structure represents graphic information as points, lines and areas. Each area is represented by the string of points forming its polygon; discrete points are symbolized (e.g., lines depicting a triangle). A one-to-many relationship is supported between graphics (IGDS) and attributes (DMRS). The Intergraph system performs symbolization against graphic (point, line, area) information. The Conversion Software assigns point, lineal or areal symbology to each input feature. Graphic element generation commands are output on a per feature basis; segment points are output (as a graphic element) once for <u>each</u> feature to which the segment belongs. Each segment is output as a separate graphic element in order to portray the original data structure. Features assigned areal symbolization are verified to ensure that the Intergraph system is directed to perform areal symbolization only against closed polygons. The input SLF is converted to the Intergraph polygonal data structure; feature-segment relationship information is lost.

The Intergraph Interactive Graphics Design System (IGDS) and Data Management and Retrieval System (DMRS) do not provide a direct mechanism for establishing and maintaining feature-segment relationships. In addition, DMRS does not support variable length records. Despite these restrictions, the Conversion Software preserves input DSI, registration point and feature attribute information via its output as SIF commands for generation of a DMRS data base.

Complex symbolization (e.g., ticked lines) is accomplished via Intergraph patterning. The Conversion Software assigns a color code for every output graphic element. All pattern component elements generated by the Intergraph PDV patterning task are assigned a color code of 0 (white). In addition, the linkage to the DMRS attribute data base is lost when large lineal features are patterned. A complex sequence of interactive commands are required to review the attributes of these features at an Intergraph graphics station.

The Conversion Software includes logic to ensure that patterning can successfully be applied. All areal features are output as shapes; features with more than one segment or more than 100 points are output as complex shapes. The Conversion Software ensures that all shapes are closed. Commands generated to direct patterning conform to all features of the SIF, both documented and undocumented. The SIF command for patterning must be output for <u>each</u> graphic element to be patterned, even if consecutive elements are assigned the same pattern. To ensure that elements are not multiply patterned, a SIF command for the null pattern must be output after each element. The Intergraph SIF processors do not recognize the SIF area pattern command; all patterns must be defined as lineal.

An Intergraph design file cannot be dynamically extended; sufficient space should be allocated when the file is created. The SIF Environment File specifies the design file allocation for the Intergraph SIF processors. Truncation of the design file must be inhibited if patterning is going to be applied.

The Intergraph patterning function creates graphic elements depicting the pattern in the IGDS design file. Each element in an IGDS design file includes a 19 word (38 byte) header plus the data points (8 bytes per x,y pair). As a result, patterned design files can consume considerable disk

resources (e.g., 10000 512-byte blocks or more), depending upon pattern complexity, pattern density and area pattern coverage (for example, a large area with a pattern of dots can be devastating). The processing time consumed by the Intergraph patterning task can be significant and appears to be related to these factors.

During development, it was apparent that the Intergraph PDP-11/70 was unable to handle heavy processing loads associated with multiple users: simultaneous use of the SIF processors, interactive graphics, the PDV patterning task and/or interactive DMRS resulted in system crashes. Coordination with other system users is required when processing the SIF data on the PDP 11-70 based Intergraph system. In the event of a system crash, the Cell Library and Pattern File must be reviewed to ensure that they have not been corrupted.

3.4 Recommendations

The Conversion Software was developed to provide a prototype capability within the USAETL research environment. The software represents a unique set of capabilities for processing cartographic data in DMA Standard Linear Format and for interfacing to the Intergraph environment via Standard Interface Format. These capabilities, in all or in part, may serve as a framework for the specification and development of software environments established to meet specific functional requirements of users of SLF data and/or Intergraph systems. These characteristics, as well as suggested enhancements to the Conversion Software, are addressed by the following.

3.4.1 Alternate Applications

DATA TRANSFER: The Conversion Software provides a generalized capacity for conversion of data in DMA Standard Linear Format into Intergraph Standard Interchange Format. The DMA is assigned responsibility for meeting mission requirements of a variety of users, primarily within the DoD. These users, including internal DMA production systems, are often faced with requirements for conversion of input data to an internal vendor format. Several vendor graphic systems provide capabilities to input data in SIF; current and future DMA digital products are supplied in SLF. The Conversion Software can serve as a tool for porting data between systems and for distributing digital information to the user community. Future applications may consider use of this software, either directly or indirectly, as a mechanism for conversion to a device independant graphics format.

DATA PROOFING: The Conversion Software will accept SLF for any product type, creating unsymbolized data for input to the Intergraph environment. The Intergraph system provides powerful capabilities for interactive review and manipulation of graphic information. Windowing and zoom capabilities enable detailed review of graphic data at a very large scale. The Conversion Software can be used as a production tool to convert an interim or final product, in SLF, to an Intergraph compatible format. Interactive review at an Intergraph graphics station can be used in lieu of, or as an adjunct to, a proof plot. The former may be appropriate when plotter resources become a production bottleneck. The latter use may include assessment of overall data characteristics and quality. Large scale displays can be used to examine nodes (i.e., segment endpoints) to identify gaps and overshoots and to evaluate line characteristics (i.e., point density, effects of line smoothing, etc.). Intergraph reference file capabilities can be used to graphically compare differences in multiple SLF data sets at a variety of scales. The Conversion Software also provides a mechanism for evaluation of data produced by systems without graphics capabilities and/or under development.

DATA AUDIT: The Conversion Software is tolerant of, and reports inconsistencies in, the input SLF data. It can be used as a batch audit program to detect incorrect feature-segment relationships, open areal features, point features with multiple points or segments and other anomalies in a SLF data set. Areal features depicting multiple closed areas (i.e., islands) are supported. The Conversion Software can thus be used as a tool to validate data input to, and generated by, software being developed or modified to provide enhancements.

SYMBOLIZATION: The Conversion Software assigns symbology, on a per feature basis, to be generated by the Intergraph system. Minor adjustments are frequently required to computer generated final symbolization prior to creating film negatives and the final products. These minor edits address the aesthetic nature of cartographic products and often include slight shifting of features to resolve symbol coalescence and more accurate text positioning. The Intergraph system is well suited to these types of edits. The Conversion Software is designed to readily support alternative or additional symbologies. Support for additional SLF product types is addressed below as an enhancement.

ALGORITHMS: The Conversion Software incorporates algorithms and design approaches for the processing of SLF data which may be applicable to future software development activities. The software provides a pattern for use of the VMS Message Utility. The Mag Tape Services provides general capabilities for dealing with foreign tapes in the VAX environment. The software for converting from SIF to SLF includes logic for sequencing segments within features, and supports multiple closed areas within areal features.

3.4.2 Enhancements

USER INTERFACE: The Conversion Software was developed to provide a prototype capability and, as such, includes a fairly primitive user interface. Several enhancements are recommended to provide requisite support for the production environment, including:

- Retention of defined jobs in a library with the capability to recall and modify an existing job, submit a job for execution, remove a job from the library and report the library contents.

- Refinement of the algorithm for estimating required design file size.

- Automatic deletion of temporary disk files created by the Conversion functions, selected as an option when a job is defined.

- More fully automated procedures for input/conversion of the SIF data on the Intergraph system.

VAX UPGRADE: The Conversion Software generates SIF data for input to Version 8.5 of the Intergraph software executing under RSX-11M on the PDP-11/70. This SIF format is not compatible with Version 8.6 and higher of the Intergraph software executing under VMS on the VAX. It is recommended that a companion version of the Conversion Software be developed to support the Intergraph VAX SIF. This can be accomplished via minor extensions to the existing software, including an expanded Symbol Library format. The Intergraph approach to patterning SIF data is significantly revised in the VAX environment. The pattern definition file, which is extremely tedious to create and maintain, as well as error-prone, has been eliminated. This will permit greater flexibility in symbol generation (since the entire file need not be redefined to add or modify a single pattern) and could potentially resolve some of the

limitations discussed in the preceding sections. The Intergraph VAX environment would be expected to support concurrent users, an area where difficulties were encountered on the Intergraph PDP system. The enhancement to support the VAX SIF may be particularly attractive to DMA, since DMA has several VAX-based Intergraph systems in the production environment. In addition, the Conversion Software could be executed on the target Intergraph VAX-based system. This will eliminate requirements to port SIF on tape and would allow conversion from SIF to IGDS/DMRS to be intergrated within the Conversion Software user interface as an optional function.

ADDITIONAL SLF SUPPORT: The Conversion Software symbolizes and outputs attributes for recognized SLF product types, including the indexing system defined in the Defense Mapping Agency Standard Cartographic Feature Digital Identification Catalog. The software was designed so that support for additional product types can be readily incorporated. The addition of a product type requires definition of a basic Symbol Correlation File and Cell Library to support that product. Certain products may require that the Symbol Correlation Library format, and associated software, be expanded. Each product type also requires software to convert feature information for output as attributes. The Defense Mapping Agency Feature File (DMAFF), currently defined by a set of draft specifications, establishes conventions for storing feature attributes for a variety of product types. It is expected that, as this format evolves, current and future DMA production systems will be enhanced/designed to support DMAFF. Any enhancements to the Conversion Software should consider support for DMAFF, both for attribute output and for product specific attribute interpretation and symbology assignment. A current limitation results from the "hard coded" (although highly parameterized) conversion from SLF attributes into a DMRS schema. It is recommended that, concurrent with any enhancement to support additional product types, a mechanism be established to describe the DMRS schema as an input parameter for the Conversion Software.

ELEVATION DATA OUTPUT: The Conversion Software accepts input data containing elevation information, but does not generate 3D SIF output data. This capability would significantly increase design file size (2 bytes per point) and could surface unforeseen problems with the Intergraph SIF processors. The software is designed to support this enhancement, which may be useful as a graphic review tool, depending upon intended use of the software. The applicability of elevation information to symbolized data is questionable.

CONVERSION TO SLF: The software for conversion from SIF to SLF was developed to provide a mechanism for digitizing additional test input data on the Intergraph system. Capabilities to output more than one SLF data set per tape are not provided. Due to the inherent difficulties associated with creation of the SLF feature-segment structure in the Intergraph environment, further revisions to this software are not recommended.

APPENDIX A

USERS MANUAL

TABLE OF CONTENTS

APPENDIX C MESSAGE DOCUMENTATION

LIST OF FIGURES

A. 1 GENERAL

A. 1. 1 Purpose of the Users Manual

The objective of the Users Manual for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF) and Software Conversion of Intergraph Corporation Standard Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF), Contract Number DAAK70-83-C-0174 is to provide users with the information necessary to effectively use the system.

A. 1. 2 Project References

The following references were used in the preparation of this document:

1. Statement of Work for Software Conversion of Intergraph Corporation Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF), ETL-TD-MA, 21 March 1984.

2. Statement of Work for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF), ETL-TD-MA, 30 June 1983.

3. Defense Mapping Agency Product Specifications for 1:50,000 Scale Topographic Maps of Foreign Areas, PS/3AA/101, July 1980.

4. Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog, July 1977.

5. Defense Mapping Agency Standard Linear Format (SLF) for Digital Cartographic Feature Data, 16 May 1983.

6. Intergraph Standard Interchange Format (SIF) Command Language Implementation, DIXD1300, 30 May 1982.

7. Intergraph Standard Interchange Format (SIF) Users Guide, DIXD0500, 15 November 1980.

8. Intergraph Interactive Graphics Design System (IGDS) Operating Manual (IGDS8), 79-067C, 13 August 1981.

9. Intergraph DMRS Data Definition Language Compiler User Manual, 79-050C, 20 June 1981.

10. Intergraph DMRS 8.3 Command Language Users Guide, 79-065C, 7 July 1981.

11. Military Standard Weapon System Software Development, MIL-STD-1679 (NAVY), 1 December 1978.

12. Department of Defense Automated Data System Documentation Standards, Standard 7935.1-S, 13 September 1977.

13. SLF to SIF Software Conversion Design Working Paper, Measurement Concept Corporation, 7 December 1983.

14. SLF to SIF Software Conversion System Specification Working Paper, Measurement Concept Corporation, 11 April 1984.

15. SIF to SLF Software Conversion Design Working Paper, Measurement Concept Corporation, 30 November 1984.

16. SIF to SLF Software Conversion Specification Working Paper, Measurement Concept Corporation, 15 February 1985.

17. SLF-SIF Conversion Software Messages, Measurement Concept Corporation, July 1985.

A.1.3 Terms and Abbreviations

The following terms and abbreviations are used in this document:

DEC	Digital Equipment Corporation
DMA	Defense Mapping Agency
DMRS	Data Management and Retrieval System
IGDS	Interactive Graphics Design System
Mc2	Measurement Concept Corporation
RMS	Record Management Services
SIF	Standard Interchange Format
SLF	Standard Linear Format
USAETL	US Army Engineer Topographic Laboratories

A.1.4 Security and Privacy

The SIF to SLF conversion software does not include any classified components. Information is not recorded for individual users.

A.2 SYSTEM SUMMARY

The Software Conversion between Defense Mapping Agency Standard Linear and Intergraph Corporation Standard Interchange Format provides a prototype software capability. Intergraph SIF including commands for point, lineal and areal feature segments and commands containing feature attribute information may be input to produce SLF output. DMA SLF data may be input to produce SIF output including commands for point, lineal and areal feature symbology as well as commands to retain feature attribute information.

A.2.1 System Operation

The SLF-to-SIF effort was performed by Measurement Concept Corporation (Mc2) under contract to the US Army Engineer Topographic Laboratories (USAETL) as one component of the Carto Data Validation Project within the Automated Cartography Branch at USAETL. This project is establishing a research environment for a production flow of auto-carto data between multiple systems. The conversion between SLF and SIF provides a prototype capability for the input of data in DMA Standard Linear Format to the Intergraph system, which can be used for graphic manipulation of the data and eventual output in a device independent graphics format. Conversion of Intergraph SIF to DMA SLF was performed to enable further software testing with appropriate feature classifications and data conditions.

The DMA has established the Standard Linear Format (SLF) as a vehicle for the exchange of digital cartographic information between the various production systems within the DMA Centers and to serve as a DMA product format. The SLF data structure supports a single spatial description which is associated with one or more cartographic features.

The Intergraph Corporation Standard Interchange Format (SIF) provides a mechanism for the exchange of digital graphic information between graphic systems of various vendors.

The Intergraph system includes the Interactive Graphics Design System (IGDS) and the Data Management and Retrieval System (DMRS). The IGDS provides the capability to generate and maintain graphic information with attribute information maintained in an associated DMRS data base. In addition to rudimentary symbolization such as color and line weight, IGDS cells provide capabilities for complex point symbology and patterning provides capabilities for lineal and areal symbology.

A.2.2 System Configuration

The SLF-SIF Conversion software executes on the DEC VAX family of processsors under Version 3 of the VMS operating system. The optional VAX-11 FORTRAN product is required.

Input and output Intergraph SIF tapes are compatible with an Intergraph system based on the DEC PDP-11/70 processor with Version 8.5.1 of the Intergraph IGDS and DMRS software. The optional Intergraph Standard Interchange Format (SIF) product is required.

A.2.3 System Organization

Distinct activities performed to accomplish conversion between DMA SLF and Intergraph SIF include:

- o creation of SIF input using the Intergraph system
- o conversion from SIF to SLF
- o definition of symbolization to be performed
- o conversion from SLF to SIF
- o conversion from SIF to IGDS/DMRS formats

Standard Intergraph mechanisms are used to build SIF for input to the conversion process. Attribute information is entered using DMRS to conform to the schema expected by the conversion software.

Symbology can be defined any time prior to performing a conversion. Once defined, a symbology can be used repeatedly or can be modified to define new symbolization.

The conversion process is initiated at an alphanumeric terminal. Software provides an interactive user interface for initiation of the job, definition of the required parameters and submittal of the conversion to a VMS batch queue. DMA SLF data may be converted to SIF. Optionally, SIF data may be used as an input to generate SLF data.

Standard Intergraph mechanisms are used to convert the resulting SIF data to Intergraph IGDS/DMRS form.

A.2.4 Performance

Several factors influence throughput, including the volume of the data and the functions performed. Interactive response times are within the 3-10 second range, depending upon the load on the VAX. Resource requirements of the conversion process depend upon the volume of data. Tape input and output is limited by the speed of the tape drive.

Resource requirements for conversion from SIF to IGDS/DMRS format are highly dependent upon the data, and increase as the number of points and number of attributes increase. Of particular note are the requirements for patterning, which requires significant disk and PDP CPU resources.

A.2.5 Data Base

The Conversion software does not maintain a data base, per se. Symbolization to be applied is defined by Symbol Correlation Files and associated Symbol Correlation Libraries. The Conversion process accepts SIF or SLF input and generates a series of temporary files which represent the data set as part of its processing.

Several external files, resident on the Intergraph system, support creation of SIF data input to the Conversion and manipulation of SIF output from the Conversion software. These include:

- Seed Design File - enables creation of IGDS Design File from SIF
- Cell Library - defines cell elements which implement point and pattern symbology specified via Symbol Correlation File
- Pattern Definition File - describes patterns which implement symbology specified via Symbol Correlation File
- SIF Environment File - provides parameter information for input to the SIF processors
- DMRS Data Definition Language File - decribes schema of DMRS data base for SIF input to Conversion and SIF output from Conversion. (Note: the two schemas differ.)

A.2.6 General Description

A.2.6.1 Inputs

Inputs to the Conversion software may be in Intergraph Standard Interchange Format (SIF) or in Defense Mapping Agency Standard Linear Format (SLF). Process parameters are input interactively and are described below.

The DMA Standard Linear Format (SLF) is designed as the vehicle for the exchange on magnetic tape of digital cartographic feature data, consisting of planimetric and/or hypsometric information created in the production of various DMA hydrographic, topographic and aeronautical products. The format of a SLF feature header varies by product type. The SLF to SIF conversion software supports the 1:50000 Scale Topographic Maps of Foreign Areas product, as shown by Figure A-2-1.

Intergraph Standard Interchange Format (SIF) provides a common mechanism for transmittal of graphics and associated data between systems and is described in Reference 6.

A.2.6.2 Processing

The Conversion Software supports interactive definition of a multi-step job to be executed in a batch mode. The Process SLF step is used to convert between external formats and the Conversion Software internal format. The following lists the functions for several typical jobs:

Convert SLF to SIF:
- Tape I/O
- Process SLF
- Convert SLF to SIF
- Tape I/O

Convert SIF to SLF:
- Tape I/O
- Convert SIF to SLF
- Process SLF
- Tape I/O

Convert SIF to SLF to SIF:
- Tape I/O
- Convert SIF to SLF
- Convert SLF to SIF
- Tape I/O

Define/Submit Job provides the capability to interactively initiate any combination of functions comprising the conversion process. Once defined, a job is submitted to the batch queue for execution. Each job produces a run summary report consisting of errors and/or exception conditions detected, and other pertinent information. The following describes functions which may be included in a Conversion job.

Build Symbol Library converts an ASCII format Symbology Correlation File into a Symbol Correlation Library format for use by the Conversion software. The Symbology Correlation File is created using a standard VMS editor.

Tape I/O provides a mechanism to input SLF or SIF data from tape to disk and to output SIF or SLF data from disk to tape.

Process SLF converts between a disk resident SLF data set and the Conversion software internal data set format. The conversion may be from SLF to internal or from internal to SLF.

Convert SLF to SIF generates a disk resident SIF file from a data set in the Conversion software internal format. Feature symbolization commands are generated in accordance with the specified Symbol Correlation Library.

Convert SIF to SLF generates a Conversion software internal format data set from a disk resident SIF file. DMRS attribute information is used to derive SLF feature, segment and DSI information. Graphic symbolization is ignored.

A.2.6.3 Outputs

Data is output in Intergraph SIF or in DMA SLF. In addition, processing summary reports are generated.

The feature header consists of one 40 byte record as illustrated below. All data values are stored as ASCII characters. The 8-bit ASCII character set is used. All fields are right justified, with leading zeros or spaces to fill the field. Unused attribute fields contain ASCII spaces (octal 040).

FIELD NAME	LENGTH	DESCRIPTION OF CONTENTS
Feature Code	4	DMA Standard Feature Code Number as per the DMA Standard Cartographic Feature Digital - Identification Catalog
Attribute1	3	Feature attribute code as per the DMA Standard Cartographic Feature Digital - Identification Catalog
Attribute2	3	Feature attribute code as per the DMA Standard Cartographic Feature Digital - Identification Catalog
Pad	30	Unused data field contains ASCII spaces (octal 040).

Figure A-2-1 Feature Header

A.3 TECHNICAL OPERATIONS

A.3.1 Define Symbology Correlation File

A Symbology Correlation File is interactively constructed or modified using a standard VMS editor.

A.3.1.1 Initiation Procedures

The utilization of VMS text editors is described in the VMS documentation set. In addition, the EDT editor includes the EDTCAI on-line instruction course.

A.3.1.2 Input Requirements

The Symbology Correlation File is an ASCII text file describing feature symbolization information associated with a feature identification code. The format is such that it is easily created and manipulated using the EDT editor. Each input line must be in the format:

FIDC, AT1, AT2, T, LV, COL, WT, S, TEXT, CELLNM, XS, YS, ZS, PAT;

A comma (,) is used to delimit fields and must be present for each field. A semicolon (;) will end the line. Blanks may be used as desired. A line beginning with an exclamation point (!) will be treated as a comment.

Fields include:

- FIDC: Feature ID Code - 4 character identification code for the feature classification as per the Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog.
- AT1: Feature Attribute1 Code - 3 character first attribute code for the feature classification as per the Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog.
- AT2: Feature Attribute2 Code - 3 character second attribute code for the feature classification as per the Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog.

- T: Type - 1 character type of symbology to be generated, where P = point, L = lineal, A = areal.

- LV: Level - 1 or 2 character number ranging from 1 to 63 specifying the level on which feature data will be placed in the IGDS design file.

- COL: Color - 1 to 3 character number ranging from 0 to 127 which specifies the color code to be associated with the feature.

- WT: Line Weight - 1 or 2 character number ranging from 0 to 31 which specifies the number of extra lines to be displayed coincident to the feature.

- S: Line Style - 1 character number ranging from 0 to 4 which specifies the style (i.e., lineal symbology) with which the feature is to be displayed.

- TEXT: Text - from 1 to 12 characters of textual information to be output in association with the feature symbology.

- CELLNM: Cell Name - 1 to 6 character ASCII cell name identifying the point symbology to be applied to the feature. The cell is defined via the Cell Library.

- XS: X Scale - 1 to 10 character number identifying the cell scaling factor (in Units of Resolution, UOR) in the X direction. If not specified, this value is defaulted to 1.

- YS: Y Scale - 1 to 10 character number identifying the cell scaling factor (in UOR) in the Y direction. If not specified, this value is defaulted to 1.

- ZS: Z Scale - 1 to 10 character number identifying the cell scaling factor (in UOR) in the Z direction. This field is ignored for 2-dimensional design files. If not specified, this value is defaulted to 1.

- PAT: Pattern Number - 1 to 3 character number ranging from 1 to 100 identifying the pattern to be applied to the feature. The pattern is defined via the Pattern Definition File.

For example:

```
!
! This is a sample Symbol Correlation File
!
!FIDC, AT1, AT2, T, LV, COL, WT, S, TEXT, CELLNM, XS, YS, ZS, PAT;
 DEFP,    ,    , P, 56,   1,  0, 0,     , DEFPT,   ,   ,   ,   ;
DEFL, , , L, 57, 2, 0, 0, , , , , , ;
DEFA,,,A,58,3,0,0,,,,,,;
!
!ROADS; HARD SURFACE - ALL WEATHER; Divided Highway, with Median Strip
 1221, 038, 019, L,  1,   3, 20, 0,   1D,       ,   ,   ,   ,   ;
```

A.3.1.3 Output Requirements

The Symbology Correlation File is subsequently input to the Build Symbol Correlation Library program.

A.3.2 Define/Submit Conversion Job

A.3.2.1 Initiation Procedures

Define/Submit Conversion Job may be invoked from any SLF2SIF account by entering:

CSLF

in response to the VMS $ prompt. The command will prompt for inputs as described below. Default values are indicated in square brackets ([]) and may be used by entering a carriage return. All file names are specified using the VMS syntax of device:[directory]filename. Device and directory are optional and may be defaulted. File names consist of 1 to 9 alphanumeric characters (A-Z, a-z and 0-9). File types are defined as appropriate by the Conversion software and should not be entered.

A.3.2.2 Input Requirements

Once invoked, the Define/Submit Conversion Job prompts:

Job Name:

Enter the name to be used for the conversion job as 1 to 9 alphanumeric characters. The command then prompts:

Print Log [N]?

Respond Y to output the process log from the conversion job to the line printer when the conversion job completes. Logs are automatically deleted once they have been printed. Respond N to retain the process log from the conversion job in a file. If you request that the log be output to a file, you are prompted for the Log File Name. File name specification conventions described above are applicable. You may specify a file type for the log file; the default file type is LOG.

A menu of options, shown in Figure A-2-2, is displayed and the user is prompted for a selection. Each option leads the user through a series of prompts to supply input parameters, as described below. Once the option is completed, the menu is redisplayed. Any combination of as many options as desired may be selected as part of the job. Each specified function will be executed, in the sequence defined, when the job is submitted to the batch queue.

```
DEFINE CONVERSION JOB JobName



1  TAPE I/O
2  PROCESS SLF
3  CONVERT SLF TO SIF
4  CONVERT SIF TO SLF
5  BUILD SYMBOL CORRELATION LIBRARY
6  SUBMIT JOB
7  QUIT

Next function:
```

Figure A-2-2 Define/Submit Conversion Job Menu

TAPE I/O: The first prompt asks you:

Tape Input?

Respond "Y" to read a file from tape to disk or "N" to write a file from disk to tape. You will then be asked:

SLF tape?

Respond "Y" if the file to be transferred is a SLF file or "N" if the file is a SIF file. You are then asked to supply both the tape and disk file names. The default name may be chosen by entering a carriage return. For a SLF tape, the specified file name is searched for or written to the mag tape header label. For input of a SLF file from tape, you are asked for the occurrence number of the file on tape. The default occurrence number is 1. The system tape operator will be prompted to mount the tape when the function executes.

PROCESS SLF: The first prompt ask:

Convert Internal to SLF?

Respond "Y" to create a SLF file from an internal format data set. Respond "N" to create an internal format data set from a SLF file. You are then asked for the name of the data set. The command then prompts:

Coordinate scaling factor?

The conversion process will multiply all coordinates by the specified scale factor.

CONVERT SLF TO SIF: The first prompt asks for the name of the SLF data set to be converted. The command requests the name of the Symbol Correlation Library and then prompts:

Default symbology only (Y/N) [N] ?

Respond "Y" to apply default symbology to all features. The command finally prompts:

Coordinate scaling factor?

The conversion process will multiply all coordinates by the specified scale factor. The scale factor should be greater than or equal to 1.0.

CONVERT SIF TO SLF: The first prompt asks for the name of the SIF data set to be converted. The command then prompts:

Coordinate scaling factor?

The conversion process will multiply all coordinates by the specified scale factor. The scale factor should be less than or equal to 1.0.

BUILD SYMBOL CORRELATION LIBRARY: The first prompt asks:

Symbol Correlation Text input file:

Supply the specification of the input Symbol Correlation file (with no file type). This input file is created as described in section 3.1, above. If the specified file is not found, you will be notified and will be asked to enter the filename again. The command then prompts for the Symbol Correlation Library name. Enter the desired specification of the output library.

SUBMIT JOB: Submit Job has no prompt messages. It will submit the defined job to the default VMS batch queue and exit to VMS.

QUIT: Quit requires that you confirm your intention to quit. To quit from the job definition, enter Y, y or any odd number (or letter). To continue, respond with N, n or any even number (or letter).

A.3.2.3 Output Requirements

The Define/Submit Conversion Job outputs a process report to either the printer or a file. When a job which was submitted to the batch queue completes, a notification message is signaled at the terminal.

The following identifies the output from each of the options available via the Define/Submit Conversion Job procedure:

- TAPE I/O: Outputs a SLF or SIF data set to disk or tape.
- PROCESS SLF: If converting from Internal Format to SLF, a SLF data set is output. If converting from SLF to Internal Format, an internal data set consisting of Data Set Identifier (DSI), Feature (FEA), Header (HDR), and Segment (SEQ) files is output.

- CONVERT SLF TO SIF: A SIF data set is output.
- CONVERT SIF TO SLF: An internal data set consisting of Data Set Identifier (DSI), Feature (FEA), Header (HDR), and Segment (SEG) files is output. A temporary Indexed Feature (IFE) file is deleted by the program.
- BUILD SYMBOL CORRELATION LIBRARY: A Symbol Correlation Library (SCL) file is output.

A.3.3 Utilization of System Outputs

A.3.4 Recovery and Error Correction Procedures

A complete list of informational and error messages generated by the Conversion software is documented under separate cover in reference 17, SLF-SIF Conversion Software Messages.

A.3.4.1 VAX Environment

Each Conversion job generates a process log which indicates the functions performed, resources used and completion status. The Conversion process generates SIF data on magnetic tape, which is transported to the Intergraph system. Intergraph utilities invoked to convert the SIF data to Intergraph internal format are described below.

A.3.4.2 Intergraph Environment

All SLF-SIF Conversion files are contained under the user account /RR with a UIC of [5,333] on the USAETL Intergraph system. The default login device and directory is QS1:[5,334]. To point to the Conversion account, enter:

```
SET /UIC=[5,333]
```

Additional disk space has been authorized on QS2:. The SIF processors must be installed to be executed, but need be done only once during each system up-period. To install the SIF processors, enter:

```
@[5,334]SIFINS
```

SIF data manipulation is controlled by a series of files; each data data set must have its own set of control files. A set of generic files is provided to serve as a pattern for any data set. Each can be copied and edited for a specific data set, replacing all occurrences of the word 'generic' with the data set name. Each control file is described below.

- Environment File, GENERIC.ENV - contains parameters used as input to the SIF processors.
- Data Definition Language File, GENERIN.DDL and GENEROUT.DDL - define the associated data base schemas for DMRS.
- SIF Input Command File, GENERIN.CMD - logically allocates and mounts the SIF tape, invokes the BTI processor, dismounts the tape, and invokes the TRI processor.
- SIF Output Command File, GENEROUT.CMD - invokes the TRD processor, logically allocates and mounts the tape, inkokes the BTI processor, and dismounts the tape.

A.3.4.2.1 SIF Input

To input a SIF file from tape, first create the customized DDL, ENV and CMD files specific for the data set using the generic files described above. These files may require size allocation adjustments as well as file name specifications. Enter:

```
DDL @dsname.DDL
```

to create an empty data base to be populated from the SIF input. Then enter:

```
@dsname.CMD
```

This will allocate and mount the tape, run BTI, dismount the tape, and run TRI. BTI and TRI processor errors will appear in BTI.MSG and TRI.MSG files, respectively. If the Conversion job log file indicates that any graphic elements have been patterned, issue the following command:

```
PDV dsname.LST=dsname.DGN
```

to perform the patterning. Patterning requires considerable processing resources and the activity should be coordinated with other system users. Patterning errors will appear in the dsname.LST file. The resultant design file and be brought up in

graphics by entering:

@G dsname.DGN

If the TRI process reports a spawn error, Attribute Services may have been left active by a prior abort. The ACT command will list active tasks; AS.Tnn is the Attribute Services task for your terminal (nn) and may be aborted via ABO AS.Tnn.

Standard IGDS/DMRS commands may be used to review the output graphics and attributes. To review attributes, the data base must be attached via the DB=dataset.DBS! keyin. Review criteria are specified by a keyin in the form RA=FEA.:*!. The PDV patterning task drops attribute linkages; to review attributes for patterned elements, place a fence around the desired elements and select the Review Attributes Report command. Respond to each prompt with a space, carriage return; complete the report with the Page Swap function key.

A.3.4.2.2 SIF Output

For successful conversion from SIF to SLF, the data must be digitized so that SLF conventions are observed and implemented via the IGDS and DMRS files created. The GENEROUT.DDL file may be used as a pattern for creating a DDL file which will create a data base with the correct schema. The GENEROUT.CMD file serves as a pattern for creating a command file to output as SIF. To invoke this process enter:

@dsname.CMD

This will run TRO, allocate and mount the tape, run BTO, and dismount the tape. BTO and TRO processor errors will appear in BTO.MSG and TRO.MSG files, respectively.

APPENDIX B

PROGRAM MAINTENANCE MANUAL

B.1 GENERAL

B.1.1 Purpose of the Program Maintenance Manual

The objective of the Program Maintenance Manual for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF) and Software Conversion of Intergraph Corporation Standard Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF), Contract Number DAAK70-83-C-0174 is to provide programmers with the information necessary to effectively maintain and enhance the software.

B.1.2 Project References

The following references were used in the preparation of this document:

1. Statement of Work for Software Conversion of Defense Mapping Agency Standard Linear Format (SLF) to Intergraph Corporation Standard Interchange Format (SIF), ETL-TD-MA, 30 June 1983.

2. Statement of Work for Software Conversion of Intergraph Corporation Interchange Format (SIF) to Defense Mapping Agency Standard Linear Format (SLF), ETL-TD-MA, 21 March 1984.

3. Defense Mapping Agency Product Specifications for 1:50,000 Scale Topographic Maps of Foreign Areas, PS/3AA/101, July 1980.

4. Defense Mapping Agency Standard Cartographic Feature Digital-Identification Catalog, July 1977.

5. Defense Mapping Agency Standard Linear Format (SLF) for Digital Cartographic Feature Data, 16 May 1983.

6. Intergraph Standard Interchange Format (SIF) Command Language Implementation, DIXD1300, 30 May 1982.

7. Intergraph Standard Interchange Format (SIF) Users Guide, DIXD0500, 15 November 1980.

8. Intergraph Interactive Graphics Design System (IGDS) Operating Manual (IGDS8), 79-067C, 13 August 1981.

9. Intergraph DMRS Data Definition Language Compiler User Manual, 79-050C, 20 June 1981.

10. Intergraph DMRS 8.3 Command Language Users Guide, 79-065C, 7 July 1981.

11. Military Standard Weapon System Software Development, MIL-STD-1679 (NAVY), 1 December 1978.

12. Department of Defense Automated Data System Documentation Standards, Standard 7935.1-S, 13 September 1977.

13. SLF to SIF Software Conversion Design Working Paper, Measurement Concept Corporation, 7 December 1983.

14. SLF to SIF Software Conversion System Specification Working Paper, Measurement Concept Corporation, 11 April 1984.

15. SIF to SLF Software Conversion Design Working Paper, Measurement Concept Corporation, 30 November 1984.

16. SIF to SLF Software Conversion Specification Working Paper, Measurement Concept Corporation, 15 February 1985.

17. SLF to SIF Conversion Software Users Manual, Measurement Concept Corporation, July 1985.

18. SLF to SIF Conversion Software Messages, Measurement Concept Corporation, July 1985.

B.1.3 Terms and Abbreviations

The following terms and abbreviations are used in this document:

DEC	Digital Equipment Corporation
DMA	Defense Mapping Agency
DMRS	Data Management and Retrieval System
IGDS	Interactive Graphics Design System
Mc2	Measurement Concept Corporation
RMS	Record Management Services
SIF	Standard Interchange Format
SLF	Standard Linear Format
USAETL	US Army Engineer Topographic Laboratories

B.2 SYSTEM DESCRIPTION

B.2.1 System Application

The Software Conversion between Defense Mapping Agency Standard Linear Format and Intergraph Corporation Standard Interchange Format provides a prototype software capability. Intergraph SIF including commands for point, lineal and areal feature segments and commands containing feature attribute information may be input to produce SLF output. DMA SLF data may be input to produce SIF output including commands for point, lineal and areal feature symbology as well as commands to retain feature attribute information.

The SLF to SIF effort was performed by Measurement Concept Corporation (Mc2) under contract to the US Army Engineer Topographic Laboratories (USAETL) as one component of the Carto Data Validation Project within the Automated Cartography Branch at USAETL. This project is establishing a research environment for a production flow of auto-carto data between multiple systems. The conversion between SLF and SIF provides a prototype capability for the input of data in DMA Standard Linear Format to the Intergraph system, which can be used for graphic manipulation of the data and eventual output in a device independent graphics format. Conversion of Intergraph SIF to DMA SLF was performed to enable further software testing with appropriate feature classifications and data conditions.

B.2.2 Security and Privacy

The SIF to SLF conversion software does not include any classified components. Information is not recorded for individual users.

B.2.3 General Description

Software supports conversion from DMA SLF to Intergraph SIF with the optional application of symbolization information. Functionally, the SLF data is first converted to an internal format from which the SIF data can be generated. The output SIF has associated attribute information reflecting the input feature attributes. The software for conversion from Intergraph SIF to DMA SLF was developed to ensure the availability of data to test the symbolization supported and to exercise the conversion software. The input SIF has associated attributes which provide sufficient information to derive feature-segment relationships as well as feature attributes. Functionally, the SIF data is converted to an internal format from which the SLF data can be generated. The same internal format is employed for both

conversions. Separate functions support transfer of either format between tape and disk. Software also supports creation of a Symbol Correlation Library.

The user interface software provides the capability to define a job consisting of any combination of the conversion functions. Once defined, a job is executed on a batch queue and produces a run summary report.

B.2.4 Program Descriptions

The following describes each of the the SIF to SLF Conversion programs. Each major function includes the description of its unique modules. The modules common to several functions (e.g., Data Management Services) are grouped into a major section. The organization of this section is intended to parallel the organization of the source files (i.e., on disk). Functions are listed alphabetically.

B.2.4.1 Build Symbol Library

Build Symbol Library (BLDSYMLIB) converts an ASCII format Symbology Correlation File into the Symbol Correlation Library format for use by the SLF to SIF conversion software. BLDSYMLIB reads one record from the input file to construct a record for output to the Symbol Correlation Library. Fields of the input record are separated by commas (,) and the end of the input record is identified by a semicolon (;). A default value is assigned to each blank input field. Each field is examined to ensure that the specified value is valid. If an error is encountered, an appropriate message is generated and a default error value is used. BLDSYMLIB is invoked from the conversion command procedure.

INPUT: The Symbol Correlation Source File is read. Input parameters are obtained from the conversion command procedure.

OUTPUT: The Symbol Correlation Library is generated and a summary report is generated.

```
LOGIC:

GETSYMFILES to access Symbol Correlation File and Library
DM_GETSEQ to read first symbol file record
DO WHILE not end of file
   IF not control character or comment
      Convert to uppercase via STR$UPCASE
      IF no string terminator or wrong number of fields
         Report error and ignore
      ENDIF
      DO WHILE MoreString AND NoErrors
         IF field terminator
            CONVERT_FIELD and point to next field
         ELSEIF record terminator
            CONVERT_FIELD and set MoreString FALSE
            DM_PUT record to Symbol Library
            Echo output
         ELSE
            Point to next character and test field width
         ENDIF
      ENDDO
      DM_GETSEQ record from Symbol Correlation File
   ENDIF
ENDDO
Close Symbol Correlation File and Library
```

B.2.4.1.1 Build Symbol Library Parameters

Get Build Symbol Library Parameters (BLDSYMLIB_PARAMS) uses the LIB$GET_SYMBOL Run Time Library routine to get the input parameters as symbols from the conversion command procedure. If any error is encountered, it is reported and the status SLI_ERRGETSYM is returned.

INPUT: Symbols are obtained from the conversion command.

OUTPUT: Characters variables for the Symbol Correlation File and Symbol Correlation Library specifications are returned.

B.2.4.1.2 Check Integer Field

Check Integer Field (CHECKINT) verifies those fields to be converted to integer (byte/word) format for output to the Symbol Correlation Library. If the field width is valid, DM_CASCI is used to convert the value from ASCII to Integer. SETERROR is invoked to report any errors. The integer value is returned.

INPUT: Inputs include the field number to convert, the width of the field, the string to verify and the value format to convert to (CP_byte, CP_word, CP_long).

OUTPUT: Output is the value to be output to the Symbol Correlation Library.

B.2.4.1.3 Convert Field

Convert Field (CONVERT_FIELD) accepts any field in the input Symbol Correlation record, verifies it, and converts it into the appropriate format for storage in the Symbol Correlation Library. STRIP_BLANKS is first invoked to remove any blanks from the input string. Verification is performed according to the field type. DM_CASCI is used to convert to integers. CHECKINT is used to verify integer fields. Any errors encountered are reported and counted.

INPUT: Inputs include the identification of the field to convert, the number of characters in the field and the string to verify.

OUTPUT: The converted value is returned.

B.2.4.1.4 Get Symbol Correlation Files

Get Symbol Correlation Files (GETSYMFILES) invokes BLDSYMLIB_PARAMS to get the input parameters. The Symbol Correlation File is opened. The Symbol Correlation Library is created and opened.

INPUT: Parameters are obtained from BLDSYMLIB_PARAMS.

OUTPUT: The logical units associated with the input and output files are returned.

B.2.4.1.5 Set Error

Set Error Message (SETERROR) logs error messages.

INPUT: The input field is identified. The actual length of the field is input.

OUTPUT: The appropriate error message is output.

B.2.4.1.6 Strip Blanks

Strip Blanks (STRIP_BLANKS) removes all leading and trailing blanks from a string. Each character is examined. Blanks are output when underscore (_) characters are found in the input.

INPUT: The length of the string and the string are input.

OUTPUT: A string with the blanks removed is returned. The length of the string is revised.

B.2.4.2 Convert SIF to SLF

Convert SIF to SLF (CSIFSLF) converts an input data set in Intergraph Standard Interchange Format (SIF) to internal data set format. Graphic symbolization commands in the SIF input are ignored. Input DMRS attribute information is used to derive feature, segment and DSI information. Segments and temporary features (in IFE format) are constructed and output as the input SIF file is read sequentially. DSI information is recorded. Once the SIF file is processed, the internal Header File is generated and the IFE feature records processed to contruct the FEA file.

CSIFSLF is invoked from the conversion command procedure. Internal data set records are retained i common areas. Process control parameters include the input SIF filename, output data set name, and coordinate scaling factor. The values for these are obtained from the user when the job is defined and are obtained as symbols from the Conversion Job command.

INPUT: A data set in SIF format is input from a disk file. The SIF format is discussed in Section 3.2.2.2, below.

OUTPUT: An internal format data set, as described in Section 3.2.2.3 below, is output.

LOGIC:

SIF2SLF_PARAMS to get input parameters
Report input parameters to process log
DM_OPEN input SIF, DM_CRENUL and DM_OPEN SEG, IFE file
PROCESS_SIF to process input SIF
HDROUT to output header record
DSIOUT to output DSI
WALK_SEGMENTS to build features
DM_CLOSE files, DM_DELFIL IFE file
Report completion messages and time stamp

B.2.4.2.1 DSI Output

DSI Output (DSIOUT) outputs the DSI record which has been constructed by the Convert SIF to SLF process. DM_CI4ASC is invoked to convert counts to ASCII. The DSI file is created, opened, the record written and the file is closed.

INPUT: The name of the DSI file to be output and the length of the filename are input. The DSI record to be output is passed through the DSI Record Common.

OUTPUT: The DSI record is output.

B.2.4.2.2 Process SIF File

Process SIF File (PROCESS_SIF) processes input SIF commands. The input commands are interpreted to construct output DSI, registration point, feature and segment records. Graphic symbolization commands are ignored. The input SIF may contain any valid SIF command. Several commands describe data and its characteristics which are not applicable to SLF data or are beyond the scope of this effort. These commands are ignored, as indicated by Figure B-2-1.

INPUT: A SIF command is input through SIF Common.

OUTPUT: Based upon the command type, type appropriate subroutine is invoked to output the reformatted information.

LOGIC:

```
SIF_INPUT_CMD to extract first command from input record
DO WHILE not end of file
   IF NOT Ignoring commands
      IF Classification (20)
         IF Class > 0 Set Ignore TRUE
      ELSEIF Identifier (30)
         IF Association = Define New Entity
            IF DMRSInput THEN
               Save input Entity ID, set DMRSInput FALSE
            ENDIF
         ELSEIF Association = Graphic Data Following
            Save input Entity ID if DMRSInput
         ELSE
            Ignore
         ENDIF
      ELSEIF Line String (40)
         SLF_OUTPUT_SEG to generate segment points
      ELSEIF Symbol (43)
         SLF_OUTPUT_PNT to generate single point segment
      ELSEIF Begin Complex String (50)
         Set Complexflag and NewComponent TRUE
      ELSEIF End Complex String (51)
         Set Complexflag FALSE
      ELSEIF Begin Symbol (52)
         Set Ignore TRUE
      ELSEIF Associative Values (83)
         SLF_NEW_ENTITY to begin SEG/IFE, REG, DSI per entityID
         Set DMRSInput TRUE
      ELSE
         Ignore and accumulate statistics
      ENDIF
   ELSEIF End Symbol (53) OR (Classification (20) AND Class = 0)
      Set Ignore FALSE
   ELSE
      Ignore and accumulate statistics
   ENDIF
   SIF_INPUT_CMD to extract next command
ENDDO
```

COMMAND	No.	ACTION
Overlay	20	Ignore
Classification	22	Only elements of class 0 (primary) are processed
Association	23	Ignore
Font	24	Ignore
Line/Area Char	26	Ignore
Text Line Char	27	Ignore
Paragraph Char	28	Ignore
Identifier	30	Occurrence num = segment ID; Entity num = data type
Line String	40	Segment points
Circle	41	Ignore
Arc	42	Ignore
Symbol - Cell	43	Origin = single point segment
Symbol - Text	44	Ignore
Curve	45	Segment points
Begin Complex String	50	Start of segment
End Complex String	51	End of segment
Begin Symbol	52	Ignore following elements
End Symbol	53	Stop ignoring elements
Text Line	60	Ignore
Paragraph - Text Node	61	Ignore
Paragraph Line	62	Ignore
Close Paragraph	63	Ignore
Pad	80	Ignore
Graphic Assoc Descr	81	Ignore
Continue	82	Continue command as appropriate
Associative Values	83	Build feature, reg point or DSI
Drawing Identification	84	Ignore

Figure B-2-1 SIF Command Disposition

B.2.4.2.3 Walk Segments

Walk Segments (WALK_SEGMENTS) processes segments of a feature derived from SIF input to sequence the segments in the correct order. Feature discontinuities are allowed, although areal features are noted as errors if each group of segments forming a discontinuity are not closed. The input IFE record is processed to construct a list of segment entry numbers which are in the desired sequence for the output feature. A group of segments with connecting endpoints are treated as a subfeature.

INPUT: Features are input from the Indexed Feature File (IFE).

OUTPUT: Features are output to the Feature File (FEA).

LOGIC:

```
DM_CRENUL and DM_OPEN FEA file
DM_GETFEA_SEQ to read first IFE record
DO WHILE not end of file
   Store IFE fixed information in FEA record
   Set FEA output pointer to first C_fsegent
   BUILD_SUBFEA to build subfeatures
   MERGE_SUBFEA to merge adjacent subfeatures
   OUTPUT_SUBFEA to output subfeatures
   Move feature header information from IFE to FEA
   Compute number of bytes in feature record
   DM_PUT to output FEA record
   DM_GETSEQ next IFE record
ENDDO
DM_CLOSE FEA file
```

B.2.4.2.4 Build Subfeatures

Build Subfeatures (BUILD_SUBFEA) processes segments of a feature derived from SIF input to build a list of the segment entries for the feature.

INPUT: The IFE record is input via common.

OUTPUT: The input sequence number of each segment entry in the input IFE record is output to the segptr array. Segments are examined and those which are adjacent (by virtue of their start or end points matching) are output as a subfeature. The first and last point and direction of each subfeature is recorded. the segptr array is constructed from the middle towards both ends.

LOGIC:

```
Extract first segment entry from IFE record
Record subfea first and last points
Establish new subfea
Store extracted segment entry as current subfea middle entry
DO WHILE number segment entries done <= C_ifnumseg
   Extract next segment entry from IFE record
   Set matched FALSE, start at first subfea
   DO WHILE NOT matched AND all subfea not tested
      IF seg 1st or last pt = 1st or last pt of subfea
         Add seg entry to correct end of this subfea
         set matchedTRUE
      ELSE
         Point to next subfeature
      ENDIF
   ENDDO
   IF NOT matched
      Establish new subfea
      Store extracted segment entry as new subfea middle entry
   ENDIF
   point to next segment entry
ENDDO
```

B.2.4.2.5 Merge Subfeatures

Merge Subfeatures (MERGE_SUBFEA) merges subfeatures by building a linked list from each subfeature to the next adjacent subfeature.

INPUT: The description of subfeatures constructed by BUILD_SUBFEA is input.

OUTPUT: Two arrays are constructed which, for each subfeature, indicate the next continuous subfeature (nextsf) and whether or not this subfeature entry actually starts a new subfeature (newsub).

```
LOGIC:

Set all subfeatures as discontinuous by clearing nextsf
DO FOR number of subfeatures
   IF last point this subfeature = first point any other subfea
      Set nextsf this subfea to subfea number of match
   ENDIF
ENDDO
DO FOR number of subfeatures
   IF last point this subfea = first point first subfea
      Set nextsf first subfea to this subfea number
   ENDIF
ENDDO
Set newsub TRUE for each subfeature
set numstarts = nsubfea
DO FOR number of subfeatures
   IF subfea points to another subfea
      set newsub subfea pointed to FALSE and decrement numstarts
   ENDIF
ENDDO
IF numstarts = 0
   set numstarts = 1 and set newsub for 1st subfea TRUE
ENDIF
```

B.2.4.2.6 Output Subfeatures

Output Subfeatures (OUTPUT_SUBFEA) outputs the segment entries for the feature according to their sequence within subfeatures.

INPUT: The subfeature descriptions contructed by BUILD_SUBFEA adn MERGE_SUBFEA are input.

OUTPUT: The FEA record is returned via common.

```
LOGIC:

DO FOR number of starting subfeatures
   Find the start of this subfeature as first TRUE newsub entry
   Set newsub entry FALSE to ignore next time
   Record subfeature first point
   DO FOR first and all adjacent subfea
      Save last subfeature point
      DO FOR number segments in this subfea
         Compute offset into IFE using segptr
         Extract segment entry from IFE
         Store IFE segment entry in FEA segment entry
         Store FEA segment entry in FEA record
         Point to next subfeature
         Record subfeature closure
      ENDDO
   ENDDO
ENDDO
```

B.2.4.2.7 SLF Input Command

SLF Input Command (SLF_INPUT_CMD) searches the current SIF record for the next command and copies the command into an internal command buffer. If end of record is reached prior to extracting the entire command, SLF_INPUT_CMD reads the next record from the SIF file.

INPUT: Records are read from the SIF file.

OUTPUT: The internal command buffer is filled with the next command.

LOGIC:

```
Save Current Command Hdr as Previous Command Hdr
Initialize Total Bytes to size of Command Hdr
DO WHILE ByteCount < Total Bytes
   IF End of Current SIF record
      DM_GETSEQ to read next SIF record
      Reset to new record OR record End of File or error
   ENDIF
   Move byte into internal command buffer
   IF NLongwords byte just processed
      Increase Total Bytes by NLongwords * 4
   ENDIF
   Increment pointers
ENDDO
IF Command Type = Continuation
   Copy Type, SubType and Value bytes into current header from
      previous command header
ENDIF
```

B.2.4.2.8 SLF New Entity

SLF New Entity (SLF_NEW_ENTITY) extracts, formats and stores information from the SIF Associative Values command. The format depends upon the current entity ID (i.e., DSI, Registration Point, or Segment).

INPUT: The Associative Values command is input via the SLF Common area, and Entity ID is used to drive attribute conversion.

OUTPUT: The appropriate internal data record is updated.

LOGIC:

```
Initialize pointers and clear output fields
DO WHILE Not Done
   DO FOR number of bytes in command
      IF current byte is NOT a delimiter
         IF we're currently getting a field number
            Convert field number from ASCII to integer
         ELSE  we're getting a field value
            Save field value in a character string
         ENDIF
      ELSE  current byte is a delimiter
         Put value in DSI, REG or SEG per field number, Entity ID
      ENDIF
   ENDDO
   SLF_INPUT_CMD to input in case of Continuation command
   IF NOT Continuation command
      Set Done flag TRUE and save command for caller
   ENDIF
ENDDO
```

B.2.4.2.9 SLF Output Feature

SLF Output Feature (SLF_OUTPUT_FEA) is invoked when all the data for a segment has been input. It connects the segment to all appropriate features and stores input feature attribute information. The segment and feature are output. Previously existing features are updated, new features are created.

INPUT: The completed segment is input, along with the feature number and attributes for the feature on its left, right and coincident.

OUTPUT: The segment is output and the feature is added to or updated in the temporary Feature (IFE) file.

```
LOGIC:

Extract first Segment Feature entry
DO WHILE not all feature entries for segment done
   DM_GETIKY for feature
   IF record not found (new feature)
      Build fixed portion of feature record
      Store attributes for this feature in feature header
      Build and store Feature Segment entry
      DM_PUTFEA
   ELSE existing feature
      IF attributes for this feature = feature header
         Build and store Feature Segment entry at end of entries
         Revise feature number of segments
         DM_UPDATE_FEA
      ENDIF
   ENDIF
   Extract next Segment Feature entry
ENDDO
DM_PUT segment
```

B.2.4.2.10 SLF Output Point

SLF Output Point (SLF_OUTPUT_PNT) processes the SIF Generate Symbol command. Depending on input Entity ID, this routine either outputs a single-point segment and its associated feature, or builds a registration point.

INPUT: The Generate Symbol command is input via the SLF Common area, and Entity ID is used for internal control.

OUTPUT: The appropriate data record is updated.

```
LOGIC:

Extract coordinates from Symbol Command
IF Entity ID = SEG
   Assign next available Seg ID
   Build remainder of segment and feature records
   DM_PUT to output segment
   SLF_OUTPUT_FEA to output/update feature
ELSEIF Entity ID = REG
   Assign next available Reg point ID
ELSE
   Report error -- invalid Entity ID
ENDIF
Reset DMRS Input flag to False to prepare for next element
```

B.2.4.2.11 SLF Output Segment

SLF Output Segment (SLF_OUTPUT_SEG) is invoked upon receipt of the first Generate Line String command for a segment. It uses inputs and processes all subsequent Continuation and/or Generate Line String commands associated with the segment. Once all segment coordinates are input, the segment and its features are output.

INPUT: The Generate Line String command is input via the SLF Common area.

OUTPUT: The segment and feature records are output.

LOGIC:

```
Compute NCoordinates in command; Initialize fixed portion
Set coordinate pointer to start of pts; seg as primary
DO WHILE Not Done
   DO FOR NCoordinates in command
      IF current seg buffer is full
         IF continuation buffer
            DM_PUT to output seg continuation record
         ENDIF
         Set current seg buffer to Continuation
         Assign next available continuation ID
         Reset seg continuation buffer pointer
      ENDIF
      Copy coordinate from command buffer to seg buffer
      Update bounding rectangle; pointers
   ENDDO
   Save Current Command Hdr and SIF command buffer pointers
   SLF_INPUT_CMD to get next command
   IF SIF Continuation command
      Update Total NCoords in seg; continue with seg
   ELSEIF LineString command
      IF Complex element in progress
         Update Total NCoords in seg; continue with seg
      ELSE
         Report error -- unidentified element
      ENDIF
   ELSE ... some other command; current segment is finished
      Let caller know that next command has been input
      Set Done flag to True
   ENDIF
ENDDO
SLF_OUTPUT_FEA for corresponding features and Primary seg
```

B.2.4.2.12 SIF to SLF Parameters

SIF to SLF Parameters (SIF2SLF_PARAMS) gets input parameters from the conversion command procedure.

INPUT: Symbols are obtained from the command procedure.

OUTPUT: Input and output file names and the scale factor are returned.

B.2.4.3 Convert SLF to SIF

The Convert SLF to SIF (CVTSLFSIF) process converts the input internal format data set to a file of properly formatted SIF commands. Features are symbolized in accordance with the specified Symbol Correlation Library. In addition to input segment and feature information, Data Set Identifier (DSI) information associated with the data set is output. CVTSLFSIF is invoked from the conversion command procedure. SIF data and statistics for the process summary report are maintained in labeled common.

INPUT: An internal format SLF data set is input.

OUTPUT: A file of formatted SIF commands is output.

LOGIC:

```
SLF2SIF_PARAMS to get process control variables
HDRIN to input HDR record
SIF_INIT to perform SIF initialization
DM_OPEN Symbol Correlation Library (for read)
CONVERT_DSI to convert DSI
CONVERT_FEA to convert features
Generate Process Summary Report
IF any bad features detected
   HDROUT to output updated HDR record
ENDIF
ESTIMATE_GRAPHIC to estimate size of the design file
DM_CLOSE Symbol Correlation Library
SIF_END to Perform SIF wrapup functions
```

B.2.4.3.1 Build Sub-Features

Build SubFeatures (BUILD_SUB_FEAS) processes segments of an areal feature to construct lists of segments which can be walked endpoint-to-endpoint (i.e., subfeatures). This routine detects donut (i.e., hole) conditions and records closure characteristics and the bounding rectangle for each subfeature.

INPUT: The number of segments belonging to the feature is input. Segments are read from the segment file.

OUTPUT: Subfeature characteristics are determined and returned to the calling routine.

```
LOGIC:

DO FOR all segments belonging to feature
   INPUT_SEG to read segment
   Initialize seg bounding rectangle
   IF any segment continuation records
      GET_MORESEG to read all segment continuations
   ENDIF
   Get 1st and last segment points, considering direction
   IF subfeature in progress
      IF segment endpoints match
         Revise subfeature endpoints
      ELSE
         Begin new subfeature with current segment
      ENDIF
   ENDIF
   IF subfeature not in progress
      Start new subfeature using current segment data
   ENDIF
ENDDO
```

B.2.4.3.2 Convert SLF DSI Record

Convert SLF DSI Record (CONVERT_DSI) generates SIF output commands for the DSI input from the internal format SLF data set.

INPUT: The input DSI data set component file is read.

OUTPUT: SIF commands to generate the output DSI are generated.

LOGIC:

```
DM_OPEN to open DSI file
DM_GETSEQ to read from DSI file
DO FOR number of DSI subrecords
   BLD_ATTR_BUF to build DSI attribute values buffer
ENDDO
BLD_ATTR_BUF to build registration point attribute values
IF >0 registration points
   DM_GETCKY to get REG point symbology
   DO FOR number of registration points
      BLD_ATTR_BUF to build registration point attribute value
      SIF_ATTR to output registration point attributes
      CONVERT_REG from SLF to SIF format
      SIF_SYMBOL to output registration point symbol
   ENDDO
ENDIF
BLD_ATTR_BUF to build accuracy subset attribute values buffer
DO FOR number of accuracy outlines
   BLD_ATTR_BUF to build Accuracy Outline attribute value
   SIF_ATTR to output Accuracy Outline
   IF Geographic data
      DM_GETCKY to get ACC symbology
      SIF_BEGIN_ELE to begin accuracy outline
      SIF_OUTPUT_ACC to output outline points
   ENDIF
ENDDO
DM_GETCKY to get DSI symbology
SIF_ATTR to output DSI attributes
SIF_SYMBOL to output DSI symbol at last registration point
DM_CLOSE DSI file
```

B.2.4.3.3 Convert SLF FEA Record

Convert SLF FEA Record (CONVERT_FEA) reads features sequentially from the internal format data set. Once the feature type and symbolization are determined, the appropriate conversion routine in invoked to generate the SIF output. Segments belonging to the feature are output as graphic elements as part of the feature processing.

INPUT: Features are read.

OUTPUT: Appropriate SIF commands are output.

LOGIC:

```
DM_OPEN segment and feature files
DM_GETSEQ to read first feature record
DO WHILE not end of file
   Extract product type, attributes from feature record
   Set symbol key per feature attributes or to default value
   BLD_ATTR_BUF to build attribute buffer for feature attributes
   SIF_ATTR to output feature attributes
   DM_GETCKY to read feature symbology
   IF point feature
      CVT_POINT_FEA
   ELSEIF lineal feature
      CVT_LINEAL_FEA
   ELSEIF areal feature
      CVT_AREAL_FEA
   ELSE
      Report bad feature representation
   ENDIF
   DM_GETSEQ to read next feature
ENDDO
DM_CLOSE feature and segment files
```

B.2.4.3.4 Convert Registration Point

Convert Registration Point (CONVERT_REG) accepts the ASCII representation of a registration point and converts this to numeric values for output as a graphic symbol location.

INPUT: The data set DSI information is input, as is the registration point.

OUTPUT: The converted registration point is output.

B.2.4.3.5 Convert SLF Areal Feature

Convert SLF Areal Feature (CVT_AREAL_FEA) processes an input areal feature to generate appropriate SIF commands. BUILD_SUB_FEAS is invoked to identify the subfeatures in the feature and to verify areal feature closure. After examining each subfeature to establish the overall feature bounding rectangle, OUTPUT_SUB_FEAS is called to output each subfeature. If the feature symbology includes textual information, SIF_TEXT is invoked to place the text string at the midpoint of the feature bounding rectangle.

INPUT: The feature record is input via common.

OUTPUT: SIF commands for the feature are generated.

B.2.4.3.6 Convert SLF Lineal Feature

Convert SLF Lineal Feature (CVT_LINEAL_FEA) processes an input lineal feature to generate SIF output commands.

INPUT: The feature is input via common. Segments are read.

OUTPUT: SIF commands are output.

LOGIC:

```
DO FOR number of segments in feature
   INPUT_SEG to input first part of Segment record
   IF segment has only 1 point
      LOGBADFEA to report error
      SIF_SYMBOL to output ERRP point symbol
   ELSE
      Compute text position on first segment in feature
      OUTPUT_SEG to output line string for segment
   ENDIF
ENDDO
IF text associated with feature symbology
   SIF_TEXT output associated text
ENDIF
```

B.2.4.3.7 Convert SLF Point Feature

Convert SLF Point Feature (CVT_POINT_FEA) processes an input point feature to generate SIF output commands.

INPUT: The feature is input via common. Segments are read.

OUTPUT: SIF commands are output.

```
LOGIC:

DO FOR number of segments in feature
   INPUT_SEG to input first part of Segment record
   IF segment has only 1 point
      IF feature has > 1 segment
         LOGBADFEA to report error
         SIF_SYMBOL to output ERRP point symbol
      ELSE
         SIF_SYMBOL to output feature symbology
      ENDIF
   ELSE
      IF feature has >1 segment
         LOGBADFEA to report error
         OUTPUT_SEG to output ERRL symbology
      ELSE
         DO FOR number of points in segment
            SIF_SYMBOL to output feature symbology
         ENDDO
      ENDIF
   ENDIF
ENDDO
```

B.2.4.3.8 Estimate Graphic File Size

Estimate Graphic File Size (ESTIMATE_GRAPHIC) estimates the number of blocks to allocate for an IGDS design file. The estimate is derived from the size of Intergraph elements and input statistics which characterize the commands output. These statistics include the number of elements, number of coordinates, number of symbols, number of DMRS attribute linkages and number of elements patterned. An overhead is added.

B.2.4.3.9 Read Segment Continuation Record

Read Segment Continuation Record (GET_MORESEG) inputs a segment continuation record from the segment file. As necessary, SIF_REVERSE_PTS is invoked to reverse the sequence of the points in the segment.

INPUT: The segment ID and continuation number to be read are input.

OUTPUT: The segment is returned via common.

B.2.4.3.10 Input Segment

Input Segment (INPUT_SEG) is a service routine to input a segment. The primary record for the segment is read and the total number of points in the segment and number of segment continuation records are determined. If the segment is traversed in the reverse direction within the feature, the last continuation record for the segment is retrieved, as necessary and SIF_REVERSE_PTS is called to swap the sequence of the points in the internal buffer.

IN. UT: The feature record is input via common. The segment ID to be read is extracted from the feature record.

OUTPUT: The segment record is returned via common.

B.2.4.3.11 Log Bad Feature

Log Bad Feature (LOGBADFEA) reports errors related to invalid input features and revises internal feature counts. The feature ID is recorded in the internal data set Header record.

INPUT: The feature ID is input.

OUTPUT: The internal header record, maintained in common, is revised and a message is output.

B.2.4.3.12 Output Segment

Output Segment (OUTPUT_SEG) generates all required SIF commands to represent the input segment graphic information. As necessary, segment continuation records are read and segment data points are reversed.

INPUT: The first segment record in input via common. Other inputs include the number of coordinates in the segment, the number of continuation records in the segment, the type of line string to output and a 'last segment in the feature' indicator.

OUTPUT: SIF commands are generated.

LOGIC:

```
IF segment direction is Reverse within feature
   set pointers to process segment in reverse
ELSE
   set pointers to process segment going forward
ENDIF
SIF_BEGIN_ELE to begin line string element
DO WHILE more segment points to be output
   SIF_OUTPUT_PTS to output points from this segment record
   IF segment has continuation record
      GET_MORESEG to input segment continuation record
   ENDIF
ENDDO
SIF_END_ELE to end the element
```

B.2.4.3.13 Output Subfeature

Output Subfeature (OUTPUT_SUB_FEA) generates output SIF commands to represent an areal subfeature. Open subfeatures are output as line string type elements. Closed subfeatures consisting of a single segment are output as shape type elements. Closed subfeatures consisting of more than one segment are output as complex shape type elements, where each segment is a component line string.

INPUT: The subfeature information established by BUILD_SUB_FEAS is input. Segments are read from the internal data set segment file.

OUTPUT: SIF commands are generated.

LOGIC:

```
IF subfeature contains only 1 data point
   LOGBADFEA to report error
   INPUT_SEG to input segment
   SIF_SYMBOL to output ERRP symbology
ELSEIF subfeature is open
   LOGBADFEA to report error
   DO FOR number of segments in subfeature
      INPUT_SEG to input segment
      OUTPUT_SEG to output points with ERRL symbology
   ENDDO
ELSE
   set for solid or hole for inner/outer subfeature
   INPUT_SEG to input segment for subfeature
   IF subfeature has >1 segment
      SIF_BEGIN_CSHAPE to begin complex shape
   ENDIF
   DO FOR number of segments in subfeature
      OUTPUT_SEG to output points with feature symbology
      IF not last segment in subfeature
         INPUT_SEG to input next segment
      ENDIF
   ENDDO
   IF complex shape
      SIF_END_CSHAPE to end complex shape
   ENDIF
ENDIF
```

B.2.4.3.14 SLF to SIF Parameters

SLF to SIF Parameters (SLF2SIF_PARAMS) gets input parameters as symbols from the conversion command procedure.

INPUT: Symbols are obtained from the command procedure.

OUTPUT: Input and output file names, the scale factor and the default symbology flag are returned.

B.2.4.4 Define/Submit Job

Define/Submit Job (DEFSUBJOB) is a VMS Command Procedure which interfaces to the user to define a conversion job and submit the defined job for execution. Lexical functions are used to determine information about the current process and specified files.

Upon user request, the conversion job (SLI_COM:CONVJOB.COM) is submitted to the default batch queue. The name of the temporary parameter file is supplied as a parameter to the batch job command procedure. The batch job command procedure reads this file, recognizes the function type and reads its parameters from the file and runs the program. The program obtains its input parameters from the value of the symbols in the conversion job command procedure. When the batch job command procedure detects the end of the temporary parameter file, it deletes the file and exits.

INPUT: The user identifies the functions to be performed as part of the conversion job and is requested to supply the required parameters.

OUTPUT: User supplied parameters are written to a temporary parameter file in the user's current directory. This temporary file contains the name of the function type followed by the user-supplied parameters for that process.

Symbols defined for each of the Conversion processes include:

BUILD SYMBOL LIBRARY
- scfnam — Symbol Correlation File specification
- sclspec — Symbol Correlation Library specification

CONVERT SLF TO SIF
- dnam — Data Set specification
- sclspec — Symbol Correlation Library specification
- defsym — Default symbology flag
- scale — Scale factor

CONVERT SIF TO SLF
- dnam — Data Set specification
- scale — Scale factor

PROCESS SLF
- toslf — Internal to SLF flag
- dnam — Data Set specification
- scale — Scale factor

TAPE I/O
- tin — Tape input flag
- slftap — SLF tape flag
- tapnam — Tape file name
- dnam — Data set specification
- occnum — SLF file occurrence number

LOGIC:

```
ClearScreen
Get jobnam and print report pflag from user
OPEN jobnam.TMP temporary file
DO WHILE Cfunction NOT confirmed quit OR submit
   ClearScreen
   Display menu of valid functions and prompt user for Cfunction
   IF Cfunction = a process step
      WRITE process step type to jobnam.TMP
      Get user parameters specific to process
      WRITE parameters to jobnam.TMP
      Save defaults for next process
   ELSEIF Cfunction = Submit
      CLOSE temporary file
      SUBMIT conversion job with jobnam name; lognam.log file and
         parameters input from temporary file
   ELSEIF Cfunction = Quit
      Ask user for confirmation
   ENDIF what type of function
ENDDO until user says submit or confirmed quit
```

B.2.4.5 Process SLF

Process SLF Input (PROCESSLF) converts between internal data set format and disk resident SLF. A data set may be converted from internal format to SLF or from SLF to internal format. PROCESSLF is invoked from the conversion command procedure. Statistics for the process summary report and data integrity checking arrays are maintained in labeled common.

INPUT: A disk resident data set is input.

OUTPUT: A disk resident data set in the desired form is output.

LOGIC:

```
PROCESSLF_PARAMS to get input parameters
IF internal to SLF
   DM_CRENUL to create empty SLF file
   DM_OPEN to open SLF output file
   INT2SLF to convert Internal to SLF
ELSE
   DM_OPEN to open input SLF file
   SLF2INT to convert SLF to Internal
ENDIF
```

B.2.4.5.1 ProcessSLF Parameters

ProcessSLF Parameters (PROCESSLF_PARAMS) gets input parameters from the conversion command procedure.

INPUT: Symbols are obtained from the command procedure.

OUTPUT: Input and output file names and the function flag are returned.

B.2.4.5.2 Convert Internal to SLF

Convert Internal to SLF (INT2SLF) reads an internal format data set from disk and creates an SLF data file on disk. Subroutines OUT_SLF_DSI, OUT_SLF_SEG and OUT_SLF_FEA are invoked to output DSI, segment and feature data, respectively. Statistics are reported to the process log.

INPUT: The internal format data set consists of DSI, Feature, Segment and Header files.

OUTPUT: An SLF data set is created.

B.2.4.5.3 Build SLF Record

Build SLF Record fills the 1980 byte SLF output buffer from an input buffer containing character data.

INPUT: A character buffer to be output is supplied by the calling routine.

OUTPUT: Data is moved from the input buffer to the SLF output buffer. The SLF record is written to the disk file when it is filled.

LOGIC:

```
DO WHILE not done
   IF point past end of input buffer
      Set done true
   ELSE
      IF output pointer past end of buffer
         DM_PUT to output record to SLF file
         DM_CASCI to convert block number from ASCII to Integer
         Increment SLF block number
         INT4_ASCII to convert block number from integer to ASCII
      ENDIF
   ENDIF
   Move byte from input buffer to output buffer
ENDDO
```

B.2.4.5.4 Convert BYTE to ASCII

Convert BYTE to ASCII (BYTE_ASCII) converts an input byte value to ASCII. The PAD_FIELD routine is used to zero or blank fill and justify the field.

INPUT: The byte to be converted is input.

OUTPUT: The ASCII value is returned.

B.2.4.5.5 Convert Word to ASCII

Convert Word to ASCII (INT2_ASCII) converts an input word value to ASCII. The PAD_FIELD routine is used to zero or blank fill and justify the field.

INPUT: The Integer*2 value to be converted is input.

OUTPUT: The ASCII value is returned.

B.2.4.5.6 Convert Longword to ASCII

Convert Longword to ASCII (INT4_ASCII) converts an input longword value to ASCII. The PAD_FIELD routine is used to zero or blank fill and justify the field.

INPUT: The Integer*4 value to be converted is input.

OUTPUT: The ASCII value is returned.

B.2.4.5.7 Output SLF DSI

Output SLF DSI (OUT_SLF_DSI) converts the internal DSI file to SLF.

INPUT: The internal format DSI file is read.

OUTPUT: SLF DSI information is output via labeled common.

LOGIC:

Initialize output buffer to blanks
DM_OPEN input internal DSI file
DM_GETSEQ to get primary DSI record
Add SLF DSI record sentinels
HDRIN to input internal HDR file information
INT4_ASCII to convert header file counts to ASCII
Store counts from header file in SLF
BLD_SLF_REC to output fixed portion of SLF DSI
BLD_SLF_REC to output SLF DSI registration point information
BLD_SLF_REC to output SLF DSI accuracy information
PAD_SLF_RECORD to pad remainder of SLF DSI record
DM_CLOSE to close input DSI file

B.2.4.5.8 Output SLF Features

Output SLF Features (OUT_SLF_FEA) converts each feature in the the internal feature file to SLF.

INPUT: Features are input from the internal format feature file.

OUTPUT: Reformatted features are output to the SLF file. The SLF output buffer is retained in labeled common.

LOGIC:

```
DM_OPEN to open input feature file
DM_GETSEQ to read first feature record
DO WHILE not end of file
   BLD_SLF_REC to output fixed portion of feature
   DO FOR each feature header entry
      BLD_SLF_REC to output feature header entry
   ENDDO
   BLD_SLF_REC to output number of segment entries
   DO FOR number of segment entries
      Convert feature segment entry to SLF form
      BLD_SLF_REC to output feature segment entry
   ENDDO
   DM_GETSEQ to read next feature record
ENDDO
PAD_SLF_RECORD to pad remainder of last feature record
DM_CLOSE to close feature file
```

B.2.4.5.9 Output SLF Segments

Output SLF Segments (OUT_SLF_SEG) converts segments from the internal format segment file to SLF.

INPUT: Segments are read from the internal format segment file.

OUTPUT: Reformatted segments are output to the SLF file. The SLF output buffer is retained in labeled common.

LOGIC:

```
DM_OPEN to open input segment file
DM_GETSEQ to read first segment record
DO WHILE not end of file
   IF primary segment record
      BLD_SLF_REC to output fixed portion of segment record
      DO FOR number of feature entries
         BLD_SLF_REC to output segment feature entry
      ENDDO
      BLD_SLF_REC to output count of points in segment
   ELSE
      Compute number of points in continuation record
   ENDIF
   DO FOR number of points in segment record
      BLD_SLF_REC to output converted X value
      BLD_SLF_REC to output converted Y value
   ENDDO
   DM_GETSEQ to read next segment record
ENDDO
PAD_SLF_RECORD to pad remainder of last segment record
DM_CLOSE internal segment file
```

B.2.4.5.10 Pad Field

Pad Field (PAD_FIELD) fills a field with blanks or zeros. The field may be right or left justified.

INPUT: Inputs include the field to be padded, its length and right/left justify and zero/blank fill indicators.

OUTPUT: The field is justified and filled according to the inputs.

B.2.4.5.11 Pad SLF Record

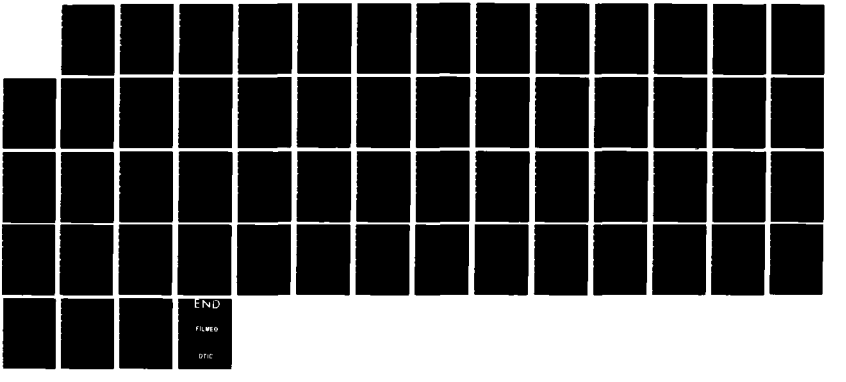
Pad SLF Record (PAD_SLF_RECORD) fills the remainder of a 1980 byte SLF output buffer with octal 177, as per SLF specifications. The DM_PUT service is used to write the record to the SLF file.

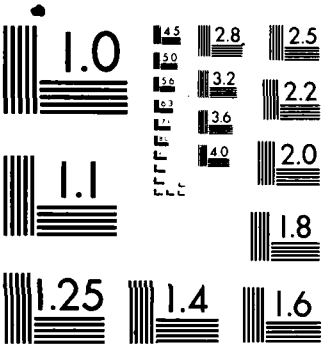
INPUT: The SLF buffer is input via labeled common.

OUTPUT: The record is output to the SLF file.

AD-A161 086 SOFTWARE CONVERSION OF STANDARD LINEAR FORMAT (SLF) TO STANDARD INTERCHANGE FORMAT (SIF)(U) MEASUREMENT CONCEPT CORP ROME NY K J SANDS ET AL. JUL 85 ETL-0394 2/2

UNCLASSIFIED DAAK70-83-C-0174 F/G 8/2 NL





B.2.4.5.12 Convert SLF to Internal

Convert SLF to Internal (SLF2INT) reads a SLF data set from disk and creates an internal format data set on disk. The internal format data set consists of a series of files which are structured to facilitate conversion from SLF to SIF. Statistics are reported to the process log. The input SLF buffer, data integrity checking arrays, statistics for the process log and status flags are maintained in labeled common.

INPUT: The SLF data set is read from disk.

OUTPUT: The internal format data set which is created consists of DSI, Feature, Segment and Header files.

LOGIC:

```
DM_GETSEG to read first SLF record
DO WHILE not end of file
   IF DSI record
      PROCESS_DSI to output DSI
   ELSEIF Segment record
      PROCESS_SEG to output segments
   ELSEIF Feature record
      PROCESS_FEA to output features
   ELSEIF Text record
      PROCESS_TXT to output text
   ELSE
      Report unknown record type error
   ENDIF
   DM_GETSEG to read next input record
ENDDO
CHECK_SEGFEA to verify segment-feature relationships
HDROUT to output data set header file
DM_CLOSE input SLF file
```

B.2.4.5.13 Build Logical Record

Build Logical Record (BLDLOGREC) extracts the next logical record from the SLF input buffer. When the SLF input buffer is exhausted, the DM_GETSEQ service is used to read the next record from the SLF input file.

INPUT: The SLF and internal format record buffers are supplied.

OUTPUT: The logical record is moved from the SLF buffer to the supplied internal buffer.

B.2.4.5.14 Verify Segment-Feature Relationships

Verify Segment-Feature Relationships (CHECK_SEGFEA) tests for the consistency of feature-segment relationships. Arrays are used to record the presence/absence of segments and features and segment-feature cross references. In each case, one bit represents the corresponding segment or feature number (from 1 to 65535); the bit is set to one (1) to indicate the presence (or reference) of a segment or feature.

The input SLF data set is processed sequentially (i.e., segments then features). As segments are processed, a segment-segment (SegSeg) array is revised to indicate all segments present in the input data set. A segment-feature (SegFea) array is also updated to indicate each feature referenced by any segment.

As features are processed, a feature-feature (FeaFea) array is revised to indicate all features present in the input data set. As each feature is processed, any segment referenced by the feature which is not in the SegSeg array indicates that the feature references a non-existent segment.

Once all features have been processed, differences in the SegFea and FeaFea arrays indicate a segment which references a non-existent feature. At this point, however, the segment number is unknown. This error will never be encountered by the conversion process, which is feature-oriented.

INPUT: An operation code is input to request a test for feature-segment relationships or a test for non-existent features. For testing feature-segment relationships, the segment ID is also input.

OUTPUT: Inconsistencies are reported to the process log and flagged to the calling routine through a returned condition value.

LOGIC:

```
IF operation = check feature segment relation
   DM_CASCI to convert segment ID to integer
   Compute array bit location for specified segment ID
   IF bit for this segment is set
      all is ok
   ELSE
      Report feature references non-existent segment and
   ENDIF
ELSEIF operation = check for missing features
   DO FOR each possible feature ID
      IF SegFea word NOT= FeaFea word
         DO FOR each bit in word
            IF SegFea bit NOT= FeaFea bit
               Compute feature ID corresponding to current bit
               Report non-existent feature referenced
               Record feature ID in BadFea array for header file
            ENDIF
         ENDDO
      ENDIF
   ENDDO
ENDIF
```

B.2.4.5.15 Get Accuracy Outline Records

Get Accuracy Outline Records (GETACCREC) moves all accuracy outline records from the input SLF data set DSI record to the output internal data set DSI file.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: Accuracy information is stored in the internal DSI record in labeled common.

LOGIC:

```
BLDLOGREC to build fixed portion of ACC record
DM_CASCI to convert accuracy record count
DO FOR number of accuracy records
   BLDLOGREC to build accuracy record
   DM_CASCI to convert number of accuracy points
   DO FOR number of accuracy points
      BLDLOGREC to build accuracy point
      move point into temporary buffer
      Move accuracy point to output
   ENDDO
ENDDO
```

B.2.4.5.16 Get Registration Points

Get Registration Points (GETREGREC) moves all registration points from the input SLF data set DSI record to the output internal data set DSI file.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: Registration points are stored in the internal DSI record in labeled common.

LOGIC:

```
BLDLOGREC to build fixed portion of registration point
DM_CASCI to convert registration point count
DO FOR number of registration points
   BLDLOGREC to build registration point record
ENDDO
```

B.2.4.5.17 Get Feature Segment Entries

Get Feature Segment Entries (GET_FEASEG_ENTS) inputs all feature segment entries for an SLF input feature.

INPUT: The SLF input buffer is input via labeled common.

OUTPUT: Feature segment entries are stored in the feature record in labeled common. Data integrity checking arrays are revised.

```
LOGIC:

BLDLOGREC to extract number of segments in feature
DM_CASCI to convert number of segments to integer
IF number of segments > 0
   DO FOR number of segments in feature
      BLDLOGREC to extract feature segment entry
      CHECK_SEGFEA to verify that segment exists
      Move feature segment entry into output feature record
   ENDDO
ELSE
   Report error feature has no segments
ENDIF
```

B.2.4.5.18 Get Feature Header Entries

Get Feature Header Entries (GET_FHDRS) header entries for an SLF input feature.

INPUT: The SLF input buffer is input via labeled common.

OUTPUT: Feature header entries are stored in the feature record in labeled common.

```
LOGIC:

DM_CASCI to convert number of header records to integer
IF number of header records > 0
   DO FOR number of header records
      BLDLOGREC to extract feature header
      Move feature header to feature buffer
   ENDDO
ELSE
   Report error feature has no header data
ENDIF
```

B.2.4.5.19 Get Segment Feature Entries

Get Segment Feature Entries (GET_SEGFEA_ENTS) inputs all segment feature entries for an SLF input segment.

INPUT: The SLF input buffer is input via labeled common.

OUTPUT: Segment feature entries are stored in the segment record in labeled common. Data integrity checking arrays are revised.

LOGIC:

```
DM_CASCI to convert number of feature entries to integer
IF number of feature entries > 0
   DO for number of feature entries
      BLDLOGREC to extract feature entry
      DM_CASCI to convert feature ID to integer
      Move feature entry into segment record buffer
      SETBIT to set bit in SegFea for segment-feature ID
   ENDDO
ELSE
   Report error segment belongs to no features
ENDIF
```

B.2.4.5.20 Get Segment Coordinates

Get Segment Coordinates (GET_SEG_PTS) inputs all coordinate sets (X,Y[,Z]) for an SLF segment record and stores the points in the internal format segment record.

INPUT: The SLF input buffer is input via labeled common.

OUTPUT: Segment points are stored in the segment record in labeled common.

```
LOGIC:

BLDLOGREC to extract number of points in SLF segment
DM_CASCI to convert number of points to integer
IF number of points > 0
   Compute point size with/without z values
   Set segment continuation ID = 0
   DO WHILE more points
      Initialize pointers, number of points to process
      DO WHILE number of points to process notdone
         BLDLOGREC to extract segment coordinate
         DM_CASCI to reformat coordinate into internal format
         Revise segment bounding rectangle
         Move coordinate into internal segment buffer
      ENDDO
      DM_PUT segment record to internal file
      Set continuation ID if more points to be output
   ENDDO
ELSE
   Report error segment has no points
ENDIF
```

B.2.4.5.21 Process DSI

Process DSI (PROCESS_DSI) creates and populates the internal data set DSI file with data from the input SLF record.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: The internal format DSI file is created, the DSI record is filled and output to the file.

```
LOGIC:

IF valid to input DSI
   IF first block from SLF data set
      CRE_DSF to create output DSI file
   ENDIF
   BLDLOGREC to extract fixed portion of DSI record
   DM_CASCI to convert DSI counts to integer values
   GETREGREC to get registration points
   GETACCREC to get accuracy outlines
   DM_PUT to output DSI record
ELSE
   Report error input SLF record out of sequence
ENDIF
DM_CLOSE output DSI file
Set flag for subsequent DSI invalid
```

B.2.4.5.22 Process Features

Process Features (PROCESS_FEA) creates and populates the internal data set feature file with data from the input SLF record.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: The internal format feature file is created. Each input feature record is converted to internal format and output to the file.

LOGIC:

```
IF valid to get features input
   IF first feature record
      CRE_DSF to create internal format feature file
   ENDIF
   DO FOR total number of features in input SLF data set
      BLDLOGREC to extract fixed portion of feature record
      DM_CASCI to convert feature ID to integer
      GET_FHDRS to build feature header entries
      GET_FEASEG_ENTS to build feature segment entries
      IF data set product type = DACD
         Move feature header buffer to output
      ELSEIF data set product type = DFAD1
         Extract FIDC from header; move buffer to output
      ELSE
         Output blank feature header
      ENDIF
      DM_PUT to output feature recrod
      SETBIT to record feature ID in FeaFea array
   ENDDO
ELSE
   Report error input SLF record out of sequence
ENDIF
Set flag for subsequent features invalid
```

B.2.4.5.23 Process Segments

Process Segments (PROCESS_SEG) creates and populates the internal data set segment file with data from the input SLF record.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: The internal format segment file is created. Each input segment record is converted to internal format and output to the file.

LOGIC:

```
IF valid to get segments input
   IF first segment record
      CRE_DSF to create temporary sequential segment file
   ENDIF
   DO FOR total number of segments in input SLF data set
      BLDLOGREC to extract fixed portion of segment record
      DM_CASCI to convert segment ID to integer
      GET_SEGFEA_ENTS to build segment feature entries
      GET_SEG_PTS to build segment points and output record
      SETBIT to record segment ID in SegSeg array
   ENDDO
   DM_CLOSE to close output segment file
   DM_CREPOP to convert sequential segment file to indexed
   DM_DELFIL to delete sequential segment file
ELSE
   Report error input SLF record out of sequence
ENDIF
Set flag for subsequent segments invalid
```

B.2.4.5.24 Process Text

Process Text (PROCESS_TXT) creates and populates the internal data set text file with data from the input SLF record.

INPUT: The SLF buffer is input via labeled common.

OUTPUT: The internal format text file is created. Each input text record is output to the file.

```
LOGIC:

IF valid to get text input
   IF first text record
      CRE_DSF to create text file
   ENDIF
   BLDLOGREC to extract character count
   DM_CASCI to convert character count to integer
   IF character count > 0
      Compute number of 80 byte text subrecords
      DO FOR number of text subrecords
         BLDLOGREC to extract 80 byte text subrecord
         DM_PUT to output text record
      ENDDO
   ENDIF
ELSE
   Report error input SLF record out of sequence
ENDIF
```

B.2.4.5.25 Set Data Check Bit

Set Data Check Bit (SETBIT) sets bits for subsequent data integrity testing. The indicated bit is set on in the specified array (SegSeg, SegFea or FeaFea).

INPUT: The segment or feature ID to be set on is input, as is the array to be revised.

OUTPUT: The input ID is used to compute the bit location within the array. The bit is set (i.e., to one).

B.2.4.6 Tape Input/Output

Tape Input/Output (TAPEIO) provides the capability to transfer files between magnetic tape and disk. Both DMA SLF and Intergraph SIF are supported. The main program gets the input parameters as symbols from the command procedure and allocates and mounts the tape. It then invokes the appropriate subroutine to perform the transfer and dismounts and deallocates the tape. The operator is prompted to physically mount the tape.

INPUT: The input and output data set names are input.

OUTPUT: A disk image of an SLF or SIF data set is transferred to/from magnetic tape.

LOGIC:

```
GET_PARAM to get input parameters
IF TapeIn THEN
   MTALLMNT to Allocate, mount and assign tape for read
ELSE
   MTALLMNT to Allocate, mount and assign tape for write
ENDIF
IF SLF THEN
   IF TapeIn THEN
      SLFTAPEIN to move SLF DataSet from Tape to Disk
   ELSE
      SLFTAPOUT to move SLF DataSet from Disk to Tape
   ENDIF
ELSEIF TapeIn THEN
   SIFTAPEIN to move SIF DataSet from Tape to Disk
ELSE
   SIFTAPOUT to move SIF DataSet from Disk to Tape
ENDIF
MTDISMDEALL to rewind, dismount, deallocate tape
Report completion message, timestamp, statistics
```

B.2.4.6.1 SLF Tape In

SLF Tape Input (SLFTAPEIN) reads an input tape to create a disk image of a SLF data set. The user specified SLF data set is to be located on the tape. Each record of the located data set is copied to a disk file.

INPUT: The tape data set name, tape data set name length, disk data set name, disk data set name length, magtape channel number for I/O, and the occurrence number of the data set on tape are passed as parameters. The file of SLF commands is read from tape.

OUTPUT: The number of records found is returned. The file of SLF commands is written to disk.

```
LOGIC:

MT_READ to read tape Volume Label
FIND_DS to find DataSet
IF DataSet found THEN
   MOVE_DSMT2D to move DataSet from tape to disk
ENDIF



FIND_DS:

DO WHILE NOT End-Of-Tape AND DataSet NOT Found
   MT_READ to read Header Record from tape
   Get data set name from Header Record
   IF data set is desired DataSet THEN
      Tell caller DataSet found
   ELSE
      MT_SKIPFILE to skip data set by skipping 3 EOFs
   ENDIF
ENDDO



MOVE_DSMT2D:

DM_CRENUL to create empty DataSet SLF file on disk
DM_OPEN to open file on disk
MT_READ to read first record from tape
DO WHILE NOT End-Of-File tape AND Success
   DM_PUT to output record to disk file
   MT_READ to read next record from tape
ENDDO
DM_CLOSE to close file on disk
```

B.2.4.6.2 SLF Tape Output

SLF Tape Output (SLFTAPOUT) generates a SLF tape from a disk-resident SLF file.

INPUT: The SLF data set is read from disk.

OUTPUT: A SLF tape containing the input data set is created.

LOGIC:

```
DM_OPEN to open SLF disk input file
MT_WRITE to write volume label tape
MT_WRITE to write tape header record to tape
MT_WRITEEOF to write end of file mark on tape
DM_GETSEQ to read first record from disk
DO WHILE not end of file
   MT_WRITE to output record to tape
   DM_GETSEQ to read next record from disk
ENDDO
MT_WRITEOF to write 3 end of file marks to tape
DM_CLOSE to close SLF input file
```

B.2.4.6.3 SIF Tape Inpput

SIF Tape Input (SIFTAPEIN) reads an input SIF tape and creates a disk image of the SIF data set.

INPUT: The SIF file is read from tape.

OUTPUT: A SIF file is created on disk.

LOGIC:

```
DM_CRENUL to create empty SIF file on disk
DM_OPEN to open SIF disk output file
MT_READ to read first record from tape
DO WHILE not end of file
   DM_PUT to output record to SIF output file
   MT_READ to read next record from tape
ENDDO
DM_CLOSE to close SIF output file
```

B.2.4.6.4 SIF Tape Output

The SIF Tape Output process generates a SIF tape from a SIF file on disk.

INPUT: A file of SIF commands is read from disk.

OUTPUT: A SIF tape, containing a single file, is created.

LOGIC:

```
DM_OPEN to open SIF disk input file
DM_GETSEQ to read first record from disk
DO WHILE not end of file
   MT_WRITE to output record to tape
   DM_GETSEQ to read next record
ENDDO
MT_WRITEOF to write 2 end of file marks tape
DM_CLOSE to close SIF input file
```

B.2.4.7 Data Management Services

The Data Management Services (DMSERVS) provide standardized mechanisms for the conversion software to perform file and record manipulations. Each service is defined as a function, reports any errors encountered and returns a condition value which indicates that the operation was a success or failure. The services include:

DM_CASCI	Convert ASCII to Integer Service
DM_CBYTASC	Convert BYTE to ASCII Service
DM_CI2ASC	Convert INTEGER*2 to ASCII Service
DM_CI4ASC	Convert INTEGER*4 to ASCII Service
DM_CLOSE	Close File Service
DM_CRENUL	Create Null File Service
DM_CREPOP	Create Populated File Service
DM_DELFIL	Delete File
DM_GEOABS	Convert Geographics to Absolute
DM_GETCKY	Get record by Character Key
DM_GETIKY	Get record by Integer Key
DM_GETSEG_ID	Get Segment by ID
DM_GETSEQ	Get Sequential Service
DMIAPPKEY	Append Key to message
DMIFSPEC	Data Management Internal File Spec Parser
DM_OPEN	Open File Service
DM_PUT	Put Record Service
DM_PUTIFE	Put Indexed Feature
DM_UPDATE_IFE	Update Indexed Feature

B.2.4.7.1 DM_CASCI

Convert ASCII to Integer (DM_CASCI) converts an input character string to an integer byte, word or longword as specified by an input parameter. A zero is returned for a zero-length string.

B.2.4.7.2 DM_CBYTASC

Convert BYTE to ASCII (DM_CBYTASC) uses the FAO system service to convert the input byte value to a character variable. The length of the resulting character string is returned.

B.2.4.7.3 DM_CI2ASC

Convert INTEGER*2 to ASCII (DM_CI2ASC) uses the FAO system service to convert the input INTEGER*2 value to a character variable. The length of the resulting character string is returned.

B.2.4.7.4 DM_CI4ASC

Convert INTEGER*4 to ASCII (DM_CI4ASC) uses the FAO system service to convert the input INTEGER*4 value to a character variable. The length of the resulting character string is returned.

B.2.4.7.5 DM_CLOSE

The Close File Service (DM_CLOSE) closes the file associated with the specified logical unit. The logical unit is released using the LIB$FREE_LUN run time library routine.

B.2.4.7.6 DM_CRENUL

Create Null File (DM_CRENUL) invokes the FDL$CREATE utility to create a null file. The DMIFSPEC service is used to extract the file type from the input file specification. An FDL file for the specified file type is used by FDL$CREATE.

B.2.4.7.7 DM_CREPOP

Create Populated File (DM_CREPOP) creates a populated file from the specified input file. Creating a file does not open the file. The DMIFSPEC service is used to extract the file types from the supplied input and output file specifications. The LIB$SPAWN run time library routine is invoked to COPY the input file to the output file (for same file types) or to CONVERT the input file to the output file using an FDL file (for differing file types).

B.2.4.7.8 DM_DELFIL

Delete File (DM_DELFIL) deletes a file. The file must be closed. The specification of the file to be deleted is input. Logical unit 1 is used to open and close the file with the DISPOSE='DELETE' qualifier.

B.2.4.7.9 DM_GEOABS

Convert Geographics to Absolute (DM_GEOABS) converts an input geographic coordinate from ASCII to a longword integer representing the geographic value in .01 arc seconds. An input flag indicates if the value is a latitude or longitude. Each portion of the input coordinate (degrees, minutes, seconds, hundreds of seconds and hemisphere) is converted to integer and tested for a legal range of values. Coordinates in the Southern or Western hemispheres are represented by negative values.

B.2.4.7.10 DM_GETCKY

Get record by Character Key (DM_GETCKY) retrieves an existing record from an indexed file using a character key value. The file must be open. The key of reference and the character key value are supplied as inputs. The record is read into the specified user buffer. Errors resulting from reading variable length records are ignored.

B.2.4.7.11 DM_GETIKY

Get record by Integer Key (DM_GETIKY) retrieves an existing record from an indexed file using an integer key value. The file must be open. The key of reference and the integer key value are supplied as inputs. The record is read into the specified user buffer. Errors resulting from reading variable length records are ignored.

B.2.4.7.12 DM_GETSEG_ID

Get Segment by ID (DM_GETSEG_ID) retrieves an existing segment record from an internal format data set segment file. Retrieval is performed on the basis of the supplied segment ID-segment continuation number. The file must be open. The DM_GETIKY service is used to read the segment into the CSEGREC common area.

B.2.4.7.13 DM_GETSEQ

Get Sequential Service (DM_GETSEQ) retrieves the next sequential record from a sequential or indexed file into the user supplied buffer. The file must be open. Positioning for an indexed file is established by an indexed read. Subsequent sequential reads will return records selected by the next ascending value for the current key. Errors resulting from reading variable length records are ignored.

B.2.4.7.14 DMIAPPKEY

Append Key to message (DMIAPPKEY) uses the DM_CI2ASC service to convert the input key value to character. The specified condition value is then reported via the CMSG service with the converted key value supplied as append text.

B.2.4.7.15 DMIFSPEC

Data Management Internal File Spec Parser (DMIFSPEC) accepts a file specification as input and returns each of the pieces of the file specification as a separate variable.

B.2.4.7.16 DM_OPEN

Open File (DM_OPEN) opens a file for access. Inputs include the file specification and desired access, format rules and record type. If write access is not requested, the file is opened readonly. The LIB$GET_LUN run time library routine is used to assign a logical unit for the file.

B.2.4.7.17 DM_PUT

Put Record (DM_PUT) writes the specified number of bytes from the user supplied buffer as a new record in an indexed or a sequential file. The file must be open.

B.2.4.7.18 DM_PUTIFE

Put Indexed Feature (DM_PUTFEA) writes a new feature record from the CIFEREC common area to the internal format feature file. The file must be open. The length of the feature record, in bytes, is computed.

B.2.4.7.19 DM_UPDATE_IFE

Update Indexed Feature (DM_UPDATE_IFE) rewrites an existing feature record from the CIFEREC common area to the internal format feature file. The file must be open. The length of the feature record, in bytes, is computed.

B.2.4.8 Magnetic Tape Services

The Magnetic Tape Services (MTSERVS) provide standardized mechanisms for the conversion software to perform tape manipulation functions. Each service is defined as a function, reports any errors encountered and returns a condition value which indicates that the operation was a success or failure. The services include:

MT_ALLMNT	Allocate and Mount Tape
MT_DISMDEALL	Dismount and Deallocate Tape
MT_READ	Read Tape
MT_SKIPFILE	Skip End of File mark on tape
MT_WRITE	Write Tape
MT_WRITEOF	Write End of File Mark to tape

B.2.4.8.1 MT_ALLMNT

MagTape Allocate and Mount (MT_ALLMNT) uses the SYS$ALLOC system service to allocate a tape drive. A mount request is issued for the specified volume using the SYS$MOUNT system service. The /FOREIGN and optional /NOWRITE flags are specified. Once the mount is completed, a channel is assigned to the tape for subsequent I/O. This routine assumes that the logical name MTDEV is defined as a logical name for whatever type of tape drive device (e.g., MT, MM, MS) happens to be configured on the system. Inputs include the volume name, the length of the tape records, in bytes, and a read/write access indicator. A channel number assigned to the tape for subsequent I/O is returned, along with the logical name associated with the tape device. Mount parameters are defined from the VMS $MNTDEF Macro. These are stored in a mount item-list as described in the VMS System Services documentation. If an error occurs, the SYS$DISMOU and SYS$DALLOC system services are used to dismount and deallocate the tape, respectively.

B.2.4.8.2 MT_DISMDEALL

MagTape Dismount and Deallocate (MT_DISMDEALL) issues a SYS$QIOW system service request to rewind the tape and a SYS$DASSGN to disassociate the channel. The tape is dismounted and deallocated using the SYS$DISMOU and SYS$DALLOC system services, respectively. The routine tries to keep going on errors to finish cleanup before exiting. The channel associated with the tape, its logical name and physical device name are input.

B.2.4.8.3 MT_READ

MagTape Read (MT_READ) uses the SYS$QIOW system service to read one record from a mounted magtape assigned to the specified channel into the user supplied buffer.

B.2.4.8.4 MT_SKIPFILE

MagTape Skip End Of File (MT_SKIPFILE) uses the SYS$QIOW system service to advance the tape associated with the specified channel forward past the indicated number of End-Of-File tape marks. An error is returned if the end of the tape is encountered before the specified number of tape marks are found.

B.2.4.8.5 MT_WRITE

MagTape Write (MT_WRITE) uses the SYS$QIOW system service to write one record from the user supplied buffer to a mounted magtape assigned to the specified channel.

B.2.4.8.6 MT_WRITEOF

MagTape Write End Of File (MT_WRITEOF) uses the SYS$QIOW system service to write one EOF mark on the tape associated with the specified channel.

B.2.4.9 Miscellaneous Services

The Miscellaneous Services (MISCSERVS) provide miscellaneous services for the conversion software. Each service is defined as a function, reports any errors encountered and returns a condition value which indicates that the operation was a success or failure. The services include:

CRE_DSF	Create Data Set File
HDRIN	Input HDR record
HDROUT	Output HDR record

B.2.4.9.1 Create Data Set File

Create Data Set File (CRE_DSF) creates and opens an internal format data set component file. The file name and type are input. The DM_CRENUL and DM_OPEN services are used to create and open the file, respectively.

B.2.4.9.2 Input Header

Input Header (HDRIN) reads the internal format data set header file into a labeled common area. The file name is input. The DM_OPEN, DM_GETSEQ and DM_CLOSE services are used to open the file, read its contents and close the file, respectively.

B.2.4.9.3 Output Header

Output Header (HDROUT) creates and outputs the internal format data set header file. The data set header contents are input via labeled common. The file specification is input. The CRE_DSF, DM_PUT and DM_CLOSE services are used to create and open, write and close the file, respectively.

B.2.4.10 SIF Services

The SIF Services (SIFSERVS) provide common services for creation and output of properly formatted SIF command. Each service is defined as a function, reports any errors encounterd and returns a condition value which indicates that the operation was a success or failure. The services include:

BLD_ATTR_BUF	Build Attribute Values
I4STUFF	Move bytes into longword
SIF_ATTR	Assign Attribute Information
SIF_BEGIN_CSHAPE	Initialize Complex Shape
SIF_BEGIN_ELE	Initialize SIF Element
SIF_END	SIF Wrapup Function
SIF_END_CSHAPE	End Complex Shape
SIF_END_ELE	End Current Element
SIF_GRA_CHAR	Establish SIF Graphic Characteristics
SIF_INIT	SIF Initialization
SIF_OUTPUT_ACC	Output Accuracy Outline
SIF_OUTPUT_CMD	Output SIF Command
SIF_OUTPUT_PTS	Output Current Element
SIF_REVERSE_PTS	Reverse Points
SIF_SYMBOL	Output SIF Symbol
SIF_TEXT	Output Text

B.2.4.10.1 Build Attribute Values

Build Attribute Values (BLD_ATTR_BUF) constructs or revises an attribute values buffer from input character data extracted from internal format DSI or feature information. The attribute values are subsequently output within an Associative Values command.

The input attribute ID is converted to a character string and output, followed by the specified number of attribute characters. A delimitor is output before and after the attribute ID and at the end of the entire string.

B.2.4.10.2 Move bytes into longword

Move bytes into longword (I4STUFF) moves input bytes into a longword.

B.2.4.10.3 Assign Attribute Information

Assign Attribute Information (SIF_ATTR) generates and outputs SIF Identifier and Association commands to establish DSI or feature DMRS attribute information. Attribute values are input via a buffer and the SIF_OUTPUT_CMD service is used to generate SIF commands in the current SIF output buffer. Continuation commands are issued as required.

B.2.4.10.4 Initialize Complex Shape

Initialize Complex Shape (SIF_BEGIN_CSHAPE) initializes the element-in-progress flags in the SIF Common area for the processing of closed, multi-segment areal subfeatures. The segments owned by such a subfeature are output as a series of open line strings. The subfeature itself is treated as a large complex shape (solid or hole) element. Desired graphic characteristics are established from the Symbol Library Record input via labeled common and are output via the SIF_GRA_CHAR service. A begin complex command element is output via the SIF_OUTPUT_CMD service.

B.2.4.10.5 Initialize SIF Element

Initialize SIF Element (SIF_BEGIN_ELE) defines desired characteristics from the Symbol Library Record input via labeled common, invokes SIF_GRA_CHAR to establish the desired graphic characteristics as current, initializes the element-in-progress flags in the SIF Common area. If the segment has more than 100 points, SIF_OUTPUT_CMD is used to output a begin complex element command.

B.2.4.10.6 SIF Wrapup Function

The SIF Wrapup Function (SIF_END) invokes SIF_OUTPUT_CMD to output a pad command to fill the current buffer and, via DM_CLOSE, closes the output SIF file.

B.2.4.10.7 End Complex Shape

End Complex Shape (SIF_END_CSHAPE) invokes SIF_OUTPUT_CMD to output an End Complex Element command to complete processing of a complex shape.

B.2.4.10.8 End Current Element

End Current Element (SIF_END_ELE) uses SIF_OUTPUT_CMD to output any remaining coordinates of an element. For component elements of complex elements, data points are repeated to ensure connectivity. SIF_OUTPUT_CMD is used to output an End Complex Element commmand if a complex line string type element was in process.

B.2.4.10.9 Establish SIF Graphic Characteristics

Establish SIF Graphic Characteristics (SIF_GRA_CHAR) records the desired graphic characteristics as current in the SIF Common area. SIF_OUTPUT_CMD is used to output graphics characteristics commands when the desired characteristics change. To ensure that the Intergraph patterning task will function correctly, a pattern command is output each time SIF_GRA_CHAR is invoked with a non-zero pattern.

B.2.4.10.10 SIF Initialization

SIF Initialization (SIF_INIT) uses DM_CRENUL and DM_OPEN to create and open the specified SIF output file, respectively. The output command buffer is initialized and default graphic characteristics are established.

B.2.4.10.11 Output Accuracy Outline

Output Accuracy Outline (SIF_OUTPUT_ACC) uses DM_GEOABS to convert input coordinates to integer values. The accuracy outline is output as a Solid Shape command by SIF_OUTPUT_CMD.

B.2.4.10.12 Output SIF Command

Output SIF Command SIF_OUTPUT_CMD moves the specified number of bytes from the input SIF command buffer to the SIF file output buffer. When the SIF buffer is filled, it is written to the file using the DM_PUT service.

B.2.4.10.13 Output Current Element

Output Current Element (SIF_OUTPUT_PTS) outputs the specified number of input coordinates as SIF element points. The first point of shape elements is saved to enable subsequent closure. SIF_OUTPUT_CMD is used to output a command header as necessary and points are output.

B.2.4.10.14 Reverse Points

Reverse Points (SIF_REVERSE_PTS) reverses the sequence of points in an internal format segment record. The segment is input and output via labeled common.

B.2.4.10.15 Output SIF Symbol

Output SIF Symbol (SIF_SYMBOL) generates an output SIF Generate Symbol command at the specified location. Desired graphic characteristics are initialized from the Symbol Library Record input via labeled common and established via the SIF_GRA_CHAR service. SIF_OUTPUT_CMD is used to output the Generate Symbol command. If text symbolization is included, an Include Symbol Text command is output using SIF_OUTPUT_CMD.

B.2.4.10.16 Output Text

Output Text (SIF_TEXT) uses SIF_OUTPUT_CMD to generate a SIF Output Text command for the specified text string at the specified location.

B.3 ENVIRONMENT

B.3.1 Equipment And Support Software Environment

The SLF-SIF Conversion software executes on the DEC VAX family of processors under Version 3 or higher of the VMS operating system. The optional VAX-11 FORTRAN product is required.

Input and output Intergraph SIF tapes are compatible with an Intergraph system based on the DEC PDP-11/70 processor with Version 8.5.1 of the Intergraph IGDS and DMRS software. The optional Intergraph Standard Interchange Format (SIF) product is required.

B.3.2 Data Base

The Conversion software does not maintain a data base, per se. Symbolization to be applied is defined by Symbol Correlation Files and associated Symbol Correlation Libraries. All conversion is done using an internal data set format, which is represented in a series of temporary files.

B.3.2.1 General Characteristics

B.3.2.1.1 DMA Standard Linear Format

The DMA Standard Linear Format (SLF) is designed as the vehicle for the exchange on magnetic tape of digital cartographic feature data, consisting of planimetric and/or hypsometric information created in the production of various DMA hydrographic, topographic and aeronautical products. SLF is described in Reference 5.

B.3.2.1.2 Intergraph Standard Interchange Format

Intergraph Standard Interchange Format (SIF) provides a common mechanism for transmittal of graphics and associated data between systems and is described in References 6 and 7.

B.3.2.1.3 Conversion Software Internal Format

An internal SLF data set consists of a series of files; the 9 character data set name is used as the filename. File types include:

- Data Set Identifier (DSI) - Sequential file with variable length records containing input DSI information.
- Feature File (FEA) - Sequential file consisting of variable length feature records.
- Data Set Header (HDR) - Sequential file with fixed length records containing control and summary information.
- Indexed Feature File (IFE) - Temporary indexed feature file used for conversion from SIF consists of variable length feature records.
- Segment File (SEG) - Indexed (RMS) file containing variable length segment records.

In addition, symbolization in defined using an ASCII Symbol Correlation File which is processed to generate an indexed Symbol Specification Library. The Symbol Correlation File is described in the Users Manual.

B.3.2.1.4 External Files

Several external files, resident on the Intergraph system, support creation of SIF data input to the Conversion and manipulation of SIF output from the Conversion software. These include:

- Seed Design File - enables creation of IGDS Design File from SIF
- Cell Library - defines cell elements which implement point and pattern symbology specified via Symbol Correlation File
- Pattern Definition File - describes patterns which implement symbology specified via Symbol Correlation File

- o SIF Environment File - provides parameter information for input to the SIF processors

- o DMRS Data Definition Language File - describes schema of DMRS data base for SIF input to Conversion and SIF output from Conversion. (Note: the two schemas differ.)

B.3.2.2 Organization and Detailed Description

To facilitate software development and maintenance, all "data base" items are defined via FORTRAN Include files and conform to a naming convention. This applies to both internal and external formats. The following prefixes apply to data item names defined via the Include files:

SLF_	DMA Standard Linear Format
ISIF_	Intergraph Standard Interchange Format
C_	SLF to SIF Conversion software internal format
CP_	SLF to SIF Conversion Parameters
CT_	SLF to SIF Conversion Transaction codes

Each record type is defined via a separate Include file. Standard suffixes are appended to data item names to define, for each item:

_sz	Size of item in bytes
_of	Byte offset of item into record (or subrecord)
_ct	Dimension of array type items

Each item in a record is defined in terms of the size and offset of the preceeding item. The first item has an offset of 1. Both size and offset are expressed in bytes. For fixed content records, each data item is equivalenced into a byte buffer defined for the record (e.g., C_segfixed). This is impossible for variable length records. For those variable length records comprised of a variable number of fixed length entries, a small buffer (i.e., subrecord) is defined which contains one entry and the item names are equivalenced to that single entry (e.g., the item C_feasegID is equivalenced into the C_feasegent 'subrecord'). Buffers are defined in the Include files for variable length records. The FORTRAN Include files also define COMMON areas which are typically used with the record type which they define (e.g., C_segfixed).

B.3.2.2.1 DMA Standard Linear Format

DMA Standard Linear Format is described in Reference 5. Include files SLFMTVOL.FOR and SLFMTHDR.FOR define the contents of the tape volume label and volume header, respectively. Include files CSLFFEA.FOR and CSLFSEG.FOR define the contents of a SLF feature and segment record, respectively. The format of a SLF feature header varies by product type. The SLF to SIF conversion software (and the CSLFFEA file) supports the features defined per the DMA Standard Cartographic Feature Digital - Identification Catalog consisting of:

TYPE	SIZE	NAME	DESCRIPTION
CHAR	4	C_sfcode	DMA Standard Feature Code Number
CHAR	3	C_sfat1	Feature attribute code 1
CHAR	3	C_sfat2	Feature attribute code 2
CHAR	30	C_sfpad	Feature pad, unused

B.3.2.2.2 Intergraph Standard Interchange Format

Intergraph Standard Interchange Format is described in References 6 and 7. SIF data contains commands which are generated from/used to generate information in a Data Management and Retrieval System (DMRS) data base. To enable the conversion software to create and interpret this information within the SIF data, the DMRS schema must conform to the structures described below.

The complete DMRS schema for SIF input is described in the GENERIN.DDL Data Definition Language file. The SIF to SLF conversion software recognizes each identifier command as the start of a segment, registration point or the DSI data. To allow derivation of feature-segment relationships, the attributes for each segment describe the feature Left of the segment, the feature Right of the segment and the feature Coincident with the segment. Attributes for each feature include a feature ID value, and the attributes (Feature Code, Feature Attribute 1 and Feature Attribute 2 - as described above). Feature ID values are unique within the data set. Each segment must reference at least one feature; multiple segments may reference the same feature ID.

The complete DMRS schema for SIF output is described in the GENEROUT.DDL Data Definition Language file. The SLF to SIF conversion software generates SIF commands to provide for creation of a DMRS data base in accordance with this schema. The entities in this data base parallel SLF format. Attributes for each feature include the feature ID, feature type (P,L,A), feature header block count, segment count and product type

attribute information. The product type attribute information includes the Feature Code, Feature Attribute 1 and Feature Attribute 2 - as described above.

B.3.2.2.3 Conversion Software Internal Format

The following describes the record format for each of the conversion software internal files.

DSI: The Data Set Identifier is a sequential file containing input DSI information. The DSI is maintained as character data which parallels the SLF. Variable names identify the DSI group to which the information belongs:

DSIG	Data Set Identifier Group
DSSG	Security Group
DSPG	Parameter Group
DSMP	Map Projection Group
DSHG	History Group
DSVG	Variable Field Address Group

FEA: Each feature is defined by a variable length feature record stored in the sequential Feature File. The contents of a feature record is defined by the CFEAREC.FOR Include file and stored in the CFEAREC Common area. A byte buffer (C_frec) is defined to contain a feature record. Fixed information is defined as the first "m" fields in C_frec. A parameter points to the offset following this fixed information. The next "n" bytes of information in C_frec represent segment entries for the feature. The buffer C_fsegent is defined to contain one of these feature segment feature entries. This is followed by a variable number of 40 byte feature header blocks. The array C_fhdr is defined to contain one 40 byte header block, and the items for SLF to SIF data are equivalenced into this area.

HDR: The data set header record is defined in the CHDRREC.FOR Include file. It contains counts for total number of features, segments, points, etc. The C_hbadfea array records feature IDs of 'bad' features.

IFE: Each feature is defined by a variable length feature record stored in the Indexed Feature File. The contents of an indexed feature record is defined by the CIFEREC.FOR Include file and stored in the CIFEREC Common area. A byte buffer (C_ifrec) is defined to contain a feature record. Fixed information is defined as the first "m" fields in C_ifrec. A parameter points to the offset following this fixed information. This is followed by a variable number of 40 byte feature header blocks. The array C_ifhdr is defined to contain one 40 byte header block, and the

items for SLF to SIF data are equivalenced into this area. The next "n" bytes of information in C_ifrec represent segment entries for the feature. The buffer C_ifsegent is defined to contain one of these feature segment feature entries.

SEG: Each segment is defined by a variable length segment record stored in the RMS indexed Segment File. The contents of a primary or continuation segment record is defined by the CSEGREC.FOR Include file and stored in the CSEGREC Common area. The first four (4) bytes of a segment record are the unique primary key, consisting of the unsigned segment ID and the segment continuation number. A primary segment record is continuation number zero (0). A byte buffer (C_srec) is defined to contain either a primary or continuation segment record. For either type of record, fixed information is retained for both primary and continuation segment records and is defined as the first "m" fields in C_srec. A parameter points to the offset following this fixed information. In a primary segment record, number of points refers to total number of points contained in the segment. The next "n" bytes of information in C_srec represent feature entries for the segment. The buffer C_sfeaent is defined to contain one of these segment feature entries. This is followed by a variable number of coordinate pairs (or triplets, if Z data is present). The buffer C_spt is defined to contain an individual coordinate point. Parameters define the start offset of segment feature entries in C_srec and the maximum size of C_srec. For a segment continuation record, number of points refers to actual count of coordinates stored in the record. Segment points immediately follow the fixed segment information. Segment coordinates are stored as offset absolutes relative to the data set origin. These points are in the unit of measure, unit of resolution and scale of the data set. All points are X,Y[,X] for Cartesian data sets and Longitude, Latitude [,Elevation] for geographic data sets.

SCL: Each symbology is defined by a fixed length symbol correlation record stored in the RMS indexed Symbol Correlation Library. The contents of a symbol correlation record is defined by the SYMLIBREC.FOR Include file and stored in the SYMLIBREC Common area. A Symbol Correlation record is stored in SCL_buf; SCL_FeaID is the first 10 characters and serves as a unique primary key value.

B.4 PROGRAM MAINTENANCE PROCEDURES

B.4.1 Conventions

The directory scheme for the conversion software is illustrated by Figure B-4-1. The login command procedure is customized for the installation to define the rooted logical name SLI_SYS to point to the SLF2SIF top directory. Logical names are defined to point to each of the commonly used subdirectories (e.g., SLI_SRC contains the source modules). The modules for each major program are combined into a single source file to simplify maintenance procedures. Each source file has a corresponding command procedure which uses the logical names (e.g., SLI_SRC, SLI_OBJ, SLI_EXE) to compile and link the program. The command procedure ALL invokes each of the other command procedures to compile and link all of the conversion software in the correct sequence.

The conversion software was developed to conform to a set of design and coding standards including modularity and the use of control constructs. Include files are used to define data structures. Data item naming conventions are listed in Section 3.2.2, above. Parameter statements are used to define constants.

The conversion software makes use of the VMS Message facility for informational and error messages, which are defined as VMS condition values. This enables standardization of error reporting, via the CMSG service, so that both conversion and system condition values may be treated the same. Functions return a condition value to the calling routine.

To enhance program readibility, comments are included to reflect the design and code indentation to reflects the hierarchy and control flow. A standard format is used to provide documentation at the start of each source module. This documentation is combination comments and code and includes the declaration and description of all variables and routines called.

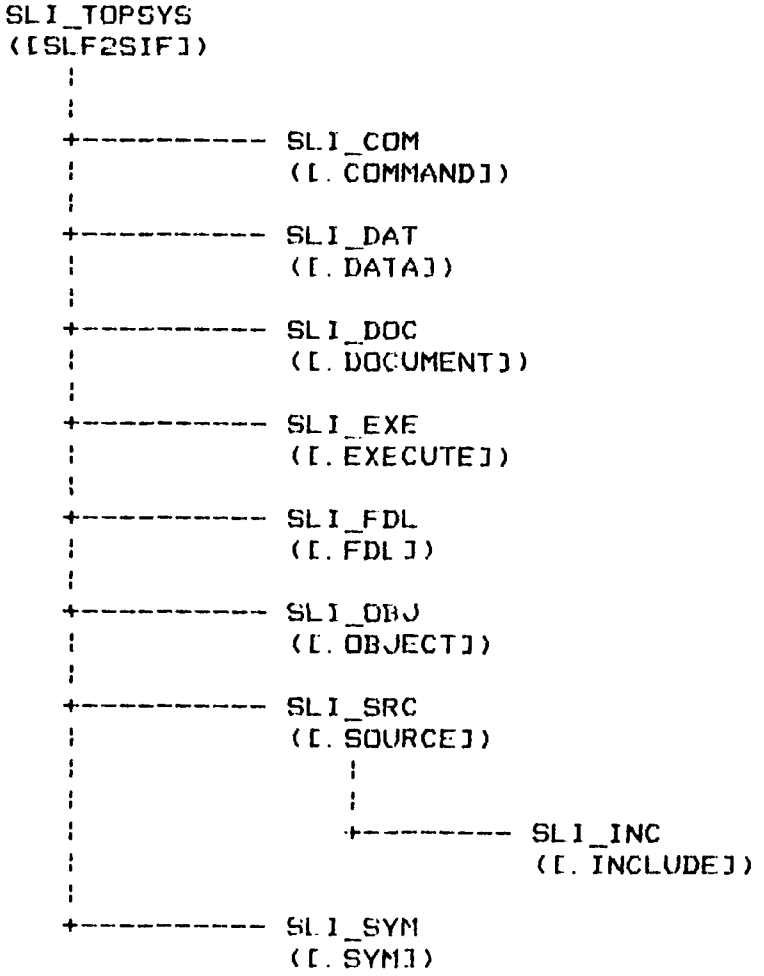


Figure B-4-1 Directory Organization

B.4.2 Verification Procedures

The ultimate test for the conversion software is input of the SIF data to the Intergraph system and conversion to IGDS/DMRS format. On the Intergraph system, the ASO processor may be used to obtain an ASCII formatted listing of the SIF file, DMRS may be used to review the attribute data base and IGDS Graphics allows review of the graphic information. Each of the SIF processors outputs error messages to a file type .MSG (e.g., TRI.MSG). Additional descriptions of error messages may be obtained by entering SIFERR and the appropriate error message.

Software development utilities are available to analyze data when the Intergraph system is unavailable. The SEEFEA and SEESEG routines provide a somewhat formatted dump of an internal feature and segment record, respectively. These dump report field contents by the Include file data item names. The SIFDUMP routine provides the capability to dump a SIF file in a format which isolates commands. The SIFDRAW utility in intended for use with the Printronix printer/plotter and uses the Digital PLXY optional software product.

B.4.3 Error Conditions

A complete list of informational and error messages generated by the Conversion software is documented under separate cover in reference 18, SLF to SIF Conversion Software Messages.

B.4.4 Special Maintenance Procedures

B.4.4.1 Software Delivery

It is recommended that the VMS Backup Utility be used to port the software between installations. This allows for the entire directory scheme (i.e., [SLF2SIF...]*.*) to be saved to tape and restored from tape. The login command procedure (LOGIN.COM) should reside in the default login directory and must be customized for each installation. Customization includes changing the definition of the rooted logical name SLI_SYS to point to the correct default device and definition of the logical name MTDEV for the type of tape device on the target system.

The command procedure SLI_COM:ALL.COM will compile and link all of the conversion software. Since the logical name LNK$LIBRARY is defined for the SLF to SIF Object Library, it is searched by default at link time. Any new programs should be added to the ALL.COM file.

B.4.4.2 Error Message Revision

The conversion software makes use of the VMS Message facility to define informational and error messages as VMS condition values. The facility prefix SLI_ is used for all messages. Messages are defined in the source file SLI_SRC:CMSGDEF.MSG, are defined as FORTRAN-accessible symbols in SLI_INC:CMSGSYM.FOR and are documented in SLI_DOC:MSGDOC.RNO. Each of these should be updated when a new message is added. The command procedure SLI_COM:CMSGDEF.COM will compile the revised message file. Additional information on the Message Utility is available from the VMS documentation.

B.4.4.3 Documentation

Documentation for the SLF to SIF conversion software is maintained on-line in Runoff input format. The source (.RNO) files are in the SLI_DOC directory; each has an associated command procedure to generate formatted output. The Program Maintenance Manual (PROGMAN.RNO), Users Manual (USERS.RNO) and Message Documentation (MSGDOC.RNO) should be revised if any changes are made.

B.4.4.4 Cell File Updates

The Cell File on the Intergraph system must contain all Cell and Pattern names referenced by the output SIF. These names will be derived from the Symbol Correlation Library. Standard Intergraph conventions are used to revise the Cell File. If text is specified for a cell (i.e., as an Enter Data field), it must be the first text line placed in the cell. Pattern cells are definied as point cells.

B.4.4.5 Pattern Definition File Updates

The Pattern Definition File on the Intergraph system must contain all patterns, by number, as referenced by the output SIF. Patterns are referenced by graphic group number; if any pattern is to be revised or a pattern is to be added, the entire Pattern Definition File must be rebuilt from scratch to ensure that graphic group numbers in the Pattern Definition File correlate to those in the Symbol Correlation Library. The following conventions are used to build the Pattern Definition File:

1. Create a design file containing a number of rectangles equal to the number of patterns to define. The file GENPATTRN.DGN may be used as a template.

2. The Cell Library must be attached to the design file. The default Cell Library (SYMLIB.CEL) will be automatically attached. This may be changed by executing the CF=cellib.CEL keyin.

3. Pattern Delay Mode must be on so that patterns (Type 5 data, reference 8) are not immediately applied to the Pattern Definition File.

4. Linear Patterns are definied first, starting with the lower-leftmost rectangle, followed by Area Patterns.

5. Each pattern to be created is identified by its name in the Cell library. A pattern cell is made 'active' by invoking the AP=pattern keyin. All line (weight, etc.) and pattern (scale, angle, delta) characteristics must be established for each pattern prior to its definition. Characteristics used in constructing the delivered Pattern Definition File are listed in Figure B-4-2.

6. Since Automatic Patterning is inhibited by Pattern Delay Mode, care must be taken when pushing the Data cursor button to define a pattern. One too many pushes will cause multiple definition of patterns per element, necessitating redefinition of the entire file.

7. When pattern definition is complete, the results may be viewed under IGDS by running the SIF Patterning task (PDV) against a copy of the Pattern Definition File. If the results are satisfactory, the following (and final) step may be executed.

8. Bring up the unpatterned Pattern Definition File under IGDS. Delete all rectangles (using Delete Fence Contents is easiest) and invoke the Compress File option, then exit the file. All that should remain are the undisplayable Type 5 pattern data elements. If any other data types exist in this file, the PDV task will not work.

GG Num	Description	Cell Name	Line Wt	Pattern Delta	Pattern Angle
1	Empty Casing	ECASE	3		
2	Empty Dashed Casing	EDCASE	1		
3	Full Tick	TICK	1		
4	Full Double Tick	DTICK	1		
5	Dashed Tick	DATICK	4		
6	Power Pylon Dashed	DASHPY	4		
7	Dashed X	DASHX	0		
8	Internat'l Boundary	CONDAS	6		
9	Admin Boundary	DITDAS	4		
10	Half Tick	HTICK	2		
11	Horizontal Hachure	HACH	0	0:100,0	75
12	Horizontal Hachure	HACH	2	0:100,0	135
13	Swamp Grass	SWAMP	0	0:250,0:250	0
14	Rice	RICE	0	0:100,0:100	0
15	Horizontal Hachure	HACH	2	0:100,0	0
16	Coniferous Tree	CTREE	0	0:100,0:100	0
17	Mixed Tree	CDTREE	0	0:250,0:250	0
18	Rocks Awash	ROCKAW	0	0:250,0:250	0

All patterns were created with the default pattern delta (0:0) and angle (0). GG Num indicates the graphic group number.

Figure B-4-2 Pattern Definition File Characteristics

B.4.5 Special Maintenance Programs

B.4.5.1 SIF Draw

SIF Draw (SIFDRAW) is intended for use with a Printronix printer/plotter and uses the Digital PLXY optional software product. An input SIF file is processed sequentially to produce an output plot file. PLOT, SYMBOL and PLOTCHAR are used to plot vectors, points and characters, respectively.

B.4.5.2 SIF Dump

SIF Dump (SIFDUMP) reads an input SIF file sequentially to produce a formatted dump.

B.4.5.3 See Feature

See Feature (SEEFEA) provides the capability to report selected features from an internal format feature file. The file is read sequentially to locate the desired features, which are reported using include file variable names.

B.4.5.4 See Segment

See Segment (SEESEG) provides the capability to report selected segments from an internal format segment file. The file is read by segment ID to locate the desired segments, which are reported using include file variable names. Segment points may optionally be reported.

B.4.6 Listings

Program listings are provided under separate cover.

APPENDIX C

MESSAGE DOCUMENTATION

This document describes the messages reported by the SLF to SIF Conversion software. It is a companion document to the Users Manual and the Program Maintenance Manual. Messages are listed in alphabetical order. Information defined for each message includes:

FACILITY:
SLI SLF to SIF Conversion

SEVERITY: Severity levels include:
S - Success
I - Informational
W - Warning (causes TraceBack)
E - Error (causes TraceBack)
F - Fatal (causes TraceBack AND aborts program)

MESSAGE_TEXT: This is a one-line description of the message. The Message Text specifies whether or not the message has Append Text and a Timestamp and where they are positioned.

EXPLANATION: This provides a description of the message, indicating the software component which reports the message and the reason for the message.

USER ACTION: This provides a description of the possible cause of errors and the actions which should be taken to resolve problems.

BADFEAFIX, Error inputting Fixed portion of feature <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Input data set contains an error in the Fixed portion of the specified feature, and the feature was bypassed.

User Action: The input data set should be corrected, if possible.

BADFEAHDR, Error inputting Header data of feature <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Input data set contains an error in the Feature Header portion of the specified feature, and the feature was bypassed.

User Action: The input data set should be corrected, if possible.

BADFEASEG, Error inputting FeaSeg entries of feature <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Input data set contains an error in the Feature Header portion of the specified feature, and the feature was bypassed.

User Action: The input data set should be corrected, if possible.

DEFVALSUB, Default Value Substituted <supplied value>

Facility: SLI, SLF to SIF Conversion

Explanation: A default value was substituted for an invalid input encountered in the Symbol Correlation File.

User Action: Correct the value in the Symbol Correlation File.

DMRASVMIS, DMRS Associative Values missing

Facility: SLI, SLF to SIF Conversion

Explanation: The conversion software detected an input DMRS Identifier command which was not followed by a DMRS Associative values command.

User Action: Correct the input SIF file.

DSFND, Data Set Found on input tape

Facility: SLI, SLF to SIF Conversion

Explanation: This is a successful return.

User Action: None.

DSIBADACC, DSI Accuracy Outline record has no points

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF DSI Accuracy Outline record contains no points.

User Action: The input data set should be corrected, if possible.

DSINOACC, Data Set has no Accuracy Outline records

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF data set contained no Accuracy outline records.

User Action: None. This is an informational message.

DSINOREG, Data Set has no Registration point records

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF data set contained no Registration Point records.

User Action: None. This is an informational message.

DSNOTFND, Data Set Not Found on input tape <dataset name>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified data set was not found on the input SLF data set.

User Action: Check that the data set name was supplied correctly and that the correct tape was mounted.

END, Process Completion at <time>

Facility: SLI, SLF to SIF Conversion

Explanation: The program recorded its completion date and time.

User Action: None. This is an informational message.

ENDOFFILE, End of File during read

Facility: SLI, SLF to SIF Conversion

Explanation: An end of file was encountered when reading from a file.

User Action: This is an informational message returned to the calling software and should not be encountered by the user.

ERRALLOC, Error During SYS$ALLOC

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned by the $ALLOC system service.

User Action: Refer to the previous error message. This may indicate that no device (e.g., tape drive) is currently available. This may also result from a software problem.

ERRASSIGN, Error During SYS$ASSIGN

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned by the $ASSIGN system service.

User Action: This indicates a software problem.

ERRMOUNT, Error During SYS$MOUNT

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned by the $MOUNT system service.

User Action: This indicates a software problem.

ERRCLOSEF, Error closing file <unit>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered closing the file on the specified unit.

User Action: This indicates a software problem.

ERRCREPOP, Error creating populated file <filespec>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered creating the specified file using spawn to perform a COPY or CONVERT.

User Action: This may indicate a software error. It may also result if you have exceeded your disk quota or need a larger bytlm or fillm in the user authorization file.

ERRFDLCRE, Error creating file using FDL file <FDLspec>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered using FDL$CREATE to create an empty file.

User Action: This indicates a software problem; possibly with the DM service calling sequence.

ERRFRELUN, Error releasing logical unit number <unit>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered during a LIB$FREE_LUN to release a unit number to the operating system.

User Action: This indicates a software problem.

ERRGETLUN, Error obtaining logical unit number

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned when attempting to obtain a logical unit number via LIB$GET_LUN.

User Action: This indicates a software problem.

ERRGETMSG, Error During SYS$GETMSG, --Date and Time--

Facility: SLI, SLF to SIF Conversion

Explanation: A VMS error status was returned from a call to the $GETMSG system service.

User Action: This indicates a software problem.

ERRGETSYM, Error from LIB$GET_SYMBOL

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned from the LIB$GET_SYMBOL routine.

User Action: This indicates a software problem.

ERRINQUIR, Error INQUIRING about file

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned when attempting to INQUIRE about a file.

User Action: This indicates a software problem.

ERROPENF, Error opening file <filespec>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned when attempting to open the specified file.

User Action: This indicates a software problem.

ERRPUTMSG, Error During SYS$PUTMSG, --Date and Time--

Facility: SLI, SLF to SIF Conversion

Explanation: A VMS error status was returned from a call to the $PUTMSG system service.

User Action: This indicates a software problem.

ERRSNDOPR, Error During SYS$SNDOPR

Facility: SLI, SLF to SIF Conversion

Explanation: A VMS error status was returned from a call to the $SNDOPR system service.

User Action: This indicates a software problem.

ERRSNSERR, Error from FORTRAN ERRSNS routine, code xxx

Facility: SLI, SLF to SIF Conversion

Explanation: The CMSG routine encountered an error while attempting to use SYS$GETMSG to translate the text for the most recent FORTRAN error as detected by ERRSNS.

User Action: This indicates a software problem.

ERRSTRTR', Error on STR$TRIM <string>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was returned by the STR$TRIM Run Time Library routine when attempting to trim the specified string.

User Action: This indicates a software problem.

FAOERR, Error performing $FAO

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered while using the $FAO system service.

User Action: This indicates a software problem.

FEABADSEG, Feature references non-existent segment <feature ID, segment ID>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified feature references the specified segment which was not found in the data set.

User Action: The input data set should be corrected, if possible.

FEANOHDR, Feature has no header blocks, feature <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF Feature record contains no feature header blocks.

User Action: The input data set should be corrected, if possible.

FEANOSEG, Feature has no segments, feature <ID>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified feature does not reference any segments.

User Action: The input data set should be corrected, if possible.

FEAREADER, Feature read error, feature <number>

Facility: SLI, SLF to SIF Conversion

Explanation: An error occurred reading the specified feature.

User Action: This indicates a software problem.

FEAUPDERR, Feature update error, feature <number>

Facility: SLI, SLF to SIF Conversion

Explanation: An error occurred revising the specified feature.

User Action: This indicates a software problem.

INVFEATYP, Invalid Feature Type <specified type>

Facility: SLI, SLF to SIF Conversion

Explanation: An invalid feature type was encountered.

User Action: Correct the input file, if possible.

INVFIDC, Invalid FIDC <value>

Facility: SLI, SLF to SIF Conversion

Explanation: An invalid FIDC with the indicated value was encountered in the input Symbol Correlation File. No symbology was output to the Symbol Correlation Library.

User Action: Correct the input Symbol Correlation File.

INVFILSPE, Invalid file specification <filespec>

Facility: SLI, SLF to SIF Conversion

Explanation: The indicated invalid file specification was provided to a DM Service routine.

User Action: This indicates a software problem.

INVSLFSEN, Invalid SLF record sentinel <value>

Facility: SLI, SLF to SIF Conversion

Explanation: A SLF record sentinel with the specified value was encountered in the SLF input data set.

User Action: The input SLF data set is incorrect. Verify that the system which generated the data set is functioning properly.

KEYREADER, Error performing keyed read for <value>

Facility: SLI, SLF to SIF Conversion

Explanation: An error occurred attempting to perform a keyed read for the specified key value.

User Action: This generally indicates a software problem, and may indicate an attempt to access a non-existent segment record.

LINPTFEA, Multi-segment Point feature number <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Specified point feature contains more than one segment entry.

User Action: The input data set should be corrected, if possible.

MAXBADFEA, Maximum number of bad features exceeded for <dsname>

Facility: SLI, SLF to SIF Conversion

Explanation: The input data set contained more than the maximum allowable number of bad features.

User Action: The input data set should be corrected, if possible.

MISSFLD, Missing field for record

Facility: SLI, SLF to SIF Conversion

Explanation: The specified input string has one or more missing fields.

User Action: Correct the input Symbol Correlation File.

MTREADER, Error on tape read

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered reading the tape.

User Action: Refer to additional messages. The tape may be bad or the incorrrect tape may have been mounted.

MTSKIPERR, Error on skip tape file

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered when skipping past a file (i.e., data set) on the tape.

User Action: Refer to additional messages. The tape may be bad or the incorrrect tape may have been mounted.

NORMAL, Normal Successful Completion

Facility: SLI, SLF to SIF Conversion

Explanation: This success message is associated with a status code returned from a SLF to SIF Conversion program.

User Action: None. This is a successful status code, which is not normally displayed by the software

NOTERM, No terminator present for record

Facility: SLI, SLF to SIF Conversion

Explanation: The specified input string contains no terminator character.

User Action: Correct the input Symbol Correlation File.

PNTARFEA, Single-point Areal feature number <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Specified areal feature contains one segment consisting of one coordinate.

User Action: The input data set should be corrected, if possible.

PNTLINFEA, Single-point Lineal feature number <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Specified lineal feature contains one segment consisting of one coordinate.

User Action: The input data set should be corrected, if possible.

ONEPNTELE, Single-point Element, Graphic Group number <GGNum>

Facility: SLI, SLF to SIF Conversion

Explanation: Specified line string element contains one coordinate.

User Action: The SLF to SIF Conversion software should be corrected such that single-point elements are output as cells.

OPENARFEA, Discontinuous Areal feature number <featureID>

Facility: SLI, SLF to SIF Conversion

Explanation: Specified areal feature contains one or more segments whose end points cannot be "walked" to form a closed feature.

User Action: The input data set should be corrected, if possible.

REGMISGRA, Registration Point missing graphic element

Facility: SLI, SLF to SIF Conversion

Explanation: The conversion software detected an input DMRS Identifier command and DMRS Associative values command for a registration point which were not followed by a graphic element.

User Action: Correct the input SIF file.

SEGBADFEA, Segment references non-existent feature <segment ID, feature ID>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified segment references the specified feature which was not found in the data set.

User Action: The input data set should be corrected, if possible.

SEGMISGRA, Segment missing graphic element

Facility: SLI, SLF to SIF Conversion

Explanation: The conversion software detected an input DMRS Identifier command and DMRS Associative values command for a segment which were not followed by a graphic element.

User Action: Correct the input SIF file.

SEGNOFEA, Segment has no features, segment <ID>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified segment does not reference any features.

User Action: The input data set should be corrected, if possible.

SEGNOPTS, Segment has no coordinates, segment <segmentID>

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF Segment record contains no coordinates.

User Action: The input data set should be corrected, if possible.

SEQREADER, Error performing sequential read

Facility: SLI, SLF to SIF Conversion

Explanation: An error occurred when performing a sequential read.

User Action: This indicates a software problem.

SLFRECSEQ, SLF record out of sequence <description>

Facility: SLI, SLF to SIF Conversion

Explanation: The input SLF data set contained records in a sequence other than that specified by the format rules.

User Action: Verify that the input data set is in the correct format.

SLFTAPBAD, SLF Tape Errors, <count> records processed

Facility: SLI, SLF to SIF Conversion

Explanation: Errors were encountered in processing of a SLF input tape. The indicated number of records were processed successfully.

User Action: This may result due to a bad tape, an improperly formatted tape or a software problem. Verify that the proper tape was mounted for the tape input process. If parity errors are being encountered, the tape drive heads may need to be cleaned or the tape may have been mounted at the wrong density.

SLFTAPPRC, SLF Tape Process Complete, <count> records

Facility: SLI, SLF to SIF Conversion

Explanation: Processing of a SLF input tape completed successfully. The indicated count of records were read from the tape.

User Action: None. This is an informational message.

START, Process Start at <time>

Facility: SLI, SLF to SIF Conversion

Explanation: The program recorded its start date and time.

User Action: None. This is an informational message.

STRCNVERR, String Conversion Error <string>

Facility: SLI, SLF to SIF Conversion

Explanation: An error was encountered attempting to convert the specified string to a byte, word or longword.

User Action: This indicates a software problem.

TAPEEOF, End of File Mark read from tape

Facility: SLI, SLF to SIF Conversion

Explanation: An End of File mark was encountered on the tape.

User Action: None. This is an informational message.

TAPEEOT, End of Tape encountered

Facility: SLI, SLF to SIF Conversion

Explanation: An End of Tape mark was encountered on the tape before processing was completed.

User Action: The tape may be in a bad format or the wrong tape may have been mounted.

TERM, Process Termination at <time>

Facility: SLI, SLF to SIF Conversion

Explanation: The process detected a non-recoverable error and terminated.

User Action: This abnormal process termination will result after some other type of error. Refer to the previous message.

TOOMANFLD, Too many fields in input <NbrFields>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified input string contains more than the legal number of fields.

User Action: Correct the input Symbol Correlation File.

TOOMANYCH, Too many characters in input <string>

Facility: SLI, SLF to SIF Conversion

Explanation: The specified input string contains more than the legal number of characters.

User Action: Correct the input Symbol Correlation File.

UNXEOF, Unexpected EOF encountered in SLF data set <dataset>

Facility: SLI, SLF to SIF Conversion

Explanation: An End-of-File condition was encountered in the specified data set before all expected data had been read.

User Action: Verify that the input data set is in the correct format.

UNXEOFSIF, Unexpected EOF encountered in SIF input file

Facility: SLI, SLF to SIF Conversion

Explanation: An End-of-File condition for the input SIF file was detected in the middle of a SIF command.

User Action: Correct the input SIF file.

WRITERR, Error performing write

Facility: SLI, SLF to SIF Conversion

Explanation: An error occurred when performing a write.

User Action: This indicates a software problem.

END

FILMED

12-85

DTIC