

NOT MEASUREMENT SENSITIVE

MIL-HDBK-857

17 June 2004

SUPERSEDING

(See Section 7.2)

DEPARTMENT OF DEFENSE HANDBOOK

Geospatial Symbols for
Digital Displays (GeoSym®)



This handbook is for guidance only.
Do not cite this document as a requirement.

FOREWORD

1. This handbook is approved for use by all Departments and Agencies of the Department of Defense.

2. This handbook is for guidance only. This handbook cannot be cited as a requirement. If it is, the contractor does not have to comply.

3. Comments, suggestions or questions on this document should be addressed to the National Geospatial-Intelligence Agency's National Center for Geospatial Intelligence Standards (NCGIS), Mail Stop P-24, 12310 Sunrise Valley Drive, Reston, VA 20191-3449, or emailed to ncgis-mail@nga.mil. Since contact information can change, you may want to verify the currency of this address information using the ASSIST Online database at www.dodssp.daps.mil.

CONTENTS

<u>PARAGRAPH</u>		<u>PAGE</u>
	<u>FORWARD</u>	ii
1.	<u>SCOPE</u>	
1.1	Scope	1
1.2	Purpose	1
1.3	Note	1
2.	<u>APPLICABLE DOCUMENTS</u>	
2.1	General	1
2.2	Government documents	1
2.2.1	Specifications, standards, and handbooks	1
2.2.2	Other Government documents, drawings, and publications	2
2.3	Non-Government publications	2
3.	<u>DEFINITIONS</u>	
3.1	Terms	3
3.2	Acronyms	3
4.	<u>ELEMENTS OF GEOSYM DEVELOPMENT</u>	
4.1	Design Criteria	5
4.1.1	User-Oriented Presentation	5
4.1.1.1	User-Defined Symbology	6
4.1.1.2	User of Query	6
4.1.2	Multiple-Product Displays	6
4.1.2.1	Product Scale and Generalization	7
4.1.2.2	Use of Color	7
4.1.3	Lighting Conditions	7
4.1.4	Dynamic View Scale	10
4.1.5	Visual Display Hierarchy	11
4.1.6	Color	12
4.1.7	Text	13
4.2	Methodology	13
4.2.1	Symbol Selection	14
4.2.1.1	Military Standards for Symbol Selection	15
4.2.1.2	Design of New, Alternative Symbols	15
4.2.2	Symbol Development	15
4.2.2.1	Point Symbology	15
4.2.2.2	Line Symbology	16
4.2.2.3	Area Symbology	16
5.	<u>GEOSYM USER ENVIRONMENT CONDITIONS</u>	
5.1	Operating System	17
5.2	Screen Resolution	17

5.3	Color Display	17
6.	<u>GEOSYM SOFTWARE IMPLEMENTATION GUIDANCE</u>	
6.1	Symbology Database Interface	17
6.2	Graphics File Format	17
6.2.1	CGM Versions	18
6.2.2	CGM Encoding	18
6.2.2.1	Binary Encoding	18
6.2.2.3	Clear Text Encoding	18
6.2.3	User-defined Application Attribute Structures in GeoSym	19
6.2.4	Appearance of Lines	29
6.3	Symbol Descriptions	29
6.4	Attribute Expressions	29
6.4.1	Attribute Expression Interpretation	30
6.4.1.1	ECDIS External Attribute Values	32
6.4.1.2	Feature in Multiple Coverages	33
6.4.1.3	Component Point Symbology	33
6.4.1.3.1	Buoys, Beacons, Lights, Light Vessel/Lightships	34
6.4.1.3.2	Lights on fixed and floating Aids to Navigation	34
6.4.1.3.3	Soundings	35
6.4.1.3.4	Low Accuracy Symbol	35
6.4.1.3.5	Point Symbol Orientation	35
6.4.1.4	Component Line and Area Symbology	36
6.4.1.5	Text as Label Attributes	36
6.4.1.6	Special Cases	37
6.4.1.6.1	Default Symbology	37
6.5	Display Hierarchy	37
6.5.1	Display Priority	37
6.5.2	Feature Delineation	38
6.5.3	Order of Rows in *sym.txt	38
6.5.3.1	Point Features with Components	38
6.5.3.2	Area Features with Multiple Fills	38
6.5.3.3	Features with Text Labels	38
6.5.4	Symbol Type	39
6.6	Color	39
6.7	Error Handling	40
6.8	Text Label Instructions	40

6.8.1	Text Label Join File (textjoin.txt)	42
6.8.2	Text Label Characteristics File (textchar.txt)	42
6.8.3	Text Label Location File (textloc.txt)	43
6.8.4	Text Label Abbreviation File (textabbr.txt)	43
6.8.5	Example Text Label Assignment File Cases	44
7.	<u>NOTES</u>	
7.1	Intended Use	49
7.2	Supersession data	49
7.3	Subject term (key word) listing	49
APPENDIX A - Additional References		50

1. SCOPE

1.1 Scope. This handbook provides information that is supplementary to the Detail Specification Geospatial Symbols for Digital Displays (GeoSym, MIL-DTL-89045, 17 June 2004).

1.2 Purpose. This document serves a three-fold purpose. Section 4 contains information that is historical in nature about the development of the GeoSym product. This information is provided for the reader as an explanation of the objectives of GeoSym. Section 5 contains information about the user environment within which GeoSym is designed to be utilized. Section 6 contains technical information critical to the design of application software that utilizes the GeoSym product.

1.3 Note. This handbook is for guidance only. Do not cite this document as a requirement. If it is, the contractor does not have to comply.

2. APPLICABLE DOCUMENTS

2.1 General. The documents listed below are not necessarily all of the documents referenced herein, but are the ones needed in order to understand the information provided by this handbook.

2.2 Government documents.

2.2.1 Specifications, standards, and handbooks. The following specifications, standards, and handbooks form a part of this document to the extent specified herein.

DEPARTMENT OF DEFENSE SPECIFICATIONS

MIL-DTL-89045	<i>Detail Specification Geospatial Symbols for Digital Displays (GeoSym), Edition 1, 17 June 2004</i>
MIL-PRF-89023	<i>Digital Nautical Chart, 19 Dec 1997 Amendment 1, 23 Feb 1999</i>
MIL-PRF-89037A	<i>Digital Topographic Data, 1 August 2002</i>
MIL-PRF-89049/1	<i>Foundation Feature Data, Draft, 30 Nov 1998</i>
MIL-PRF-89049/10	<i>Tactical Ocean Data Level 0, (Draft N01-0220)</i>
MIL-PRF-89049/11	<i>Tactical Ocean Data Level 1, (Draft N01-0220)</i>
MIL-PRF-89049/12	<i>Tactical Ocean Data Level 2, (Draft N01-0220)</i>
MIL-PRF-89049/14	<i>Tactical Ocean Data Level 4, 21 March 2002</i>
MIL-PRF-89035A	<i>Urban Vector Map, 1 August 2002</i>
MIL-PRF-89033	<i>Vector Map Level 1, 1 June 1995 Amendment 1, 27 May 1998</i>
MIL-PRF-89039	<i>Vector Map Level 0, 9 Feb 1995 Amendment 2, 27 June 2002</i>
MIL-PRF-89040A	<i>Vector Product Interim Terrain Data, 8 May 1996</i>
MIL-PRF-0089049 (NIMA)	<i>VPF Products, General Specification 24 Nov 1998</i>

MIL-V-89032	<i>Vector Smart Map Level 2, Draft, 30 Sep 1993 Amendment 1, 14 Jan 1994 Amendment 2, 16 Sep 1994 Amendment 3, 22 Dec 1994</i>
MIL-PRF-89012A (NIMA)	<i>World Vector Shoreline Plus, 24 Aug 1999</i>

DEPARTMENT OF DEFENSE STANDARDS

MIL-STD-2402	<i>Mapping, Charting and Geodesy Symbols for Graphic Products</i>
MIL-STD-2407	<i>Interface Standard for Vector Product Format Change Notice 1, 26 October 1999</i>

(Copies of these documents are available online at <http://assist.daps.dla.mil/quicksearch/> or www.dodssp.daps.mil, or from the Standardization Document Order Desk, 700 Robbins Avenue, Building 4D, Philadelphia, PA 19111-5094. Copies of draft documents can be obtained from the National Geospatial-Intelligence Agency's National Center for Geospatial Intelligence Standards (NCGIS), Mail Stop P-24, 12310 Sunrise Valley Drive, Reston, VA 20191-3449, or at http://www.nga.mil/cda/article/0,2311,3104_12141_115326,00.html, or request by email to ncgis-mail@nga.mil.)

2.2.2 Other Government documents, drawings, and publications. The following other Government documents, drawings, and publications form a part of this document to the extent specified herein.

DMA PRODUCT SPECIFICATIONS

PS/2BD/040	<i>DMA Product Specifications for Bottom Contour Charts (All Series), 4th Edition, March 1985</i>
PS/2DA/013	<i>DMA Product Specifications for Hull Integrity Test Site (HITS) Charts, 2nd Edition, December 1984</i>
PS/2DA/010	<i>DMA Product Specifications for Navy Operating Area (OPAREA) Charts, 1st Edition, December 1984</i>

(Copies of ex-DMA Product Specifications are available from the National Geospatial-Intelligence Agency's National Center for Geospatial Intelligence Standards (NCGIS), Mail Stop P-24, 12310 Sunrise Valley Drive, Reston, VA 20191-3449, or request by email to ncgis-mail@nga.mil.)

Symbology and Annotation for Maps and Imagery (SAMI) CGM profile of the ISO/IEC 8632-1 CGM Standard

(Copies of the SAMI Profile are available from ISO, and from NGA at http://www.nga.mil/cda/article/0,2311,3104_12141_115326,00.html.)

2.3 Non-Government publications. The following documents form a part of this document to the extent specified herein

ISO/IEC 8632-1:1999(E). Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 1:Functional Specification- Second edition, 1999-12-15.

ISO/IEC 8632-3:1999(E). Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 3:Binary encoding - Second edition, 1999-12-15.

ISO/IEC 8632-4:1999(E). Information technology - Computer graphics - Metafile for the storage and transfer of picture description information - Part 4:Clear text encoding- Second edition, 1999-12-01.

(Copies of these documents are available from the American National Standards Institute (ANSI) at <http://webstore.ansi.org>)

IHO S52. 1996. International Hydrographic Organization Specifications for Chart Content and Display Aspects of ECDIS. Special Publication No. 52. 5th Edition, December 1996.

IHO S52 Appendix 2. Colour & Symbol Specifications for ECDIS. Special Publication No. 52, APPENDIX 2, 4th Edition, July 1997.

IHO S52 Appendix 2, Annex A. IHO ECDIS Presentation Library Users' Manual. Special Publication No. 52, ANNEX A of APPENDIX 2, Edition/revision 3.1, January 1999. Note: GeoSym is compliant up to and including C&S Maintenance Document 1.

IHO S57. International Hydrographic Organization Transfer Standard for Digital Hydrographic Data. Edition 3.

(Copies of these documents are available from the International Hydrographic Organization (IHO) at <http://www.iho.shom.fr>)

SAE ARP5289. The Engineering Society For Advancing Mobility Land Sea Air and Space International. Aerospace Recommended Practice. Electronic Aeronautical Symbols. October 1997.

(Copies of this standard are available at http://www.sae.org/servlets/techtrack?PROD_TYP=STD)

3. DEFINITIONS

3.1 Terms Refer to MIL-DTL-89045 for a definition of the terms used in this document.

3.2 Acronyms

ASCII	American Standard Code for Information Interchange
CGM	Computer Graphics Metafile
CLD	Centerline Data
COL	Character of Light
COTS	Commercial-off-the-shelf
DNC®	Digital Nautical Chart

DOS	Disk Operating System
DPI	Dots per inch
DTOP	Digital Topographic Data
ECDIS	Electronic Chart Display and Information System
FACC	Feature Attribute Coding Catalogue
FLIP	Flight Information Publication
GIS	Geographic Information Systems
IHO	International Hydrographic Organization
INT	International Chart
MEDS	Mission Essential Data Set
NAM	Name
NGA	National Geospatial-Intelligence Agency (formerly NIMA)
RGB	Red Green Blue
RDBMS	Relational Database Management System
SAIC	Science Applications International Corporation
SAMI	Symbology and Annotation for Mapping and Imagery
SGD	Symbolized Graphic Data
SSC	Structure Shape Category
TLM	Topographic Line Map
TOD	Tactical Ocean Data
UVMap	Urban Vector Map
VITD	Vector Product Interim Terrain Data
VMap	Vector Map
VPF	Vector Product Format
WVSPPlus	World Vector Shoreline Plus
WYSIWYG	What-You-See-Is-What-You-Get

4. ELEMENTS OF GEOSYM DEVELOPMENT

4.1 Design Criteria. Objectives of the GeoSym development effort were:

- To develop a standardized, user-oriented presentation that facilitates map analysis
- To develop a cross-product digital symbolization standard that supports multiple-products
- To utilize the Computer Graphics Metafile ("CGM") format for storage and exchange of symbology graphics

The following National Geospatial-Intelligence Agency (NGA) VPF™ products are included within the scope of GeoSym .

Digital Nautical Chart (DNC®)
 Digital Topographic Data (DTOP)
 Foundation Feature Data (FFD™)
 Tactical Oceanic Data Level 0 (TOD™0)
 Tactical Oceanic Data Level 1 (TOD™1)
 Tactical Oceanic Data Level 2 (TOD™2)
 Tactical Oceanic Data Level 4 (TOD™4)
 Urban Vector Map (UVMaP™)
 Vector Product Interim Terrain Data (VITD™)
 Vector Map Level 0 (VMaP™0)
 Vector Map Level 1 (VMaP™1)
 Vector Smart Map Level 2 (VMaP™1)
 World Vector Shoreline Plus (WVSPPlus®)

These products have a broad range of scales and feature content. Therefore, the design for a symbology standard had to address the differences among (or utilize the similarities between) products. As a consequence, several aspects of digital symbology were considered:

- Dynamic display scale
- Variable lighting conditions
- Display hierarchy
- Use of color
- Use of text

The design objectives and the aspects of digital symbology, which needed to be considered, formulated the basis of the design criteria. These criteria are reviewed in the following subsections and resulted in the development of quality digital symbolization across all identified products.

4.1.1 User-Oriented Presentation Map design is more complex in the digital realm, as the traditional cartographic controls (map purpose, audience, scale) are increasingly more varied. The intended use of the map should nonetheless always be a fundamental determinant in the selection of the graphic elements and how the data are portrayed.

Digital VPF data provide more versatility than a static paper map product, as the map viewers can take part in the map design. Unlike a hardcopy product, which is often loaded with data in an effort to present as much information as possible for all users, the digital map viewer has the flexibility to customize the amount and type of data that are presented (and may even modify the symbology). This offers the possibility for the creation of application-

specific map displays (for example, a land navigation display, an air navigation display, a terrain analysis display) from the same (or multiple) feature data sets. Whereas a paper map must be all things to all readers (the tailoring for a specific application requires the reader to mentally filter out the undesired information), the digital map display can provide many different views based on the reader's application and interest.

4.1.1.1 User-Defined Symbology Standard symbology provides a standardized presentation of VPFdata that satisfies many general user applications and provides a reference implementation for developers. However, to maximize the potential for digital map analysis among a varied audience, the capability for creating customized displays with user-defined symbology should always remain an option.

User assignment and modification of symbology may be important for accomplishing various analytical tasks. However, it is important to emphasize that in a navigation application, user modification of symbology could present a hazard to a ship or aircraft and should be restricted. Users who wish to modify the standard symbology should be forewarned that modifications may impact others in a multiple-user viewing environment.

4.1.1.2 Use Of Query One advantage of digital products over hardcopy map products is the varied amounts of feature/attribute information and textual descriptions that can be stored within the data itself. Users of digital data may query the database to obtain complete attribute information on selected features. This capability was particularly important when developing the digital symbology for the following reasons:

- To reduce display clutter by minimizing the amount of text displayed
- To reduce the number of unique symbols for features that have extensive attribution (such as buildings) a generic symbol can be used. Then a user can query for the specific attribute information pertaining to that feature.

4.1.2 Multiple-Product Displays Standardized digital symbology should provide for a consistent symbolization presentation supporting the capability to view multiple products simultaneously. Adherence to a functional, multiple-product digital symbology standard minimizes unique, product-specific symbolization applications and limits inconsistencies that hinder the user's ability to recognize the map elements and analyze the data in the most efficient manner. Whenever necessary, product-specific exceptions took precedence over the overall objectives of a cross-product symbolization standard.

Users of NGA lithographic map products are familiar with the standards that are employed specifically for a particular map product and, initially, may prefer a digital map display, which simulates the look of the more-familiar lithographic products. On traditional NGA printed products, specific symbol assignments are made to features, taking into account the product type and purpose, specific attribution, and product scale. For example, a nautical chart is significantly different from a Topographic Line Map (TLM), even when the feature content overlaps.

Historically, there has been no real effort to ensure consistency across product lines, and the symbology in many cases is rooted in tradition and the

special needs of the users. Developing a cross-product symbology standard is a challenge both from a design standpoint and from a user acceptance standpoint. Consequently, users may have to reconsider some traditional symbolization techniques.

4.1.2.1 Product Scale and Generalization. Variations in the level of generalization across the identified VPF products are due to the variations in scale. The delineation of features appropriate for a particular level of generalization varies. Many examples of this are shown in Appendix A of MIL-DTL-89045. For example the built-up area feature (FACC AL020) is represented as either a point or area on the DNC product and as an area on the VMap Level 2, UVMaP, and VITD products. It is likely that in the event that multiple-scale products are displayed simultaneously, the appearance of duplicate features of the same delineation type and like features of different delineation types may present a confusing display. In this situation, the user needs the capability to customize the display view to include only a select set of features from each product (or, the display software may be designed to do this automatically based on a set of display conditions).

4.1.2.2 Use of Color NGA printed products have distinctive, product-specific color palettes. Nautical products generally are characterized by the emphasis on open water areas and use white instead of blue to portray safe deep-water areas. They also employ bright magenta and purple to draw the user's attention to important features, such as anchorages, buoys, lights, routes, maritime limits, and restricted areas. In contrast with NGA topographic products, the colors assigned to land features (such as roads, rivers, and vegetation) are subdued (and generally all black) so as not to be visually prominent.

Another example of product-specific symbology is the use of red to number major commercial and industrial buildings on the NGA City Graphic product. The red complex areas and red text used for the building identification numbering system should be the most visually prominent element of the display. Black is only used on the City Graphic product for text, and the roads and buildings are tan so they are more subdued in the visual hierarchy of the display.

A standardized color table that is applicable across the suite of identified VPF products is necessary in order to provide an intelligible and consistent display when viewing multiple product types simultaneously. The use of a standard cross-product color palette, rather than a product-specific color scheme, requires that some users adapt to a digital display, which differs from the paper map products to which they are accustomed. Nonetheless, the primary choice of colors for the cross-product color palette was based on those colors employed on NGA hardcopy charts. When combined with the application of solid color area fills, and the use of a "paper" white background color this contributes significantly to the aesthetic, "map-like" quality of the digital display. However, the use of digital cartographic products in different environments and conditions requires that symbology be able to maintain its usefulness in various lighting and environmental conditions.

4.1.3 Lighting Conditions As the use of VPF data becomes more prevalent, the VPF symbology must accommodate different types of user environments. Viewing conditions do vary in the type and quality of ambient light. Also affecting viewing conditions is the amount of viewing distance

between the user and the screen. For the GeoSym development effort, it was necessary to comply with the viewing specification prescribed in the Electronic Chart Display and Information System (ECDIS) described in the International Hydrographic Organization (IHO) Color and Symbol Specifications for ECDIS (S52, Appendix 2). User environments identified in this publication include:

- Bright Sun (DAY_BRIGHT)
- Day - White Background (DAY_WHITEBACK)
- Day - Black Background (DAY - Black Background)
- Dusk
- Night - (For use with Filter)

The ECDIS display requires that symbology be versatile across display devices in various specified ambient lighting conditions. Because of this requirement, IHO ECDIS symbols make use of a token system within the symbology used. Tokening allows for the modification of the color characteristics of a particular feature based on lighting condition. Due to the different interpretations of RGB values over particular displays, IHO defines the color tokens in the *Commission Internationale de l'Éclairage* (CIE) system of colorimetry. The application of the color token allows a great deal of flexibility for those symbols defined. The need for such flexibility is a requirement of ECDIS that mandates that symbols be made available for optimal viewing in multiple lighting conditions. Without having to maintain a different symbol set for each lighting condition, the examples provided from the IHO document supply several token-based color tables. A token acts as a slot for assigning defined CIE colors for a given ambient light condition.

Table 1 shows the color tokens and corresponding CIE values for the DAY_BRIGHT color table described in IHO S-52, Appendix 2, Annex A, Edition 3.1. For GeoSym symbol creation, the RGB values corresponding to the CIE values from Table 1 were used as default colors for symbol creation and viewing on the screen (see MIL-DTL-89045, Appendix C).

TABLE 1 - Color Tokens Established Within IHO: S-52

TOKEN	COLOR	X	Y	LUMINANCE
NODTA	GREY	0.28	0.31	45.00
CURSR	ORANGE	0.52	0.39	28.00
CHBLK	BLACK	0.28	0.31	0.00
CHGRD	GREY	0.28	0.31	25.00
CHGRF	GREY	0.28	0.31	45.00
CHRED	RED	0.48	0.30	25.00
CHGRN	GREEN	0.30	0.52	60.00
CHYLW	YELLOW	0.41	0.47	70.00
CHMGD	MAGENTA	0.30	0.17	20.00
CHMGF	MAGENTA	0.28	0.24	48.00
CHBRN	BROWN	0.42	0.45	30.00
CHWHT	WHITE	0.28	0.31	80.00
SCLBR	ORANGE	0.52	0.39	28.00
CHCOR	ORANGE	0.52	0.39	28.00
LITRD	RED	0.48	0.30	25.00

TABLE 1 - Color Tokens Established Within IHO: S-52

TOKEN	COLOR	X	Y	LUMINANCE
LITGN	GREEN	0.30	0.52	60.00
LITYW	YELLOW	0.41	0.47	70.00
ISDNG	MAGENTA	0.30	0.17	20.00
DNGHL	RED	0.48	0.30	25.00
TRFCD	MAGENTA	0.30	0.17	20.00
TRFCF	MAGENTA	0.28	0.24	48.00
LANDA	BROWN	0.36	0.40	49.00
LANDF	BROWN	0.45	0.45	15.00
CSTLN	GREY	0.28	0.31	10.00
SNDG1	GREY	0.28	0.31	25.00
SNDG2	BLACK	0.28	0.31	0.00
DEPSC	GREY	0.28	0.31	10.00
DEPCN	GREY	0.28	0.31	25.00
DEPDW	WHITE	0.28	0.31	80.00
DEPMD	PALE_BLUE	0.27	0.30	65.00
DEPMS	LIGHT_BLUE	0.24	0.26	55.00
DEPVS	MED_BLUE	0.22	0.24	45.00
DEPIT	YELLOW_GRN	0.28	0.36	40.00
RADHI	GREEN	0.30	0.52	60.00
RADLO	GREEN	0.30	0.52	20.00
ARPAT	BLUE-GREEN	0.26	0.42	30.00
NINFO	ORANGE	0.52	0.39	28.00
RESBL	BLUE	0.18	0.15	22.00
ADINF	YELLOW	0.41	0.47	35.00
RESGR	GREY	0.28	0.31	25.00
SHIPS	BLACK	0.28	0.31	0.00
PSTRK	BLACK	0.28	0.31	0.00
SYTRK	GREY	0.28	0.31	25.00
PLRTE	RED	0.58	0.35	18.00
APLRT	ORANGE	0.52	0.39	28.00
UINFD	BLACK	0.28	0.31	0.00
UINFF	GREY	0.28	0.31	25.00
UIBCK	WHITE	0.28	0.31	80.00
UIAFD	MED_BLUE	0.22	0.24	45.00
UINFR	RED	0.48	0.30	25.00
UINFG	GREEN	0.30	0.52	60.00
UINFO	ORANGE	0.52	0.39	28.00
UINFB	BLUE	0.18	0.15	22.00
UINFM	MAGENTA	0.30	0.17	20.00
UIBDR	GREY	0.28	0.31	25.00
UIAFF	BROWN	0.36	0.40	49.00
OUTLW	BLACK	0.28	0.31	0.00
OUTLL	BROWN	0.36	0.40	49.00

TABLE 1 - Color Tokens Established Within IHO: S-52

TOKEN	COLOR	X	Y	LUMINANCE
RES01	GREY	0.28	0.31	45.00
RES02	GREY	0.28	0.31	45.00
RES03	GREY	0.28	0.31	45.00
RES04	GREY	0.28	0.31	45.00
RES05	GREY	0.28	0.31	45.00

The requirement for the GeoSym development mandated that the symbology developed for all hydrographic products be ECDIS compliant. It was necessary to consider the best approach as to how to implement the color token concept for hydrographic products, and yet maintain the usability of symbology created for non-hydrographic VPF products. The use of indexed color, along with the definition of all color within the symbol graphic in RGB values for typical viewing displays has proven the best approach. Refer to section 6.6 for an explanation of how viewing software can access the correct RGB or color token for the display of GeoSym symbols.

4.1.4 Dynamic view scale The capability to "zoom in" and "zoom out" allows the user to enlarge or reduce the geographic extent that is displayed in the view window. Each time the area being viewed is enlarged or reduced, the display is redrawn. This has the same effect as changing the view display scale. Using the zoom command can be compared to using a magnifying glass on a paper map to focus on a smaller area, in that the area under magnification contains no more data than it did previously. Similarly, when zooming in, the enlarged magnified view that is redrawn does not change the product accuracy or extraction scale (e.g., no more data appears than that which was originally displayed at the smaller-scale view).

Users must be aware of the accuracy of the product data and be cautioned not to misinterpret the accuracy of large scale display views. Some examples which illustrate this issue further are provided in the following paragraphs.

The first example applies to a cartographic design or analysis scenario. It is a relatively common practice for map users to mentally translate the map scale into a common unit of measure. For example, at the scale of 1:1,000,000, 2 mm on the map represents a 2,000-meter ground distance. Map users experienced with a particular product learn to associate an approximate level of generalization and accuracy with that product (Robinson:46). On a 1:1,000,000 scale ONC product, the specifications require that features are positionally accurate to (+/-) 2,000 meters, which translates into a 2 mm map unit distance. If the size of a point symbol is 1 mm, a map user may assume that the symbol is horizontally accurate within a circle of 2 mm centered on the point symbol. However, if a user zooms in 10x (to a 1:100,000 display scale) and the point symbol size remains constant at a 1 mm size on the display, then at the new larger viewing scale, the point feature appears to represent only 100 meters of ground space, even though it is still only accurate to within (+/-) 2,000 meters from its center. When zooming into display scales that are much larger than the product scale, the map user can no longer assume that the horizontal accuracy of the point symbols falls within a reasonably close (i.e., 1 mm) proximity of the center of the symbol.

A second example concerns the use of map background displays for navigation. Suppose the mariner has the display scale of a navigation display set such

that the area being viewed is 10 nautical miles (nm) by 10 nm, offering a good overall presentation of shipping and navigation hazards in the immediate vicinity. Suppose the mariner then zooms in to a view of 1 nm by 1 nm and the point symbols (e.g., buoys and other navigation aids) remain the same size as they were in the 10 nm by 10 nm display. In the absence of other cues, this may present the illusion that the display is set to the same scale as previously, giving the mariner a false sense of security when there may actually be unseen hazards to his ship in his immediate vicinity. Alternatively, if the point symbols became proportionately larger as the mariner zoomed in, they would offer a cue that he was actually viewing a very small geographic area.

A third example concerns display symbolization for spatial data analyses that might be accomplished in a GIS. In an analytical environment, the user would typically desire point symbolization in order to facilitate feature identification. Zooming in on the display would discern greater detail about the geography displayed (e.g., details of coastlines, area features, etc.) and to eliminate some clutter on the display by limiting the quantity of features displayed.

The above examples illustrate that there might be a need for scalable symbols, fixed size symbols, or scale-driven symbol changes based on specific applications. For the last example, point symbols that scale up in size as the display is zoomed in would not contribute to the view of the geography and would actually cause greater clutter. In such an analytical situation, the user would probably desire that the point symbols remain a consistent size when zooming in. When zooming out, however, the user might actually desire that the symbols change to a different representation, such as small circles or dots, in order to reduce clutter as more features are added to the display. On the converse, for situations similar to the first two examples, the display needs to mimic what happens when a magnifying glass is used with a paper map; the symbol enlarges proportionately with the area under the magnifying glass. The 1mm symbol on a 1:1,000,000 scale product should always be 1,000 meters in ground units regardless of the zoom factor, (i.e. in order to cover 1,000 meters of ground distance at a 1:100,000 viewing scale, the symbol would need to be 10 mm).

Dynamic view displays have a dramatic effect on the visual clarity and effectiveness of all symbolization. In particular, point symbols (even simplified shapes) can become unreadable at small display view scales.

Of these display issues, the GeoSym product currently only addresses the "zoom out" scenario for hydrographic products, where the viewer wishes to switch to a different, simplified set of symbols. GeoSym includes both full chart symbol definitions (for all VPF products), as well as a set of simplified symbols (for DNC only). The user may toggle between the two symbol sets at his discretion. The other issues in regard to viewing scale are still under discussion at NGA and are not resolved by this version of GeoSym.

4.1.5 Visual display hierarchy. An important characteristic of visual perception, known as the figure/ground phenomenon, often is used by cartographers as a fundamental design tool for organizing a map display (Dent:127). Essentially, humans tend to automatically segregate elements in their visual field into figures (objects that stand out and have form) and grounds (amorphous backgrounds). Any map design, including a digital map display, will be more effective when the graphic elements such as contrast,

color, and overlay are used to create a visual hierarchical organization of the map components. Visual confusion results from an undifferentiated display. For example, the application of a solid color area fill for water bodies and land masses can greatly enhance the figure/ground development and improve the clarity of the map display. For this reason, the use of solid color area fill is recommended, and is considered a fundamental capability necessary for establishing a visual hierarchy, as well as adding a more graphic, map-like quality to the digital display.

The digital presentation of map information also requires that a sequential order be defined for the display of the features, not only to present a logical organization of the data but also to prevent destructive overwrite during the display update. For example, in the VPFView software, the "painter's algorithm" or display priority for VPF feature classes, follows in this order: areas, lines, points, and text. Within each class, the themes are drawn in the order in which they were selected during the creation of the views. Once a view is created, users need to be aware that modifications to the symbolization could adversely affect the display priority and potentially disrupt the view. For other applications, the use of a display priority value sets the display priority for a feature irrespective of the feature's delineation. GeoSym utilizes a combination of the "painter's algorithm", an assigned display priority (ranging from 0 to 9, with nine being the highest priority), and the ordering of the rows in the symbol assignment table to control the order that features are displayed on the screen. See section 6.5 for a detailed description of how these display control mechanisms are utilized in GeoSym.

4.1.6 Color. The color scheme used for the map display is effective when it organizes the data and facilitates user recognition of features. Color is a fundamental design element and, when used properly, is a primary element in feature identification on the digital display. Care must be given to the selection of the color palette and to the effects of color combinations that can occur when color assignments are made for areas, lines, points, and text. In general, lighter, "easy on the eye" colors should be used for background area feature displays and strong colors like magenta and black should be limited to small areas. The choice of colors similar to those employed on NGA hardcopy charts can contribute much to the aesthetic, "map-like" quality of the digital display.

In addition to general color selection, however, one must also consider the changes to color values over map regions with changes in ambient lighting conditions. A great deal of study has gone into the color designations for symbology in various lighting conditions. The creation of symbology based on the IHO color token definitions ensures that the color designations and dynamic harmony are maintained.

The use of a white background and solid color area fills allows the use of black for symbols and text. Black is a very effective color for text and some symbols since it is very distinguishable (when combined with any other color) and is not prone to color induction. This is desirable when the digital display can potentially include a variety of color combinations between features.

The GeoSym digital symbology uses many conventional cartographic and symbolic aspects of color to achieve a map-like display, such as green for vegetation, blue for water, brown for landforms, and red for importance. In addition, the

IHO (IHO:36) recommendations for general display colors were used as a starting point for the colors defined within created CGM symbology. Note that for hydrographic products, the color token indices will instead be used as the basis for color selection depending on the ambient lighting condition setting defined by the user.

4.1.7 Text Clutter can be defined as any map data that impede the communication objective of the map because the data are illegible or contain a level of detail that is inappropriate for the display. The problem of reducing clutter in a digital display is an issue that must be resolved to provide effective map displays.

The hardcopy symbology standard (MIL-STD-2402) is "text intensive" in that it employs an abundant use of text for symbol identification. Many features are represented with the same symbol. Consequently, the use of text to accompany the symbol is necessary for feature identification (although the proximity of other features/symbols may provide other clues in the data to assist in identification). For example, numerous features, such as wrecking yards, power plants, windmills, substations, grain bins, grain elevators, forts, and buildings, can be symbolized by a small, black, square-point symbol. There also are numerous applications for a thin, black, dashed-line symbol: fairgrounds, athletic fields, firing ranges, stockyards, mines, and scrap yards. In these instances, trying to replicate the hardcopy symbology, complete with text, for digital display viewing would not be successful. Computer display technology is still far from replicating the resolution of printed media. To be legible, text must be larger on the screen than on paper. Consequently, numerous features requiring labels yield a cluttered, ambiguous display.

Digital VPF products have an advantage over hardcopy products because varying amounts of feature/attribute information and textual descriptions are stored in the data dictionary. Previous studies (Dodd, 1991; SAIC, 1996:29; and SAIC, 1993:8) have concluded that more effective, less cluttered screen displays result from minimizing the use of text. Dodd recommends that users of digital VPF data rely on cursor inquiry and the use of queries to obtain complete attribute information on selected features. The Dodd recommendations regarding text were adopted for the GeoSym digital symbology.

However, in some cases, the use of cursor inquiry to display textual information is cumbersome, if not impractical, in fulfilling the requirements of the user. A prime example of this situation is seen in the treatment of sounding information. It is difficult to imagine the relative usefulness of a hydrographic chart where soundings are displayed as a generic symbol whose value must be obtained only after cursor inquiry -- for each sounding. In this case, a provision for the display of text information must be made. Refer to section 6.4.1.5 for an explanation of how to utilize the text labels defined in GeoSym.

4.2 Methodology. The development of the GeoSym symbology has been an on-going process. Key objectives were to focus on prior symbology efforts and address any important findings or recommendations for the general improvement of the GeoSym Symbology. The level of illustrative graphic detail in the symbol that is necessary for feature identification is subject to the technical limitations of the software and display capabilities. Whether symbols are replicated from existing standards or new symbols are designed,

empirical testing is necessary to evaluate the clarity and legibility of the symbols at various display view scales.

4.2.1 Symbol Selection. To facilitate symbol selection, the following priorities were established and utilized for each VPF product.

- *1st priority:* Product Mission - Symbol selections were made such that the mission of the product would be supported in the best possible fashion. For this reason, there are some instances where a certain feature may be symbolized differently from one product to the next for products with non-similar product missions.
- *2nd priority:* Consistency within each product - Symbol selections were made such that non-similar features did not inadvertently receive the same or similar symbology. The intent here was to provide contrast between non-similar features within the same product. Conversely, similar or identical symbology was used for features that were similar, where contrast was not preferable. For example, a mine feature and a quarry feature would receive the same or similar symbology whereas a built-up area feature would receive contrasting symbology.
- *3rd priority:* Consistency with other products - Symbol selections were made such that features would be symbolized in a consistent manner among all products. The intent here was to ease the transition from one product to another in cases where multiple products are used in conjunction with one another. An additional benefit of this is the reduction in the total number of symbols that must be created and maintained, and the number of symbols with which the users must become familiar.
- *4th priority:* Compliance with applicable government symbology standards document(s) - Several Military Symbology Standards were utilized to select symbols that met the first three priorities (see section 4.2.1.1). Whenever possible, symbols were selected that were compliant with the applicable government standard for that feature. Non-compliant symbology selections were made only in cases where compliant symbology compromised any or all of the previous priorities.

The symbol selection process was performed separately for each product. The first step taken in this process was an evaluation of each feature and its associated attributes for the product being reviewed. This evaluation was based on the appropriate VPF product specification. Based on the mission of the product, a determination was made as to which attributes, if any, would drive the symbology for a given feature. For example, a mine feature may not have attributes driving the symbology, whereas building features can be differentiated by the building function category (BFC) attribute. The appropriate military symbology standard document was reviewed to help determine which attributes, if any, should drive symbolization. If attribution was driving the symbol for a particular feature, attribute values were then logically grouped, if necessary, and associated with a particular symbol. If there was no attribution driving the symbology, then an appropriate symbol was selected for the feature.

Once the symbol selection for all features in each of the products was complete, the symbols selected for each product were compared on a feature-by-feature basis to same or similar features in other products. Adjustments were

made whenever possible to provide consistency in symbol selection across products.

4.2.1.1 Military Standards for Symbol Selection Because the need for DNC symbology was critical, DNC was the first product reviewed for the GeoSym development effort. This became the DNC mini-prototype effort. The symbology prototype was tasked with the development of symbology that was compliant with ECDIS as defined in the IHO S-52. The symbology and guidance presented within this publication were used as the guideline for the definition and selection of symbols as well as feature attribute expressions for hydrographic features for GeoSym. For the DNC product, feature attribute expressions that were derived from IHO S-57 and correlated with DNC drove the symbol selection process. For subsequent hydrographic products, such as LWD*, TOD and WVSPPlus, the DNC mini-prototype, the IHO Presentation Library, and the appropriate Navy hardcopy charts (OPAREAS, HITS, BC) were used as guidelines for symbol creation and selection.

Symbol selection for topographic/terrain or land based features was based on Military Standard MIL-STD-2402. The topographic products reviewed included UVMMap, VMap0, VMap1, and VMap2. The terrain products reviewed included VITD, DTOP-MEDS1*, DTOP-MEDS3*, and DTOP-MEDS5*. Although the MIL-STD-2402 was written for graphic products, it was very useful in the selection of symbols for NGA VPF products. For each VPF product, symbol selection was based on the symbology specified in MIL-STD-2402 for the graphic product that most closely corresponded to the VPF product being symbolized. This not only provided some consistency in the symbols being selected, but also utilized much of the symbology from the graphic maps with which users are currently familiar.

Symbol selection for aeronautical features in DFLIP* was based on United States Government Flight Information Publications: Enroute, Terminal and DOD Military Training Route Charts

*Although GeoSym prototype 3 and 4 were developed to support LWD, DTOP-MEDS, and DFLIP, since those products have not matured as of the writing of this document, the official release of GeoSym does not include the symbols and attribute expressions to support their display.

4.2.1.2 Design of New, Alternative Symbols The suite of VPF products encompasses varying resolutions, feature content, feature types, and feature attribution. Therefore, it was necessary to design new, alternative symbols to produce a comprehensive, consistent digital symbology set for all of the identified VPF products. Alternative, illustrative symbols were required to minimize the use of text for feature identification.

4.2.2 Symbol Development A description of how the point, line, area and text symbology was developed is provided below along with a discussion on the color model that was used.

4.2.2.1 Point Symbology ActiveCGM Author was used for the creation of all point symbols in the symbol creation effort. Symbols were created in color using a 16 bit format and written out in both ASCII clear text and binary encoding as version 4 CGMs. The binary encoding CGM format was used for the final production symbol set in order to minimize storage requirements and reduce the task of CGM interpretation only to this more widely accepted format.

Point symbols were drawn such that each attribute expression was associated with only one point symbol CGM file. However, some features derive symbol components from expressions containing more than one attribute. For example, buoys, lights, and beacons have several attributes that drive the symbology for that feature. Because of the extremely large number of symbols and complex expressions that would be required to represent each of the combinations, the attributes themselves are symbolized separately. That is, for a buoy, the base is symbolized based on the value of the SSC and BTC attributes. The symbol is called from one attribute expression that contains only the SSC or BTC attribute value, which matches the feature being symbolized. Other aspects of the buoy, if present, (such as sound, topmark, etc.) are symbolized in a similar fashion based on the relevant attribute. This greatly reduces the number of symbols and expressions that must be created and maintained and has minimal impact on display speed.

4.2.2.2 Line Symbology. Line symbols are used to represent both linear features and the boundaries of area features. Line symbols were drawn using one of two methods in ActiveCGM Author. Solid lines were drawn using a predefined solid line type within the ActiveCGM Author software. Dash lines along with more complicated dashed-dotted lines and lines, which incorporate point symbols in their definition, were drawn using the add-on VPF line style functionality within Author developed exclusively for GeoSym. This additional functionality provides a method for creating a variety of complex line styles such as dashed lines made up of varying dash and gap lengths, dashed lines with embedded symbols, dual and triple-case solid lines, and solid lines with beginning and ending line terminators. Complex lines can contain as many embedded point symbols as necessary. The resulting line CGM has embedded in it the name of each point symbol that is required to define the line. For this reason, each complex line symbol CGM that utilizes one or more external point symbols must be accompanied by the CGM(s), which contain(s) the point symbol(s) referenced. In GeoSym, all point symbols that are referenced within a complex line CGM are numbered 5008 and greater.

4.2.2.3 Area Symbology An area feature can be represented using any combination of area fill, boundary, and/or point symbology. The CGM symbols used to represent area fill for area features were drawn using two methodologies depending on whether the area was a solid color fill or a pattern fill. Solid fill areas were drawn using a pentagon as the basic shape with the appropriate solid color fill. Edge visibility was turned off for these symbols so that only the solid color fill is visible. Area pattern fills are also represented using a pentagon shape within the CGM. Each pattern fill was defined by creating a 32 by 32 bitmap cell using a bitmap editor. The single pattern definition is stored in the CGM as a pattern table, which is "tiled" to fill the pentagon drawn in the CGM.

Area boundary symbology utilizes line symbology as described in section 4.2.2.2. When used for area boundaries, the line symbols should be rendered in a clockwise direction around the area. This will ensure the correct orientation of the line style components relative to the area feature.

Some area feature symbology also utilizes a point to be placed at the center of an area feature. For these cases, a point symbol is used. Section 4.2.2.1 describes the creation of point symbols.

5. USER ENVIRONMENT FOR GEOSYM EXPLOITATION

5.1 Operating System. In order to accurately represent the area patterns required by both S52 and MIL-STD-2402, the pattern elements in the GeoSym area symbols are drawn using a 32x32 pattern element. This requires the user to utilize the Windows NT or Windows 2000 operating systems. Windows 9x will only support an 8x8 pattern.

5.2 Screen Resolution. The recommended screen resolution for the optimum display of GeoSym symbols is 1280x1024, with a minimum resolution of 1024x768. The symbols will become distorted below this minimum.

5.3 Color Display. Those users who wish to view the GeoSym symbols in their intended colors should refer to S52, Appendix 2 (Colour & Symbol Specifications for ECDIS), Annex C for instructions on how to calibrate their monitors.

6. SOFTWARE DESIGN AND IMPLEMENTATION FOR GEOSYM EXPLOITATION

6.1 Symbology database interface The following sections supplement the information contained in the Detail Specification Geospatial Symbols for Digital Displays, MIL-DTL-89045. They contain key engineering issues that a developer should consider during the design and implementation of software written to exploit GeoSym. It is assumed that the developer is familiar with the MIL-DTL-89045, IHO S52, IHO S57, and SAMI documentation.

6.2 Graphics file format The GeoSym symbols are in CGM format. CGM, or *computer graphics metafile* is a rich, open standard format that can encapsulate many kinds of graphic data including vector, raster, or a mixture of both.

Essentially, the CGM is composed of a series of instructions that describe the geometric and presentation characteristics of graphics. The file does not contain pictorial data, but rather is made up of a set of defined instructions used for describing one or more graphic elements. The graphics are only displayable through the use of a translator capable of reconstructing the information described in the CGM instructions.

The main objective of the CGM file is to facilitate the storage and retrieval of graphical information between applications, software systems, and/or devices. The format and content of these instructions are defined by the International Standards Organization, in ISO/IEC 8632-1. The CGM is the only ISO-standard vector graphics format. Several other formats have been derived from the 8632-1 that have been developed to support the specific needs of end users.

The CGM standard was originally published in 1986 by ANSI as X3.122. The same standard was published a year later by ISO as ISO 8632:1987. To make it easier to manage changes, ANSI X3.122:1986 was replaced by ANSI/ISO 8632:1987 in 1991. There were 3 amendments to CGM:1987, two of which are incorporated in the new, republished CGM:1992. The other amendment and CGM:1987 were cancelled. The latest published CGM standard is ISO/IEC 8632-1:1999, 8632-2:1999, 8632-3:1999.

According to ISO/IEC 8632-1, a CGM may contain the following.

- vector graphics (e.g., polylines, ellipses, NURBS)
- raster graphics (e.g., tile array)
- text

6.2.1 CGM Versions The CGM standard defines four upwardly-compatible versions, each providing additional functionality. *Version 1* provides a basic drawing and picture interchange capability. The *Version 1* metafile definition includes approximately 90 individual function or entity elements. *Version 1* metafiles are essentially the same as those described in the 'old standard', CGM: 1987. The *Version 2* metafile definition contains approximately 30 more elements in addition to all of those described in *version 1* metafiles. The most significant new capability of *Version 2* however, is the *graphical segment*. *Version 3* metafiles represent a major increase in graphical expressive power. *Version 3* metafiles contain approximately 40 added elements to the *Version 2* capabilities. *Version 3* metafile functionality includes: the capability to represent compressed tiled images, define external symbol libraries, and provide greater control of drawing aspects for graphics arts, presentation graphics, and electronic publishing.

The latest version of metafiles is the *Version 4* metafile. *Version 4* metafiles include *Versions 1, 2, and 3* and add hierarchical structures and user-defined attributes so user communities can define and create "intelligent graphics." *Version 4* metafiles also define directories so applications can randomly address large metafiles containing many pictures and structures.

GeoSym symbols are all *Version 4* CGMs due to the need for user-defined attribute structures (see section 6.2.3). This does not mean, however, that all possible elements in *versions 1 through 4* are represented in the GeoSym symbols. The subset of CGM elements that are utilized in GeoSym is defined in the Symbols and Annotation for Maps and Imagery (SAMI) profile. Alternatively, the developer may reference any clear text encoded CGM and examine the "MFELEMLIST" element for a complete list of those CGM elements that are contained in the GeoSym CGMs.

6.2.2 CGM Encoding The CGM file is defined in ISO/IEC 8632-1 as having two encoding capabilities. These encodings are binary and clear text encoding. The choice of the encoding used in implementation must be based on the following considerations.

- interpretative speed
- readability (human) of the encoded files

6.2.2.1 Binary Encoding The focus of binary encoding, which is described in ISO/IEC 8632-3, is the speed of file creation and interpretation. The use of binary data format is efficient due to its similarity to the representations of data structures within computer systems.

6.2.2.2 Clear Text Encoding The purpose of the clear text encoding, which is described in ISO/IEC 8632-4, is to provide a representation of the metafile syntax in a human-readable form. Not only can the syntax be interpreted by the end user, but the file may also be edited or even created from any standard text editing device. The key objectives of the clear text format are listed below. Note that the speed of interpretation (machine) is a secondary consideration for this encoding.

- Human editable
- Human friendly (readable)
- Machine readable - the syntax is able to be parsed by software
- Suitable for text editors
- Interchangeable between systems and platforms
- Standardized codes and abbreviations

The GeoSym CGM files are compliant with the Symbology and Annotation for Maps and Imagery (SAMI) profile. This profile was developed to support the creation and use of CGM files for the GeoSym effort. The profile describes the minimum implementation of the CGM functions in order to allow interpreters to be written to support the CGM files, as well as provide the basis from which additional symbols may be created to augment the existing CGM symbol library. Note that though the clear text encoding is included with GeoSym, it is not intended as a source from which interpreters would read symbology definition. Only the binary encoding is supported as described in the SAMI profile.

6.2.3 User-defined Application Attribute Structures in GeoSym. As stated previously, all GeoSym symbols are version 4 due to the existence of user-defined application attribute structures. The elements of each one of these structures consist of the following: BEGAPS, APSATTR, BEGAPSBODY, ENDAPS. Each type of application attribute structure has a unique identifying name that is contained within the BEGAPS line. All GeoSym CGMs contain the following named application attribute structures: IC_ViewportTable, PictureProperties, and IC_ColorNameTable. Both IC_ViewportTable and PictureProperties should be ignored by GeoSym application software. These two structures are used solely by the ActiveCGM Author tool and are retained for consistency between the cleartext and binary versions of the GeoSym CGMs. Refer to section 6.6 for an explanation of how GeoSym application software can utilize the IC_ColorNameTable structure to access color tokens.

There is a fourth application attribute structure that is contained only in the complex line CGMs of GeoSym. The name of this structure is LineStyle and it consists of a series of nested application attribute structures that define the characteristics of each specific complex line. The complex CGMs also contain the graphic primitives that depict a sample of the rendered representation of the complex line style. However, those elements are only present to support the generation of MIL-DTL-89045 and should not be used for the actual display of VPF features. GeoSym application software should always use the LineStyle application attribute structure definition to render the complex line style along the linear features or area boundaries to which the complex line symbol applies. At a high level, the LineStyle application attribute structure has the following construction:

```
LineStyle
  LineStyleComponent
    LineStyleComponentElement
    LineStyleComponentElement
    .
    .
```


LineStyleComponent

.

.

.

Table 2 lists the parameters at each level of the LineStyle structure and their allowable values. Software developers should refer to the SAMI for a complete definition of the LineStyle and the IC_ColorNameTable application attribute structures for the information that is relevant to the GeoSym exploitation software.

Table 2 - Complex Line Style Parameters

Parameter Name	Description	Type	Units
Line style name	Unique name of the line style	String	NA
One to many line components	A line style may contain one to many line components. For each line component the parameters are defined below		
Line width	Thickness of the line	Real	CGM
Line color	Color of the line	Int	Index Number
Start anchor of element	Placement of the first element relative to the line in the x-axis. Any one of the following placements can be used. "Start", "Middle", or "End". <i>Note: When projecting the line style, this is in effect for the first iteration of the line style only.</i>	Enumerated 0,1,2	NA
Iteration Type	Determines if the component should be iterated continuously or a single time only.	Enumerated 0,1	NA
Start phase displacement of element	The offset distance of the first element from the start anchor	Real	CGM
One to many Elements	A line component may contain one to many elements. For each element the parameters are defined below.		
Element type	Type of element. Any one of the following element types can be used. "Gap", "Dash", or "Point Symbol".	Enumerated 0,1,2	NA

Table 2 - Complex Line Style Parameters

Parameter Name	Description	Type	Units
Element length	The length of the gap or dash. When the length of a dash is specified to be zero, then a solid line will be rendered to span the projected line in its entirety. <i>Note: Not applicable for an element type of "Point Symbol"</i>	Real	CGM
Vertical displacement	The distance in the y-axis between the center of the element and the projected path of the line.	Real	CGM
Symbol definition	The name of the IHO point component symbol <i>Note: Applicable only to element type of "Point Symbol"</i>	String	NA
Symbol scale size	The scale factor of the IHO point component symbol. <i>Note: Applicable only to element type of "Point Symbol"</i>	Real	None
Symbol orientation type	The IHO point component symbol orientation type. One of the following orientation types can be used: "Oriented at a constant angle to x-axis", or "Oriented tangentially to the line". <i>Note: GeoSym will only utilize "Oriented tangentially to the line", which is designated as numeral 1.</i> <i>Note: Applicable only to element type of "Point Symbol"</i>	Enumerated 0,1	NA
Symbol initial angle	The IHO point symbol rotation relative to the x-axis. <i>Note: Applicable only to element type of "Point Symbol" and symbol orientation of "Oriented at a constant angle to x-axis"</i>	Real	Degrees

Following are some examples of how to interpret the LineStyle application attribute structure, highlighting in particular the use of the start anchor and its purpose in the structure. The Start Anchor attribute appears at the component level of the application attribute structure for complex lines and determines the starting location of the first element within the component.

The possible values of the attribute are 0 for beginning, 1 for middle, and 2 for end. A Start Anchor of 0 (beginning) is the default for all complex line components and indicates that the element is to begin at the beginning of the line feature being symbolized.

Example 1: Symbol 0613.cgm is a thin dash line, LS(DASH,2,DEPSC), which utilizes a beginning start anchor. This symbol is shown in Figure 1.



Figure 1. Example of Beginning Start Anchor - Symbol 0613

The following is an excerpt from the cleartext CGM 0613.cgm that shows the application attribute structure for symbol 0613.

```

BEGAPS "0613.1" "LineStyle" STLIST;
BEGAPSBODY;
BEGAPS "0613.1.Component.1" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.64";
APSATTR "LineColor" " 11 1 41";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0613.1.Component.1.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 3.6";
APSATTR "VerticalDisplacement" " 12 1 0";
BEGAPSBODY;
ENDAPS;
BEGAPS "0613.1.Component.1.Element.2" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 0";
APSATTR "ElementLength" " 12 1 1.8";
BEGAPSBODY;
ENDAPS;
ENDAPS;
ENDAPS;

```

This symbol is made up of one component, which consists of two elements: a dash and a gap. Notice the StartAnchor attribute is listed at the component level and has a value of 5 1 0 indicating that the first element, the dash, will start at the beginning of the line. Notice also that the IterationType attribute value of 0 indicates that the elements will be repeated in a continuous fashion.

Example 2: A Start Anchor of 1 (middle) is used if the symbol requires that an element begin in the middle of the line. Symbol 0752 is a very thin dashed line with centered waves LC=LS(DASH,1,CHGRD)+ SY(WATTUR02). This symbol utilizes a middle start anchor and is shown in Figure 2.



Figure 2. Example of Middle Start Anchor - Symbol 0752

The following is an excerpt from the clear text CGM 0752.cgm that shows the application attribute structure for symbol 0752.

```

BEGAPS "0752.1" "LineStyle" STLIST;
BEGAPSBODY;
BEGAPS "0752.1.Component.1" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.32";
APSATTR "LineColor" " 11 1 20";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0752.1.Component.1.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 3.6";
APSATTR "VerticalDisplacement" " 12 1 0";
BEGAPSBODY;
ENDAPS;
BEGAPS "0752.1.Component.1.Element.2" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 0";
APSATTR "ElementLength" " 12 1 1.8";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "0752.1.Component.2" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.32";
APSATTR "LineColor" " 11 1 20";
APSATTR "StartAnchor" " 5 1 1";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0752.1.Component.2.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 14.51044";
APSATTR "VerticalDisplacement" " 12 1 -1.2";
APSATTR "SymbolDefinition" ' 14 1 "5107.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
ENDAPS;

```

This symbol is made up of two components. The first component represents the dashed line and consists of two elements, a dash and a gap. Notice the start

anchor for this component is 0 (beginning) indicating that the dash element will begin at the beginning of the line. The IterationType attribute value of 0 indicates that the elements will be repeated in a continuous fashion.

The second component represents the wave and consists of only one element, the wave point symbol (5107.cgm). Notice the start anchor for this component is 1 or middle indicating that the point symbol element will begin at the middle of the line. The IterationType attribute value of 1 indicates that the element will only be a single iteration.

Example 3: Figure 3 illustrates another symbol that utilizes the middle Start Anchor value. The clear text application attribute structure associated with this symbol is also given below.

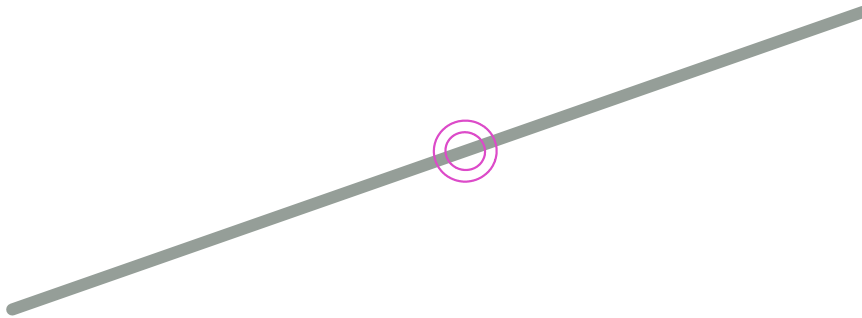


Figure 3. Example of Middle Start Anchor - Symbol 0753

```

BEGAPS "0753.1" "LineStyle" STLIST;
BEGAPSBODY;
BEGAPS "0753.1.Component.1" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 1.6";
APSATTR "LineColor" " 11 1 20";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0753.1.Component.1.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 0";
APSATTR "VerticalDisplacement" " 12 1 0";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "0753.1.Component.2" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 1.6";
APSATTR "LineColor" " 11 1 20";
APSATTR "StartAnchor" " 5 1 1";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0753.1.Component.2.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 8.297122";
APSATTR "VerticalDisplacement" " 12 1 0";
APSATTR "SymbolDefinition" ' 14 1 "5106.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
ENDAPS;

```

Example 4: A Start anchor of 2 (end) is used if the symbol requires that an element "begin" at the end of the linear feature being symbolized. Symbol 0751.cgm is a black solid line with tick terminators. This symbol utilizes both a beginning and ending start anchor and is shown in Figure 4.

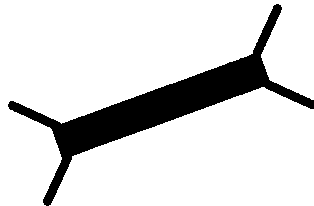


Figure 4. Example of End Start Anchor - Symbol 0751

The following is an excerpt from the clear text CGM 0751.cgm that shows the application attribute structure for symbol 0751.

```

BEGAPS "0751.1" "LineStyle" STLIST;
BEGAPSBODY;
BEGAPS "0751.1.Component.1" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 2";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 1.6";
BEGAPSBODY;
BEGAPS "0751.1.Component.1.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 0";
APSATTR "VerticalDisplacement" " 12 1 0";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "0751.1.Component.2" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 2";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "0751.1.Component.2.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 2.571082";
APSATTR "VerticalDisplacement" " 12 1 0";
APSATTR "SymbolDefinition" ' 14 1 "5103.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "0751.1.Component.3" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 2";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 2";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 1.8";
BEGAPSBODY;
BEGAPS "0751.1.Component.3.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 1";
APSATTR "VerticalDisplacement" " 12 1 0";
APSATTR "SymbolDefinition" ' 14 1 "5104.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
ENDAPS;

```

Symbol 0751 is composed of three components. The first component represents the thick black line and is made up of one element, a solid line indicated by the dash element type in combination with a 0 element length. The start anchor for this component is 0 or beginning and the iteration type is 0 or continuous indicating that the element will begin at the beginning of the line and will be repeated in a continuous fashion. The start phase of 1.6 indicates that the solid black line should start 1.6 units to the right of the true start of the feature being symbolized. This small gap will accommodate the left ticks. The second component represents the left ticks, point symbol 5103.cgm, which also have a start anchor of 0 or beginning, however, the iteration type is single for this component.

The third component represents the right ticks and consists of only one element, the right ticks, point symbol 5104.cgm. Notice the start anchor for this component is 2 or end indicating that the point symbol element will begin at the end of the line. The iteration type attribute value of 1 indicates the element will be a single iteration. The start phase of 1.8 allows the origin of the point symbol to be shifted right so that the left side of the symbol aligns flush with the end of the solid black line.

Example 5: Figure 5 illustrates another symbol that utilizes the end start anchor value. The clear text application attribute structure associated with this symbol is also given below.



Figure 5. Example of End Start Anchor - Symbol 1572

```

BEGAPS "1572.1" "LineStyle" STLIST;
BEGAPSBODY;
BEGAPS "1572.1.Component.1" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.64";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "1572.1.Component.1.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 2";
APSATTR "VerticalDisplacement" " 12 1 0.5";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "1572.1.Component.2" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.64";
APSATTR "LineColor" " 11 1 17";

```



```

APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 0";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "1572.1.Component.2.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 1";
APSATTR "ElementLength" " 12 1 2";
APSATTR "VerticalDisplacement" " 12 1 -0.5";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "1572.1.Component.3" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.64";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 0";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "1572.1.Component.3.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 2.640207";
APSATTR "VerticalDisplacement" " 12 1 0";
APSATTR "SymbolDefinition" ' 14 1 "5008.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
BEGAPS "1572.1.Component.4" "LineStyleComponent" STLIST;
APSATTR "LineWidth" " 12 1 0.64";
APSATTR "LineColor" " 11 1 17";
APSATTR "StartAnchor" " 5 1 2";
APSATTR "IterationType" " 5 1 1";
APSATTR "StartPhase" " 12 1 0";
BEGAPSBODY;
BEGAPS "1572.1.Component.4.Element.1" "LineComponentElement" STLIST;
APSATTR "ElementType" " 5 1 2";
APSATTR "ElementLength" " 12 1 2.640496";
APSATTR "VerticalDisplacement" " 12 1 0";
APSATTR "SymbolDefinition" ' 14 1 "5009.cgm"';
APSATTR "SymbolScale" " 12 1 1";
APSATTR "SymbolOrientation" " 5 1 1";
BEGAPSBODY;
ENDAPS;
ENDAPS;
ENDAPS;

```

This symbol consists of 4 components. The first component represents the top line, which is a straight line vertically offset from the feature centerline by 0.5mm. The second component represents the bottom line, which is a straight line vertically offset from the feature centerline by -0.5mm. Notice the iteration type on both of these components is 0 or continuous so that the line continues along the length of the linear feature, creating the cased line effect.

The third component represents the left ticks. The component consists of one element, left tick point symbol 5008.cgm, and is defined with start anchor 0 so that the element appears at the beginning of the linear feature. The fourth component represents the right ticks. The component consists of one element, right tick point symbol 5009.cgm, and is defined with start anchor 2 so that the element appears at the end of the linear feature. Notice the Iteration type on both of these components is 1 or single indicating that only a single iteration of the element is to be rendered.

6.2.4 Appearance of Lines. In addition to the basic line attributes of width and color, GeoSym CGMs contain the line attributes LINECAP, LINEJOIN, and LINETYPECONT.

LINECAP indicates the way that the end of a line is rendered when drawn and also the way that the ends of internal dashes in a dashed line are to be drawn. GeoSym linear CGMs contain a LINECAP value of "2 2" indicating that the line should be rendered with a butt (squared) ending and that internal dashes will also be squared off. GeoSym point CGMs contain a LINECAP value of "3 3" indicating that the line should be rendered with a rounded ending and that internal dashes will match that depiction, meaning that they will be rounded as well. Currently, there are no point symbols that utilize a dashed line element.

LINEJOIN indicates the way that line elements connect, both within an element and between two or more line elements. GeoSym linear CGMs contain a LINEJOIN value of "2" indicating that the lines are to be mitered (squared off) around the connecting endpoint. GeoSym point CGMs contain a LINEJOIN value of "3" indicating that the lines are to be rounded around the connecting endpoint.

LINETYPECONT indicates how a dashed (complex) line pattern symbol should be maintained across a vertex. Both the linear and point CGMs in GeoSym contain a LINETYPECONT value of "2" indicating that the linear pattern should continue across the vertex. This CGM element is included in the point CGMs, although it is not utilized in point symbols.

6.3 Symbol Descriptions Symbol descriptions may be found in MIL-DTL-89045, Appendix C listed by symbol name. The format and structure of the complete set of symbology assignment files is defined in MIL-DTL-89045.

6.4 Attribute Expressions Attribute expressions are used to map a specific type of a feature to the appropriate symbol(s) for display. As described in MIL-DTL-89045, there are two symbol assignment files: fullsym.txt and simpsym.txt. Regardless of which file is applicable (based on user selection), each row in the file contains the symbolization instructions for a given delineation of a feature in a VPF product. Some features can be adequately symbolized based only on the FACC code. Features such as a Mine (AA010), or a Ramp (AL195) do not require unique symbols based on the type of mine or ramp.

Other features require unique symbols based on the value defined for specific attributes of that feature. For example, a Road (AP030) must be symbolized differently based on whether it is hard/loose, paved/unpaved, or with or without a median. All attribute-based expressions are contained in a separate file, the attexp.txt file. The link between a row from the appropriate symbol assignment file (*sym.txt) and the associated attribute expression in the

attexp.txt file is the row id from the *sym.txt file. The format of the attexp.txt file is defined in MIL-DTL-89045. Example content and how application software should utilize this content is described in the following paragraphs.

6.4.1 Attribute Expression Interpretation Each row of the attexp.txt file defines a separate and unique attribute expression component. There may be from one to 255 attribute expression components in a single attribute expression. The attribute expression components are grouped by the *cond_index* field. The value in the *cond_index* field will be the same as the value of the *id* field in the *sym.txt file to which the attribute expression applies.

The *seq* field in each attribute expression component controls the order in which the attribute expression is to be constructed. This value is a number from 1 to 255 that identifies each row of an attribute expression.

The *att*, *oper*, and *value* fields contain the actual conditional expression that is to be tested for true/false status. *Att* is a set of characters identifying an attribute name. In most cases, this will be a three character FACC attribute code, such as NAM, RST, or TMC, as defined on the feature in the VPF product structure. In certain cases, however, *att* will be a four-character acronym identifying an external attribute pertinent to ECDIS symbolization instructions (see section 6.4.1.1 for a definition of each of these external attributes and its possible values). *Oper* defines the relational operator to apply to the attribute code. This will be a number from 1 to 6 where each value is defined as follows:

1	=	(equals)
2	<>	(not equals)
3	<	(less than)
4	>	(greater than)
5	<=	(less than or equal to)
6	>=	(greater than or equal to)

The *value* field contains the value against which to test the attribute code for true. The *value* field may contain an integer, a real number, a text string enclosed in quotes (i.e., "UNK"), or the word NULL (no quotes). Note that the meaning of NULL is unique based on the data type of the attribute in the VPF product data being displayed. Application software must interpret NULL per Table 62 in the VPF Standard (MIL-STD-2407) in order to accurately evaluate the attribute expression component.

The last field in each row of the attexp.txt file is the *connector* field. The purpose of this field is to provide the means to connect the multiple attribute expression components into a single attribute expression. The connector field will contain a number from 0 to 4 where each value is defined as follows:

0	None (the expression consists of a single component)
1	or
2	AND
3	and
4	OR

The "or" connector is used to connect multiple attribute expression components that define multiple valid values for the same attribute. The *att* value must be the same for components that are connected using "or". For example,

```
bfc=0 or bfc=6 or bfc=10
```

The "AND" connector is used to connect multiple attribute expression components that define valid values for more than one attribute. The *att* value must be different for components that are connected using "AND". For example,

```
bfc=0 AND hgt>50
```

The "and" connector is used to connect multiple attribute expression components that define the valid range for the same attribute. The *att* value must be the same for components that are connected using "and". For example,

```
wid>0 and wid<40
```

The "OR" connector is used to connect multiple attribute expression components that are independent of each other. The *att* value must be different for components that are connected using "OR". For example,

```
bsc=2 OR bot=7
```

Following the final attribute expression component is a 0 in the connector field. This indicates that the current attribute expression component is the last component in the expression and that there are no further components to the complete attribute expression.

Using these four different connectors, it is possible to construct complex attribute expressions that are easily interpretable by both the reader and application software. The connectors provide an inherent precedence to the attribute expressions evaluation, equivalent to the use of parentheses. The lowercase connectors provide the means to represent those portions of the attribute expression that are effectively "inside the parentheses", while the uppercase connectors are flags that there is either another condition component that must also be met before the entire expression is true OR there is a completely different attribute expression (itself potentially composed of multiple component expressions) that could result in a valid evaluation of "true" for the entire attribute expression.

The following list provides examples of attribute expressions first in English text (with parentheses where needed to clarify the order of evaluation) and then, as the expression would look as a tab-delimited row in the *attexp.txt* file. These are examples only and are not intended to reflect actual rows from the delivered *attexp.txt* file. Note the 0 in the connector position to indicate the final attribute expression component in each group.

Example:

1. tmc=2
25|1|tmc|1|2|0
2. nam<>"UNK"
200|1|nam|2|"UNK"|0

3. bfc=0 or bfc=7 or bfc=52
42|1|bfc|1|0|1
42|2|bfc|1|7|1
42|3|bfc|1|52|0
4. btc=0 AND ssc=7
500|1|btc|1|0|2
500|2|ssc|1|7|0
5. (bsc=2 or bsc=7) OR bot=11
222|1|bsc|1|2|1
222|2|bsc|1|7|4
222|3|bot|1|11|0
6. ((bfc=0 or bfc=2) AND (hgt>0 and hgt<46) AND exs<>2) OR hwt=7
350|1|bfc|1|0|1
350|2|bfc|1|2|2
350|3|hgt|4|0|3
350|4|hgt|3|46|2
350|5|exs|2|2|4
350|6|hwt|1|7|0

6.4.1.1 ECDIS External Attribute Values In the symbolization of certain hydrographic features on ECDIS systems, variation of symbol representation occurs based on attributes not found within the VPF data structure. To this end, external attributes were developed as four-character codes to distinguish them from the attributes available within VPF. The five external attributes implemented within the expressions are described below. It is the responsibility of the reading application to provide for some interface to allow the user to input or establish the values of these variables. Additionally, in the event a variable cannot be matched to existing data, the application is responsible for the assignment of the next most applicable value as the operative value of the variable.

ISDM: Interactive Shallow Display Mode - ECDIS defines the display mode of shallow water areas (shallow depth zones) to be one of two symbology scenarios. The attribute values are 1 and 0, which toggle the shallow display mode to be on or off respectively. When ISDM is set on (1), the display of all depth zones shallower than the defined values of the Ship's Safety Depth Contour (SSDC) are overprinted with a lattice pattern. This mode can be initiated in the four- or two-depth zone display modes (not including the drying line), defined by the Interactive Display Selection Mode, (IDSM). The shallow display mode is made available due to viewing limitations of the shallow depth zones in night displays.

IDSM: Interactive Display Selection Mode - ECDIS defines the display of depth zones to be divided into two or four depth areas. This variable allows for the mariner to interactively set either display mode. The two-zone mode uses only the ship's safety depth contour (SSDC) as a zone separator, whereas the four-zone mode further divides zones based on the mariner selected deep and shallow contours (MSDC, MSSC). Attribute values are 0 and 1 meaning four- and two-zone modes respectively.

SSDC: Ship's Safety Depth Contour - The ship's safety depth contour represents a safe contour based on the draft of the ship. This value must be entered by the mariner using an application interface. This interface must ensure that if a contour value does not exist within the data, that a next deeper value is specified as the SSDC. This checking must be dynamic as one traverses tile boundaries within the data.

MSDC: Mariner Specified Deep Contour - The four-zone display mode requires the establishment of a deep contour that must be specified by the mariner through application inquiry. A default value may be implemented at 30m according to the ISO Color and Symbol Specification directives.

MSSC: Mariner Specified Shallow Contour - The four-zone display mode requires the establishment of a shallow contour that must be specified by the mariner through application inquiry. A default value may be implemented at 2m according to the ISO Color and Symbol Specification directives.

6.4.1.2 Feature in Multiple Coverages In certain products, the same feature (fcode) is represented in multiple coverages with a different set of attributes. Three different methods are implemented in the symbol assignment table based on the manner in which the given VPF product defined the feature.

Unique Attributes:

In some products, when a feature exists in multiple coverages, a different set of attributes is defined in each coverage. Therefore, in the symbol assignment table, although there may be multiple records for a specific fcode, there will only be one record that is found to be true for a particular instance of that feature.

Unique Attribute Value:

In most of the VPF products, there is a Well feature defined (AA010). There is an attribute on the AA010 called PRO that defines what type of Well it is. The symbology for a water well is different than that symbology for an industry well. Therefore, the condition for a well is specified based on the unique value of the PRO attribute

Coverage Column (COV):

The DTOP product contains a group of features (referred to as the "common open water features") that are present on all four of the continuous coverages in that product. Only in the surface drainage coverage (sdr) do the features contain any attribution. In the other coverages, there are no attributes defined on the features. Since the features in the sdr coverage must be symbolized differently than in the other coverages, those feature's symbol assignment records contain a value in the COV column. This value will either be "sdr" or "<>sdr" to provide the application software the means by which to determine which symbology to apply.

6.4.1.3 Component Point Symbology In most cases, the symbol defined for a point feature is based on a small number of attributes that are able to be represented in a "reasonable" quantity of finite rules. However, the following sections describe cases where this method of symbol assignment is not practical.

6.4.1.3.1 Buoys, Beacons, Lights, Light Vessel/Lightships In the case of beacons (BC010), buoys (BC020), lights (BC040), and light vessels or lightships (BC070) it is not feasible to generate a single composite point symbol for each possible permutation of symbolic attributes since this would have forced the creation of thousands of attribute expressions and symbols for each feature. Instead, there are multiple rows for these features in the symbol assignment file (*sym.txt) with each row representing a single attribute's conditional expression. Application software must construct the appropriate composite symbol for the BC010, BC020, BC040, and BC070 features based on the final display of all specified components. It is important that the order in which the rows are accessed from the *sym.txt file be maintained in the display of the component point symbols in order to achieve the correct complete symbol (see section 6.5).

In addition, should a given component attribute's value result in no row being found in *sym.txt, application software should interpret this situation as "there is nothing to place for this symbology component" instead of the normal behavior for selecting the default unknown symbol (section 6.4.1.6.1). For example, there is no row in *sym.txt for a buoy with a Top Mark Category (tmc) attribute equal to 33 (None). The interpretation of this situation is that if the data being symbolized has a value of 33 in the tmc attribute, then there is simply no topmark component to be placed as a part of the total composite symbolization for this buoy.

6.4.1.3.2 Lights on fixed and floating Aids to Navigation. Due to the fact that the color of the light flare on certain fixed and floating aids to navigation is represented as an abbreviation embedded in a character text attribute (COL), the *sym.txt file does not contain the rules for the selection of the appropriately colored light flare. In order to select the appropriate colored light flare symbol, software should parse the text of the COL attribute, searching for the standard abbreviations for color defined in Chart 1. The Character of Light (COL) attribute is any identifier expressed as a 10-character text string (per MIL-STD-2407) and composed of the class, the number of lights, and the color(s) of flashes or occultations, of a light or lights associated with a single fixed or floating aid to navigation. For example: F1 (3) R. In this case, a light flashes three times per sequence and illuminates as red. The standard abbreviations associated with the colors are:

W (white)
 R (red)
 G (green)
 Bu (blue)
 Vi (violet)
 Y (yellow)
 Or (orange)
 Am (amber)

Note that a COL value of "N/A" means NULL, thereby signifying that the feature is unlighted (BC020 only). A value of "UNK" means the feature is lighted, but the characteristics are not known.

The correct CGM to place based on color is as follows:

white/yellow	0503.cgm
green	0502.cgm
red	0501.cgm

all others 0500.cgm

Once the correct CGM is selected, it should be the last symbol component to display for the resulting composite symbol for fixed and floating aids to navigation. Fixed and floating aids to navigation include the point delineations of the following features:

AL015 - BUILDING, Not Maritime Station (BFC<>81)
 AT080 - COMMUNICATION TOWER
 AL240 - TOWER (NON-COMMUNICATION)
 AM070 - TANK
 BC020 - BUOY
 BC040 - LIGHT
 BC070 - LIGHT VESSEL /LIGHTSHIP
 GB010 - AIRPORT LIGHTING

Because the above features do not have rules to populate the *sym.txt with the symbolization parameters for the appropriate light flares, the following information is provided for clarity:

Display Priority - 8
 Radar - ON TOP OF RADAR
 Display Category - STANDARD DISPLAY
 Viewing Group - 27070 (LIGHTS)

6.4.1.3.3 Soundings The sounding feature (BE020) presents a unique challenge for digital display since the value of the sounding is not stored as a discrete textual entity, but rather as an attribute (HDP or HDH) on the sounding feature in the VPF data. Additional symbology such as a circle (for soundings of low accuracy) or an overbar (for drying heights) may need to be displayed along with the sounding depth. Furthermore, both the sounding text and the overbar must be displayed in different colors based on the depth of the BE020 feature relative to the Ship's Safety Depth Contour (SSDC). Therefore, the symbolization of soundings must be accomplished through the display of the (possibly) multiple components for that feature. For the depth value itself, application software must retrieve the value of the HDP or HDH attribute (they are mutually exclusive) and convert it to a displayable text string. Furthermore, additional processing will be required if the user desires to see the historical display of soundings (with a base and a subscript).

6.4.1.3.4 Low Accuracy Symbol ECDIS display requirements for many features require the use of a low accuracy symbol, SY(LOWACC01). Certain of the S52 symbolization instructions in the lookup tables or in the symbolization procedures require that a low accuracy symbol be added to the "regular" symbolization for a given feature. Again, rather than create duplicate attribute expressions (one set with and one set without the accuracy attribute, acc) for these features, there will be an additional row in the *sym.txt file that controls the placement of the low accuracy symbol. For this reason, as well as for the placement of text labels (see section 6.4.1.5), application software must access all rows from the *sym.txt file for a given point feature code in order to ensure full symbolization.

6.4.1.3.5 Point Symbol Orientation For some of the IHO point symbols, it is important to know the orientation at which to place the point symbol. For those features to which this situation applies, there is an attribute

value specified in the *orient* field of the **sym.txt* file. If this field is populated, application software should place the specified point symbol at the angle of orientation that is contained within the attribute specified in the *orient* column. This field may be populated for both point and area features.

6.4.1.4 Component Line and Area Symbology It is also possible for there to exist multiple symbology components for linear and area features. The most common situation is the need for the addition of the low accuracy symbol (see 6.4.1.3.4). This situation is implemented in the same way as for points, with an additional row in the **sym.txt* table to control the placement of the low accuracy symbol.

It is also possible for the symbolization of an area feature to require multiple rows to specify the components of the full area symbol. This situation will exist for those area features requiring both a solid fill and a pattern fill. In this case, the two area symbols will be specified using two rows in the **sym.txt* file with the row specifying the solid fill always preceding the row specifying the pattern fill. If the same area feature also requires a boundary and/or a centered point symbol, those symbols will be specified in the second row for the area feature (along with the area pattern). Section 6.5 explains the ramifications of this approach in more detail.

For these reasons, as well as for the placement of text labels (see section 6.4.1.5), it is crucial that application software access all rows from the **sym.txt* file for a given product/delineation/feature code in order to ensure full symbolization for any feature.

6.4.1.5 Text as Label Attributes The use of text in the GeoSym symbology is limited to those features that have a populated label attribute, such as NAM, for non-DNC products. For DNC, text labels are assigned only for those features whose corresponding S57 object class has a text label assigned per the S52 Presentation Library lookup tables. Text can be placed on point, line or area features and is not rotated. The pivot-point (origin) for text for an area feature is the center of the area. The pivot-point for text for a linear feature is the center of a single segment line. For a multi-segment line, the pivot-point is the midpoint of the run-length of the linear feature.

For those features that require text labels, there will be additional rows in the **sym.txt* file that contain the information required to place the desired text label. Each text label will be based on one or more attributes of the feature. There will be one additional row per specified text label. There are two fields in each text label row that provide the instructions for the symbolization of the text label: *labatt* and *txrowid*.

The *labatt* field will have one or more three-character FACC attribute codes identifying the attribute(s) that defines the text label. The attribute(s) specified may be of type text (such as *nam*), short float (such as *hdp*), or short integer (such as *mcc*). Furthermore, the short integer attribute(s) may be a coded value whose English meaning is defined in the *int.vdt* of the VPF product data or, in the case of abbreviations, in the *textabbr.txt* file. Application software must behave appropriately when accessing label attributes in order to place the desired text label. For example, if the desired text label for a buoy is based on the *ccc* attribute (which is a coded value in DNC) software must translate the code value into the character string associated with that code in the *int.vdt*.

The *txrowid* field will have one or more numbers which are pointers to rows in the Text Label Join File (*textjoin.txt*). This file identifies links to the Text Label Characteristics File (*textchar.txt*) and the Text Label Location File (*textloc.txt*). These files provide the display characteristics and the default placement parameters for the text label.

Multiple entries in the **sym.txt* *labatt* field indicate there is more than one attribute value that needs to be displayed as a concatenated text string. Any *labatt* multiple attribute code entries are comma-delimited and the *txrowid* field must have the same number of comma-delimited values in order to point to the appropriate information in the *textjoin.txt* file. Section 6.8 describes in detail how to use the text assignment files to construct the desired text label.

6.4.1.6 Special Cases The following paragraphs document situations that are unique and require special processing to achieve the correct symbolization.

6.4.1.6.1 Default Symbology Although all efforts have been made to symbolize all features according to the appropriate symbology specifications, should the application software be unable to determine at least one "true" condition for a particular instance of a feature in the VPF dataset, there are default CGMs provided in GeoSym for displaying that feature. The following table identifies which symbol should be used to symbolize an "unknown" point, line, or area feature.

Table 3. Symbols to Place for Unknown Features

Unknown Feature Type	Point Symbol to Place	Linear Symbol to Place
Point (full chart or simplified)	5000.cgm	
Line		5001.cgm
Area - Full Chart	5000.cgm	5001.cgm
Area - Simplified	5000.cgm	0634.cgm

Note that the default symbols should only be placed under two conditions:

1. There is no row in the **sym.txt* file for the feature (fcode)
2. After evaluating all rows in the **sym.txt* file for that fcode, there is no row that results in an evaluation of "true".

6.5 Display Hierarchy. In order to ensure that features that overlap each other are drawn in the correct sequence, the GeoSym application software must allow for the definition of a display order for the features being displayed. GeoSym uses four "nested" methods to define the order in which symbols should be displayed.

1. Display priority
2. Feature delineation
3. Order of rows in **sym.txt*
4. Symbol type

6.5.1 Display Priority. The first criterion to use to determine the order to display features is the display priority. Each row in the **sym.txt*

file defines a display priority number to that specific feature/attribute. The display priority is a value between 0 and 9, where 9 identifies the highest priority. A feature/attribute with a higher priority should be displayed after (on top of) one with a lower priority. For DNC, the values contained in the display priority field are defined based on the S52 lookup tables. Each DNC feature/attribute was mapped to a corresponding S57 object class/attribute. The S52 lookup tables were then utilized to obtain the display priority values assigned to each object class. For all other products, the display priority values were based on cartographic experience with the corresponding hardcopy map/chart.

6.5.2 Feature Delineation. Once the features to be displayed have been sorted based on display priority, they must then be sorted by their delineation (area, line, point). If the display priority is equal among features, then the "painter's algorithm" should be used to display the area features first, followed by the linear features and then the point features

6.5.3 Order of Rows in *sym.txt. The third criterion that affects the display order is the order of the rows in the *sym.txt file. The order will be represented in the id column of each row. Row ids are sequential and consecutive and must be used to correctly display the final symbology for those features whose symbology consists of symbol components, including text labels. Symbology being displayed based on this criterion should be displayed based on the least row id to the greatest row id.

6.5.3.1 Point Features with Components. For point features (e.g., buoys, beacons, lights) that are composed of several symbol components, displaying the components according to the row ids in the *sym.txt file will result in the properly constructed composite symbol. Each symbol component is based on the value of a specific attribute and the components will vary in display priority so should already have been sorted according to that value before examining the row ids in the *sym.txt file.

6.5.3.2 Area Features with Multiple Fills. There are some area features (e.g., Maritime Areas) that require both a solid fill and one or more pattern fills. Since the areasym column can only contain a single CGM reference, there is a separate row in the *sym.txt file for each of the area symbols, as well as for the line symbol and/or point symbol that apply to the specific area feature. These multiple rows will have sequential row ids in the *sym.txt file according to the order in which the symbols are to be displayed on the screen: solid fill, pattern fill (may be more than one), linear boundary, centered point symbol (may be more than one).

6.5.3.3 Features with Text Labels As a general rule, the display priority specified for a feature's text label(s) is the highest value possible, regardless of the display priority assigned to the base feature. In DNC, all text labels will have a display priority of 8 (per IHO S52). In all other products, text labels have a display priority of 9. Therefore, the text labels will by default be displayed on top of all other symbols since the display priority value is the first criterion controlling the order of symbology display on the screen. However, it is possible that a critical feature in a product (e.g., area bridges in DNC) may be assigned the same high priority as the text label for that feature. Since the row(s) defining the text label(s) will always follow the row(s) defining the CGM symbols for a feature, displaying the symbols and text labels according to their row ids

(lowest to highest) in *sym.txt will still result in the correct final symbology for a feature being displayed.

6.5.4 Symbol Type. The final criterion for determining the order for symbol display applies only to area features. Area features can be symbolized by any combination of centered point symbol, linear boundary, solid fill and/or pattern fill. Except for the special case where an area feature's symbology requires both a solid and pattern fill(s) (see section 6.5.3.2), all symbols for an area feature will be specified in the same row of the *sym.txt file. In other words, any or all of the symbol columns may be populated. In these cases, the symbols are to be displayed according to the same "painter's algorithm" that applies at the feature level: the area symbol (from areasym) should be drawn first, followed by the linear symbol (from linesym) followed by the point symbol (from pointsym).

6.6 Color IHO defines the color token as a means by which colors are defined within symbol components. The application of the color token allows a great deal of flexibility for those symbols defined. The need for such flexibility is a requirement of ECDIS, which mandates that symbols be made available for optimal viewing in multiple lighting conditions. Without having to maintain a different symbol set for each lighting condition, a token acts as a slot for assigning defined CIE colors for a given ambient light condition.

It is necessary to inform application software as to what token is being assigned per primitive in order for it to apply the correct local RGB values for display on a local calibrated device based on the ambient viewing conditions. From a symbolization development standpoint, it was also necessary to employ the use of a "viewing" RGB color table for authoring and quality control of symbols designed. Additionally, the GeoSym development task is not limited to the development of symbology to support hydrographic VPF products, but also those of aeronautical and topographic products -- products that have not yet mandated the modification of the appearance of digital displays based on ambient lighting conditions. This variation meant that the design must not only be optimized for hydro exploitation, but also continue to support color static symbolization procedures. The color signature design has been implemented with the use of the CGM color index function. Each CGM file has within it a color table, which has been standardized over the CGM symbol library. Each color index value correlates to one of the 64 defined IHO color tokens. An additional 25 RGB triplet "pseudo-tokens" are colors used to support non-hydro symbolization.

The tokens and pseudo-tokens are numbered using the color index values utilized within the CGM structure, which correlate to the RGB triplets in the external color index table, named color.txt in the GeoSym product, as well as to the application attribute structure, IC_ColorNameTable. An identical table exists within each CGM (the COLRTABLE element) in order for the color index value assignments from the authoring application to be correctly assigned and interpreted. The RGB values are used only in supporting the color definitions for non-hydro symbol primitives (where an appropriately colored token was not available), for authoring and in-house QC, and for reviewer inspection. The RGB values shown for IHO color tokens are derived from a startup table found for a given lighting condition and device. Though more complex versions of CGM implementations may employ the CIE color tables with embedded calibration information, it is felt that this procedure would not be as useful, nor

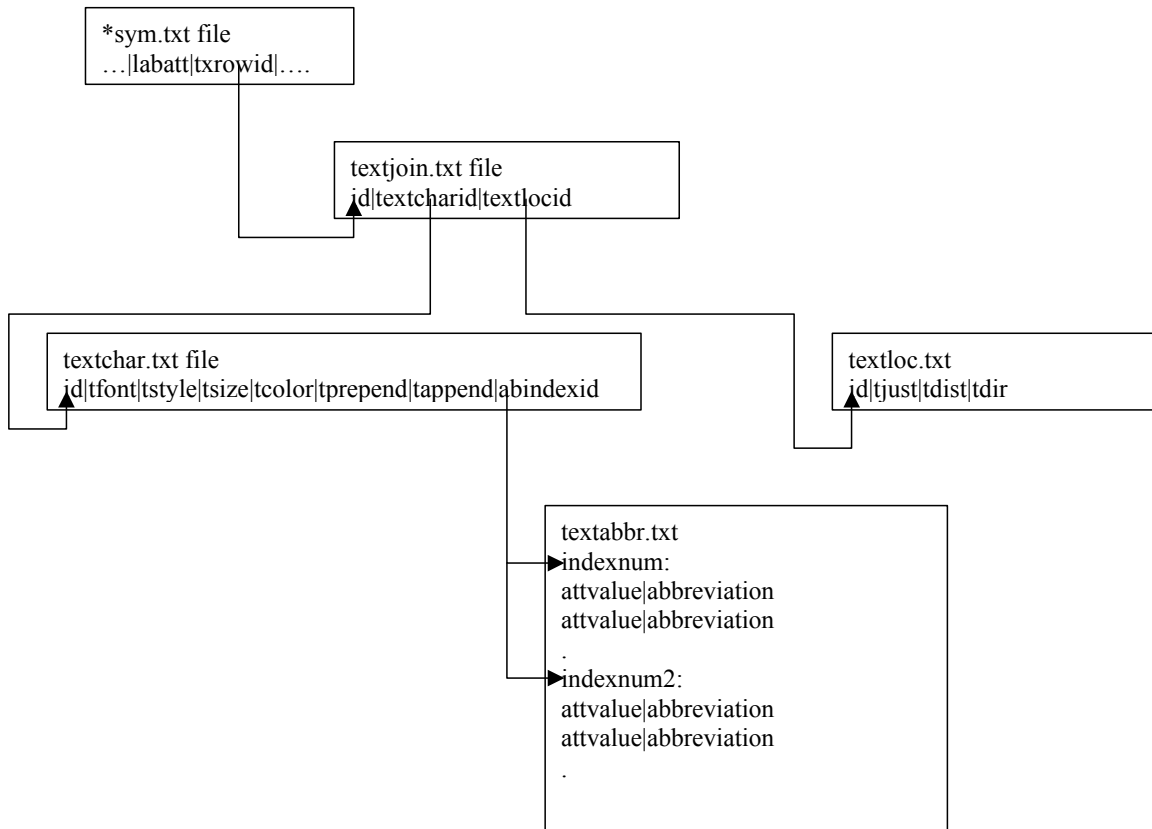
feasible to support since the calibration of devices on so many platforms/sites cannot be anticipated nor predetermined.

The hydrographic viewing applications are to substitute the color index value with that of the assigned color token. There are two methods that may be utilized to accomplish this. The software may access the appropriate row from the color.txt file or it may access the appropriate value from the IC_ColorNameTable element in each CGM. This application attribute structure contains the token or pseudo-token name that corresponds to the RGB triplets in the COLRTABLE element. Once the application has obtained the correct color token, it must obtain the ambient viewing condition (supplied by a user interface) and the appropriate device-dependent color lookup tables, which have been created through CRT calibration procedures for the local viewing device. The application will use these derived RGB values to correctly display the symbology. Other users may make use of the RGB triplets in the display, or substitute using local lookup tables. In this way, the definition of ECDIS color token colors may be done for non-hydrographic product features as well.

6.7 Error Handling Error recovery will be handled in accordance with the error handling plan laid out in ISO/IEC 8632-1 Functional Specification Annex D. The SAMI specifies that error handling extend only to requirements for error recovery. Other aspects of error handling, such as classification of error severity and error reporting, are not required.

6.8 Text Label Instructions The instructions for how to symbolize a desired label attribute depend upon on the specification of several pieces of information. As stated in section 6.4.1.5, the initial items of information are specified in the *sym.txt file in the *labatt* and *txrowid* fields. The values contained in those fields are used to build the desired text label.

The following diagram depicts the relationships between the **sym.txt* file and the text assignment files:



In most cases, the *labatt* field will contain a single 3-character attribute code whose value is to be placed as the text label. For example, many features are labeled with their Name attribute (*nam*). Correspondingly, the *txrowid* field will contain a single value that maps to a row id in the *textjoin.txt* file.

The row id in the *textjoin.txt* file will contain pointers to row id's in two other files, *textchar.txt* and *textloc.txt*. The specified row in the *textchar.txt* file must be accessed to determine the characteristics of the text label to be placed. The specified row in the *textloc.txt* file must be accessed to determine the location to place the text label, relative to the feature being labeled.

In some cases, the desired text label must be constructed to form a concatenated string of two or more attributes. In order to support this capability in GeoSym, both the *labatt* and *txrowid* fields are allowed to contain more than one value. For the *labatt* field, the list of attributes that need to be concatenated together must be separated by a comma and must be in the desired order for final portrayal. For example, it is standard practice in DNC to label light descriptions for navigation aids based on multiple attribute values. A BC020 (buoy) in DNC may contain a *labatt* value similar to the following: *col,per,eol,lvn,mlr*.

For those cases where the *labatt* column contains multiple attribute codes, the *txrowid* field must contain the same number of row id values that will be accessed in the *textjoin.txt* file, with each value corresponding to each attribute code. These multiple values will also be separated by commas, similar to the following: 22,23,24,25,25. Each specified row id in the *textjoin.txt* file must be accessed to determine the characteristics and location of the final concatenated text label.

The following sections describe each text assignment file in detail. Additional information is found in MIL-DTL-89045 where Section 3.5.3.1.3 has descriptions of the headers and formats for the text assignment files and Appendix B presents the text label information contained in the symbology assignment tables..

6.8.1 Text Label Join File (*textjoin.txt*). The purpose of the *textjoin.txt* file is to specify the desired text label characteristics (defined in the *textchar.txt* file) along with the desired location for the text label (defined in the *textloc.txt* file) for each attribute that is specified in the *labatt* field. Each row of the *textjoin.txt* file points to row id's contained in the *textchar.txt* and *textloc.txt* files.

There are three fields in the *textjoin.txt* file: *id*, *textcharid*, *textlocid*.

The *id* field contains a unique, non-negative integer number identifying each row in the *textjoin.txt* file. It is this value(s) that is specified in the **sym.txt txrowid* field.

The *textcharid* field will contain the value of the row id in the *textchar.txt* file that defines the text characteristics for the target label attribute.

The *textlocid* will contain the value of the row id in the *textloc.txt* file that comprises the location information for the target label attribute. When a text label is being constructed from a concatenated set of attribute values, the *textlocid* value defined in the *txrowid* that corresponds to the second and subsequent attributes specified in *labatt* will be set to negative one (-1).

6.8.2 Text Label Characteristics File (*textchar.txt*). The purpose of the *textchar.txt* file is to define the text characteristics of the label being constructed. The fields in the *textchar.txt* file are: *id*, *tfont*, *tstyle*, *tsize*, *tcolor*, *tprepend*, *tappend*, and *abindexid*.

The *id* field contains a unique, non-negative integer number identifying each row in the *textchar.txt* file.

The *tfont* and *tstyle* fields contain numeric codes whose possible values for the specified font and style of the text label are defined in the *code.txt* file.

The *tsize* field provides the body size of the text in pica points (1 pt = 0.351 mm). The default text size is 10 pts, which is also the minimum permissible size.

The *tcolor* field contains the row id of the record in the ASCII color file (*color.txt*) that holds the color token to be used to display the text.

The *tprepend* field will contain a list of characters that are to be placed before the actual attribute value that is being placed as the text label. Similarly, the *tappend* field will contain a list of characters that are to be placed after the actual attribute value that is being placed as the text label. In both of these fields, the mechanism used to define these extra, embedded characters is based on ISO 10646 (Unicode) definitions.

For example, suppose that the desired text label needs to have parentheses placed around the *nam* attribute. The value of the *tprepend* field for the appropriate row in *textchar.txt* would be 0028, which is the defined Unicode value for a LEFT PARENTHESIS. Similarly, the value of the *tappend* field for the same row in *textchar.txt* would be 0029, the Unicode value for a RIGHT PARENTHESIS.

The method for specifying no character to prepend or append is to leave the field empty of any value. The method for specifying more than one character to prepend or append is to separate each Unicode value with a PLUS SIGN (+). For example, suppose that the desired text label is to append the string "m/sec" after the speed attribute. In this case, the *tprepend* field will be empty and the *tappend* field will contain 006D+002F+0073+0065+0063.

The *abindexid* field is used to store an index id that points to the *textabbr.txt* file where a set of abbreviations is defined for replacing the attribute value of the text label with a specified abbreviation. If there is no abbreviation to apply, then the field is left empty.

6.8.3 Text Label Location File (textloc.txt). The purpose of the *textloc.txt* file is to define the display location of the label being constructed. The fields in the *textloc.txt* file are: *id*, *tjust*, *tdist*, and *tdir*.

The *id* field contains a unique, non-negative integer number identifying each row in the *textloc.txt* file.

The *tjust* field contain a numeric code whose possible values for the specified justification of the text label is defined in the *code.txt* file. In addition to the traditional vertical and horizontal justifications, one justification (sounding text) specifies that the attribute is a depth value. In this case, the default display for the text for the integer part of the depth value is centered on the point feature's position and any fractional part is displayed as a subscript.

The *tdist* and *tdir* fields describe the distance in millimeters and the azimuth degrees from North for the default offset of the text label relative to the feature's position. Zero values in these fields indicate that the text is to be displayed at the center of the feature.

6.8.4 Text Label Abbreviation File (textabbr.txt). The purpose of the *textabbr.txt* file is to define the abbreviated text values to be used to replace the actual value stored in the specified label attribute. Each set of abbreviations is associated with a unique index number that is assigned to a row in the *abindexid* field of the *textchar.txt* file.

The format of the `textabbr.txt` file differs from the other text assignment files. In place of a header there is simply a brief description of the format for the ensuing data. Each set of abbreviations begins by having a row that contains an index id followed by a colon. All subsequent rows, until the next row containing an index id and colon, contain two pipe-delimited fields that define the appropriate abbreviation to use based on the attribute's actual value. This means of specifying an attribute's abbreviations was chosen so that application software could read each set of abbreviations into an array of values and then use the associated index value to access the appropriate set of paired values (attribute value|abbreviation).

For example, suppose that a beacon is to be labeled with its `ccc` (color code category) attribute according to specified abbreviations. The `ccc` attribute is a coded attribute. The default behavior for GeoSym application software placing a text label based on a coded attribute is for the software to take the numeric code value from `ccc` and access the `int.vdt` table in the VPF structure in order to obtain the actual text to place as the label. Therefore, if `ccc` contained a value of 1, the text string "Black (B)" would be accessed and placed as the label (this is based on MIL-PRF-89023). However, the actual label desired is the letter B (the specified abbreviation for "black"). This is accomplished by defining a set of rows for the `ccc` attribute in the `textabbr.txt` file and assigning an index id to that set rows. That index id is then specified in the `abindexid` field in the `textchar.txt` file that is used to define the characteristics for the `ccc` attribute. In essence, the `textabbr.txt` file information overrides what is specified in the `int.vdt` table in the VPF structure. The following rows are an example of what is contained in the `textabbr.txt` file for the `ccc` example:

```

1:
1|B
2|Bu
3|Br
.
25|GR
26|GB
.
.
46|WGW
2:  next set of abbreviations
.
.
3:  next set of abbreviations
.
.
```

In the `textchar.txt` file, the row defining the characteristics of the `ccc` text label to place would contain a value of 1 in the `abindexid` field. This is the indicator to the GeoSym application software that the characters to place in the text label based on the coded values of `ccc` are to be obtained from the `textabbr.txt` file instead of the VPF `int.vdt` table.

6.8.5 Example Text Label Assignment File Cases. To conclude the definition and explanation of the text assignment files, Table 4 is provided with a series of use cases for text label placement. These cases illustrate

the information the various files contain and how to correctly interpret that information in each case.

Note: the word NULL is used in this table to indicate blank/empty fields in the textchar.txt file.

Table 4. Text Label Use Cases

Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
1. Label with NAM, 3mm to the right of symbol.	labatt=NAM txrowid=14	id=14 textcharid=1 textlocid=13	id=1 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=NULL	id=13 tjust=1 tdist=3 tdir=90	
Use Case 1 Implementation Instructions: Get contents of NAM and place it to the right of symbol with bottom-left justification. Text should be Sans Serif, Normal, 10 point, CHBLK. There are no special characters or abbreviations to apply.					
Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
2. Label with the TXT attribute 11mm to the left of the symbol.	labatt=TXT txrowid=4	id=4 textcharid=1 textlocid=3	id=1 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=NULL	id=3 tjust=3 tdist=11 tdir=250	
Use Case 2 Implementation Instructions: Get contents of TXT and place it to the left of symbol with bottom-right justification. Text should be Sans Serif, Normal, 10 point, CHBLK. There are no special characters or abbreviations to apply.					

Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
<p>3. Label with PSC and/or MCC centered on the feature (with a space separation if both labels apply). Abbreviate both attributes according to Chart 1 values. Three possible combinations are considered.</p>	labatt=PSC txrowid=10	id=10 textcharid=11 textlocid=1	id=11 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=2	id=1 tjust=5 tdist=0 tdir=0	2: 1 brk 2 crs . . 12 ten . .
	labatt=MCC txrowid=11	id=11 textcharid=15 textlocid=1	id=15 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=3	id=1 tjust=5 tdist=0 tdir=0	3: 4 Vol Ash 8 bls . . .
	labatt=PSC,MCC txrowid=29,30	id=29 textcharid=11 textlocid=9	id=11 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=2	id=9 tjust=6 tdist=0 tdir=0	
		id=30 textcharid=12 textlocid=-1	id=12 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=0020 tappend=NULL abindexid=3		

Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
<p>Use Case 3 Implementation Instructions: For the first combination where PSC<>0 AND MCC=0, get value of PSC and use it to access the textabbr.txt file for those rows associated with index 2. Use the value of PSC to obtain the desired label text and then place it centered on the origin of the feature as Sans Serif, Normal, 10 point, CHBLK.</p> <p>For the next combination, where PSC=0 AND MCC<>0, get value of MCC and use it to access the rows in textabbr.txt associated with index 3. Use the value of MCC to obtain the desired label text and then place it centered on the origin of the feature as Sans Serif, Normal, 10 point, CHBLK.</p> <p>For the final combination, where PSC<>0 AND MCC<>0, recognize that a multi-attribute text string is to be built (based on commas in labatt field). Get value of PSC and begin building the text label with Sans Serif, Normal, 10 point, CHBLK. The PSC parameters control the location for the entire concatenated text string (note that the MCC attribute has textlocid of -1). Use value of PSC to access the rows in textabbr.txt as described above. Use the value of MCC to obtain the desired abbreviated text label as described above. Then concatenate the MCC abbreviation with a space separator. Place resulting string justified center-right on the origin of the feature.</p>					

Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
<p>4. Label with col,per,eol,lvn,mlr in a concatenated string. Append "s" after per, "m" after eol, "M" after lvn and "M after mlr. Place entire string 10mm to the right of the symbol.</p>	<p>labatt=COL,PER,EOL,LVN,MLR txrowid=22,23,24,25,25</p>	<p>id=22 textcharid=1 textlocid=19</p> <p>id=23 textcharid=5 textlocid=-1</p> <p>id=24 textcharid=6 textlocid=-1</p> <p>id=25 textcharid=7 textlocid=-1</p>	<p>id=1 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=NULL tappend=NULL abindexid=NULL</p> <p>id=5 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=0020 tappend=0073 abindexid=NULL</p> <p>id=6 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=0020 tappend=006D abindexid=NULL</p> <p>id=7 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=0020 tappend=004D abindexid=NULL</p>	<p>id=19 tjust=1 tdist=10 tdir=90</p>	
<p>Use Case 4 Implementation Instructions: Recognize that a multi-attribute string is to be built (based on commas in labatt field). Obtain value of COL and begin building text label with Sans Serif, Normal, 10 point, CHBLK. The COL parameters control the location for the entire concatenated text string (note that the trailing attributes have textlocid's of -1). Obtain value of PER and concatenate it using the same characteristics, except prepend it with a space and append to it a lowercase "s". Do the same for EOL, except prepend with a space and append a lowercase "m". Do the same for LVN, except prepend with a space and append with uppercase M. Since 25 is specified again as the last value in txrowid, concatenate MLR the exact same way LVN was. Place resulting string 10mm to the right of the symbol with bottom left justification.</p>					

Use Case	*sym.txt	textjoin.txt	textchar.txt	textloc.txt	textabbr.txt
5. Label CCC 11mm to the upper-left of the symbol. Abbreviate it and enclose in parentheses.	labatt=CCC txrowid=6	id=6 textcharid=13 textlocid=5	id=13 tfont=0 tstyle=0 tsize=10 tcolor=17 tprepend=0028 tappend=0029 abindexid=1	id=5 tjust=3 tdist=11 tdir=300	1: 1 B 2 Bu . 12 R 14 Vi . 30 YBY
Use Case 5 Implementation Instructions: Get value of CCC. Since abindexid has a populated value, use the value of CCC to access abbreviations associated with index number 1 in textabbr.txt instead of accessing rows from the VPF int.vdt table. Once the abbreviation is determined, place it enclosed in parentheses 11mm to upper left of symbol with bottom right justification.					

7. NOTES

7.1 Intended Use. This handbook is intended to supplement the information contained in MIL-DTL-89045 and inform users of GeoSym about key items that warrant special attention in either the development of software to exploit GeoSym or in the expansion of the symbol assignment files to include future products. GeoSym® is a registered trademark of the U.S. Government, National GeoSpatial-Intelligence Agency.

7.2 Supersession data. This handbook supersedes all previous versions of this document, including the Draft Military Handbook Geospatial Symbols for Digital Displays (GeoSym), dated 20 February 1998, and the Draft Military Handbook Geospatial Symbols for Digital Displays (GeoSym), Prototype 4, dated 5 June 2000, which supplemented the design of earlier versions of GeoSym. Also superseded is the Draft Military Handbook Geospatial Symbols for Digital Displays (GeoSym), Edition 1, dated 3 December 2003. This handbook is the official initial release of this document and corresponds with the final design of GeoSym.

7.3 Subject term (key word) listing.

ActiveCGM
Application attribute structure
Attribute expression
Buoys, Beacons, Lights
CGM
Color token
Component
Composite
Display Priority
GeoSym Line Style
Soundings
Symbology
Text Label
Version 4

ADDITIONAL REFERENCES

A.1 SCOPE

A.1.1 Scope. This appendix provides a list of additional reference sources for the elements of symbology information contained in section 4 herein.

A.2 APPLICABLE DOCUMENTS

This section is not applicable to this appendix.

A.3 DEFINITIONS

This section is not applicable to this appendix.

A.4 ADDITIONAL REFERENCES FOR SYMBOLOGY TOPICS

Burger, Peter, Gillies, Duncan, Interactive Computer Graphics, Addison Wesley, Wokingham, 1989.

Dent, Borden, Principles of Thematic Map Design, Addison-Wesley Publishing Company, 1985.

Foley, James D, Van Dam, Andries, Fundamentals of Interactive Computer Graphics, Addison Wesley, Reading, 1983.

Henderson, Lofton R Mumford, Anne, E., CGM Handbook, Academic Press, London, 1993 ISBN0125105606.

Itten, Johannes, The Elements of Color, Translated by Ernst Van Hagen. Van Nostrand Reinhold, New York, 1971.

Robinson, A.R., Sale, R., Morrison, J., Elements of Cartography, John Wiley & Sons, 1978.

Science Applications International Corporation, (SAIC) 1992. "Digital Symbology Technical Considerations Report". Prepared under contract NIMA600-91-D-0001 for the U.S. Defense Mapping Agency Systems Center, Fairfax, Virginia.

Science Applications International Corporation, (SAIC) 1993. "Final Report Digital Symbology, Phase III Draft". Prepared under contract NIMA600-91-D-0001 for the U.S. Defense Mapping Agency Systems Center, Fairfax, Virginia.

Science Applications International Corporation, (SAIC). "Vector Product Format Symbology Military Handbook", March 1996

MIL-HDBK-857
Concluding Material

Custodians:

Air Force - 09
Army - TI
Navy - NO
DISA - DC2

Preparing Activity

NGA - MP
(Project GINT-0297)

Review activities:

Air Force - 33
Navy - CG
DLA - DH
Marine Corps - MC
NSA - NS
NORAD - US

NOTE: The activities listed above were interested in this document as of the date of this document. Since organizations and responsibilities can change, you should verify the currency of the information above using the ASSIST Online database at www.dodssp.daps.mil.