(4)

Naval Ocean Research and Development Activity

[illegible]

Report 194

Design and Implementation of the Digital World Vector Shoreline Data Format

Maura Connor Lohrenz
Mapping, Charting, and Geodesy Division
Ocean Science Directorate

Foreword

This report describes the data structure and format that NORDA designed for the Defense Mapping Agency's (DMA) digital World Vector Shoreline (WVS) data set. Methods of accessing the data set, current and future naval applications of WVS, and suggested future additions to the data set are also discussed. The WVS data structure and format were designed to meet Navy requirements for a digital global shoreline map, as outlined in CNO OP-096 memo, serial number 006C/50322906, 5 August 1985. Applications and further suggestions are based on responses to a questionnaire prepared by NORDA to survey DoD users who evaluated a WVS prototype of the Mediterranean Sea area. NORDA's compilation of Navy feedback to DMA's WVS prototyping effort was requested by the Oceanographer of the Navy (CNO OP-096).

W. B. Moseley
Technical Director

A. C. Esau, Captain, USN
Commanding Officer

Executive Summary

This report defines the data structure and format that NORDA designed for the Defense Mapping Agency's (DMA) digital World Vector Shoreline (WVS) data set. Methods of accessing the data set, current and future applications for WVS, and suggested future additions to the data set are also discussed. Applications and suggestions are based on responses to a questionnaire prepared by NORDA to survey DoD users who evaluated the latest WVS prototype, the Mediterranean Sea area.

NORDA's role in designing the WVS data format and structure was a direct result of our recommendations to DMA on their original WVS prototype data set and format, presented in NORDA Reports 146 and 154. In these reports, NORDA recommended that specific changes be made to DMA's original WVS to be sure that it would meet the Navy's requirements for a digital global shoreline map (according to CNO OP-096 memo, ser 006C/50322906, 5 August 1985). In response, DMA requested that NORDA assist them in the design and implementation of a new WVS format and data structure. This new product was completed and distributed to potential Navy and other DoD users in March 1987.

Acknowledgments

This work was sponsored by the Defense Mapping Agency (DMA) under Program Element 980101. I would like to thank Mr. Bruce Seaton, the WVS project manager at DMA, and Ms. Gail Langran, Ms. Mary Clawson, and Mr. Jim Braud of NORDA's Mapping, Charting, and Geodesy Division for their helpful comments during this study. I would also like to thank CDR Richard Blumberg at CNO OP-096 and CDR Dudley Leath and Mr. Keith Archer, of DMA's Hydrographic-Topographic Center, for their help and guidance in designing the NORDA/DMA WVS data format.

Contents

Design and Implementation of the Digital World Vector Shoreline Data Format

1. Introduction

Summary of the WVS Prototyping Effort

In FY85, a NORDA study of naval digital mapping, charting, and geodesy (MC&G) data requirements revealed that a large number of Navy systems and commands currently use or plan to use a digital world vector shoreline data set (Langran and Connor, 1986). Table 1-1 lists potential Navy users of such a data set, based on several questionnaires circulated by the Defense Mapping Agency (DMA) and NORDA in 1986 and 1987. The CIA's World Data Banks I and II (WDB I and WDB II) are supporting most current users. Because continued use of WDB I and WDB II by the Navy was deemed inadvisable for various reasons, the Oceanographer of the Navy (CNO OP-096) requested that DMA produce an improved data set, named the World Vector Shoreline (WVS). Table 1-2 lists the WVS's required characteristics (CNO OP-096 memo, serial 006C/50322906, 5 August 1985).

To save production costs, DMA developed a method of deriving vector shoreline data from its Digital Land Mass Blanking (DLMB) data, a land mask gridded at 3 arc-second intervals. Shorelines were processed by contouring the land/sea interface in DLMB and generalizing the vertices to several different resolutions. Political boundaries and country/state names were added from separate sources.

In March 1986, DMA released a prototype WVS for four groups to evaluate: NORDA, the Naval Undersea Systems Command, the Naval Air Development Center, and the Joint Cruise Missile Program Office. As the Navy's lead MC&G research laboratory, we at NORDA decided to use the three weeks available for this project to assess the data set's processing efficiency and the suitability of its content. No time was allotted for assessing data quality, although we strongly recommend such a study. The results of NORDA's evaluation were presented in two reports: "Recommendations on DMA's Standard Linear Format" (Langran, et al., 1986) and "Recommendations on the World Vector Shoreline" (Langran and Connor, 1986). The first concentrated on DMA's original data format for WVS and recommended several major changes. The second report evaluated WVS itself and recommended certain changes and possible future additions to the data set.

Our primary recommendations to DMA can be summed up in one paragraph from NORDA Report 154:

> The WVS will be no less a DMA product than a map or chart; therefore, it merits a format designed to hold the data efficiently in an ordering that meets its users' needs. The WVS format should be designed to promote one-pass input. Logical records should be small for easy handling, and . . . WVS data should be broken into similarly sized pieces so a fixed format can be effectively applied.

We felt that DMA's Standard Linear Format (SLF) did not meet these requirements and that a new format should be designed specifically for the WVS. NORDA Report 146 outlines our reasons for rejecting SLF as the WVS format.

In response to NORDA's recommendations, DMA requested that we assist them in designing a new WVS format and data structure that would best meet the Navy's requirements. This new format was completed and a new prototype data set was distributed to potential Navy and other DoD users in March 1987. The data set was accompanied by a questionnaire that addressed such issues as current and future applications for the WVS and desirable additions or changes to the data set.

Order of Report

The body of this report describes the general layout of DMA's new prototype WVS, according to NORDA's WVS format. Section 2 gives a WVS file overview, explaining the file organization, data structure (termed "chain-node") and data format. Section 3 describes three methods of accessing the WVS data: reading simple "spaghetti" chains of X, Y coordinates, sequentially accessing a window of data, and directly accessing a window. Section 4 presents responses to the WVS questionnaire that DoD users answered. Section 5 summarizes the report. Section 6 cites references.

Table 1-1. Potential Navy users of WVS.

System	Computer Model(s)	Bits/Word (respectively)
AEGIS	CDC-865, VAX Cluster, IBM-PC, UYK-7, UYK-43B	60, 32, 16, 32, 32
APP	undetermined	
ACDS	UYK-43	32
ASWOC (baseline)	CP-901	8
ASWOC (upgrade)	undetermined	
CV-ASWM	UYK-7, UYK-43	32
CCS-MK1	UYK-7	32
DSAT	VAX 11/780	32
DUET	VAX 11/780	32
FHLT	UYK-7	32
HYCAT	UYK-44	16
ICAPS	(15 different models)	
NEAT	VAX 11/780	32
NISC-OFM	VAX 11/750	32
NWSS	undetermined	
OPINTEL	VAX 11/780, SUN 3/160, IBM-PC/2-80	32, 32, 32
OTH	VAX 11/785	32
P3-UPDATE	CP-901, UYK-14	8
POST	undetermined	
SACC	Victor AN-ASQ114V	8
SEABASS	VAX 11/780	32
SEANYMPH	UYK-20	16
SEA WATCH II	CYBER 170/730	60
SEA WATCH III	undetermined	
SOCC	undetermined	
STT	ROLM-1666, HP-9350	16, 32
TERPES	CP-808, UYK-43	30, 32
TESS	HP-9020A	32
TOMAHAWK	ROLM-1666, HP-9020, VAX 11/780, mVAX	16, 32, 32, 32
TWCS	UYK-19, UYK-64	8, 16

Table 1-2. Required WVS characteristics.

- WVS must use a minimum number of points to display the shoreline at the desired scale, since some systems have limited storage and processing capabilities.
- WVS must support output at scales ranging from 1:250,000 to 1:12,000,000.
- WVS must use a vector format.
- WVS must have an accuracy comparable to paper products.
- WVS must identify the shoreline's land and water sides to allow color fills.
- WVS must include international boundaries, and include their maintenance status (recognized, disputed, indefinite).
- Country labels must be associated with international boundaries.
- WVS must be compatible with DMA's DTED, DFAD, HOD, PPDB, and the future digital bathymetric and contour data bases.
- WVS must embed certain characteristics (particularly feature counts and other data size estimates) to promote automated data entry.
- A programmer's appendix to the documentation must supply illustrations, examples of data content, and software programs to assist users in reading the data.

2. WVS File Overview

File Characteristics

Tapes: For portability, WVS data is ASCII coded and written to unlabeled ANSI-standard magnetic tapes. To reduce storage requirements, 6250 bpi tape is preferred; however, 1600 bpi tape can be substituted if specified by a user.

Record Size: Records in NORDA's WVS file have a fixed length of 48 bytes and are blocked by 200 on tape for a standard block size of 9600 bytes. Optionally, block size can be decreased in units of 48 for users with block size limitations.

File Content: WVS Edition One describes shorelines, boundaries, and country names, which are identified by Feature Attribute Coding Standard (FACS) codes of 2A010, 6Axxx, and 9A010, respectively. Boundary data are subcoded from 6A000 to 6A999 (see Appendix A) according to type and disputation status. Future WVS editions may include lakes, rivers, major cities, etc. The WVS format has been designed to accommodate such expansions.

Appendix B is a sample listing of the WVS file, including the file header and a sample of each record type. Figure 2-1 is a plot of the WVS prototype: the Mediterranean Sea area (10°W to 42°E and 25°N to 50°N). Table 2-1 lists general WVS data characteristics not included in the file itself.

Table 2-1. WVS information not included in file headers.

Security classification:	Unclassified
Data type:	Geographical; i.e., LONG, LAT coordinates
Horizontal units of measure:	Degrees for origins, seconds for data points
Geodetic datum:	World Geodetic System, 1972
Ellipsoid:	World Geodetic System, 1972
Sounding datum:	Mean high water
Digitizing system:	AGDS

Data Resolution and Scale: The WVS was derived from DLMB, a gridded land mask with data points spaced at 3 arc-second intervals. DLMB was digitized from nautical maps and charts with an average scale of 1:250,000. Both DLMB and the final WVS vectors are therefore described as having a scale of 1:250,000. That is, data quality of the DLMB and WVS products is considered to be comparable to that of the original data for scales of up to 1:250,000. In comparison, WDB II was digitized from sources with an average scale of 1:3,000,000. Thus, the resolution of WDB II (and its data quality at any given scale) is much lower than that of WVS. Figure 2-2 depicts the difference in data quality and resolution between WDB II and WVS. It shows the coastlines of the Odessa region in the Black Sea from both data sets.

Global Partitioning: Data are partitioned into 1-degree cells that wrap the globe sequentially in the X (longitude) direction. Figure 2-3 illustrates this idea with a map of the Mediterranean Sea area that has been divided into 5-degree cells for simplicity. Appendix C describes a method to locate the cell that contains any given point.

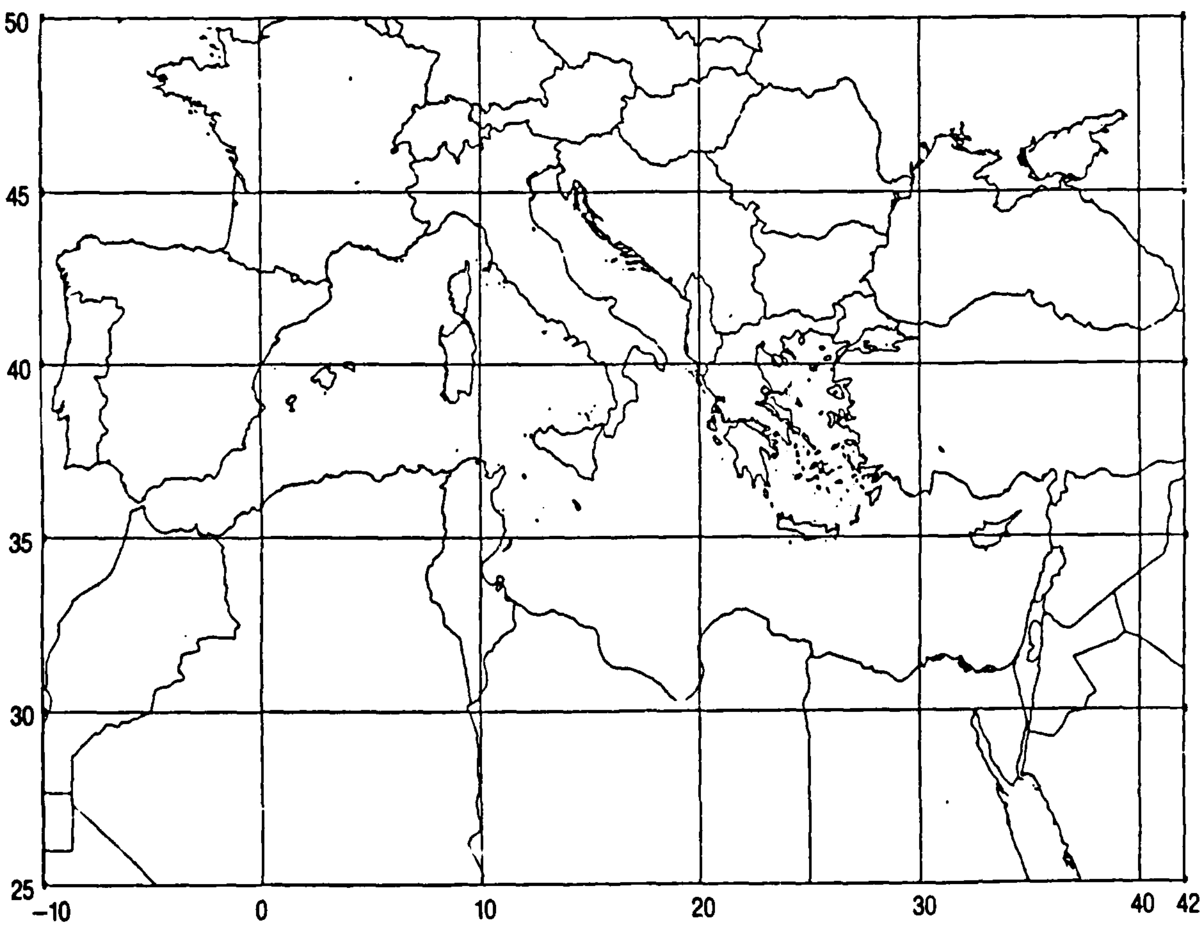
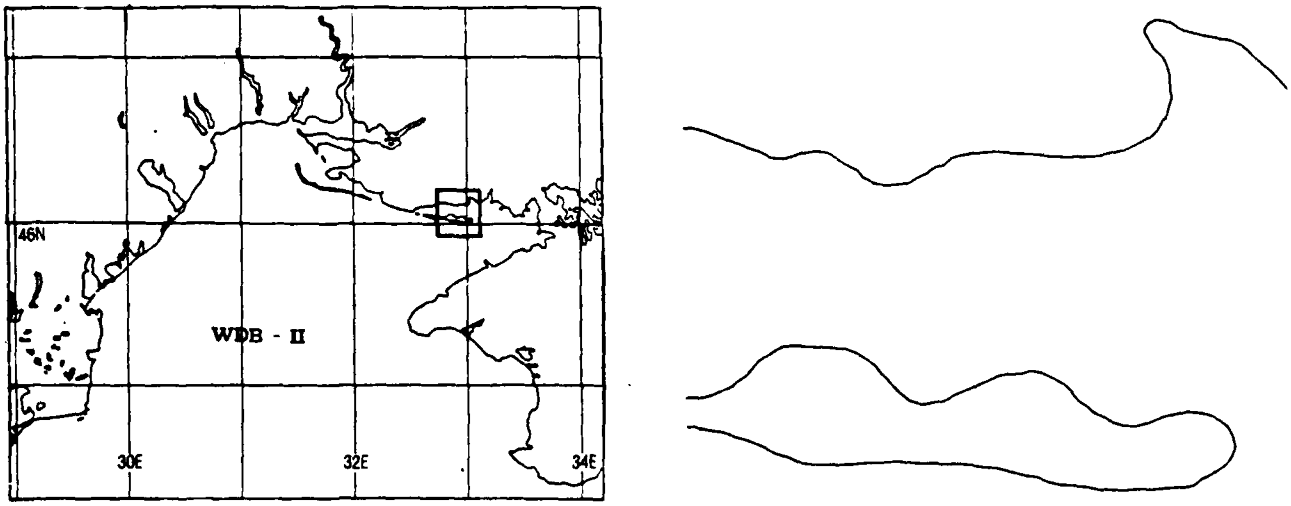


Figure 2-1. Plot of WVS prototype: Mediterranean Sea.

(a)
WDB II



(b)
WVS

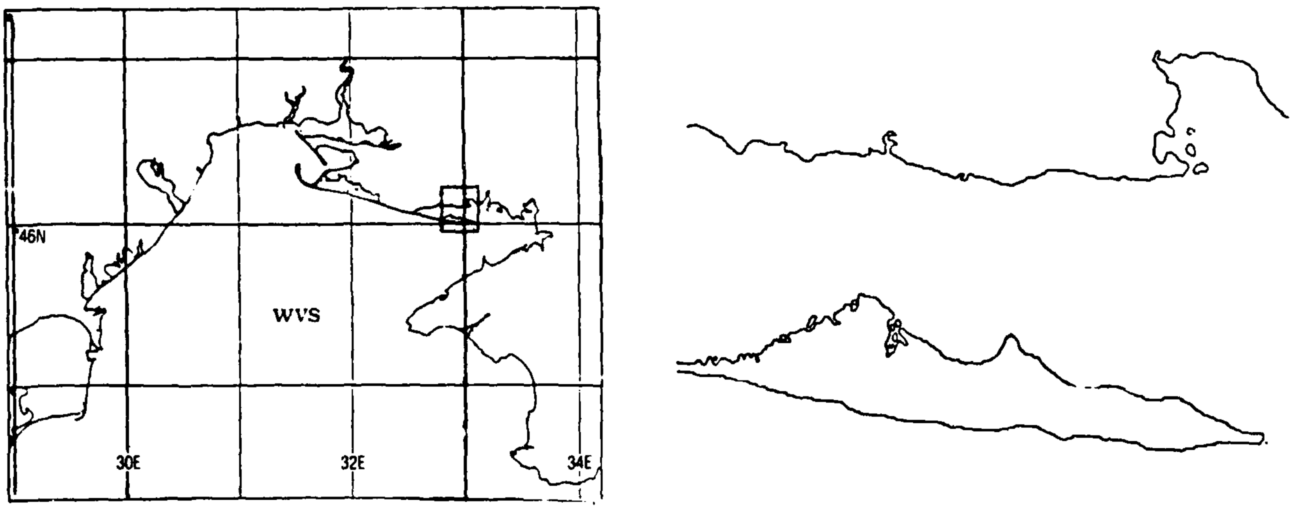


Figure 2-2. Comparison of WDB II and WVS: data resolution. Area shown is the Odessa region in the northern Black Sea. (a) WDB II was digitized at an average scale of 1:3,000,000. (b) WVS was digitized at an average scale of 1:250,000. Therefore, the relative quality of WVS will be much greater than WDB II at any given display scale. The difference in resolution between WDB II and WVS is increasingly evident as display scale increases: maps on the right are enlargements (approximately 16 times) of maps on the left, and the difference in resolution between WDB II and WVS is even more apparent here. Figures are courtesy of the Naval Ocean Systems Center (Mr. Larry McCleary and Mr. Tom Wescott, Code 422).

Coordinate Values: All coordinates are signed decimal values ordered by (longitude, latitude)—NOT (latitude, longitude)—to align with the Cartesian system's standard (X, Y) ordering. Coordinates of file and data origins have an implied decimal point at 0.0001. A "+" indicates east or north and a "−" indicates west or south, following the International Standard (#6709-1983). Thus, an origin coordinate of (365000, −750000) refers to 36.5°E, 75°S. Coordinates of vertices, however, are given in positive 0.1 seconds of offset from their cell origin. Since a cell origin is the lower left corner of a cell, a Y data value (YSECONDS) of 003480 in a cell whose latitude origin (LATCELL) is −750000 may be computed as follows:

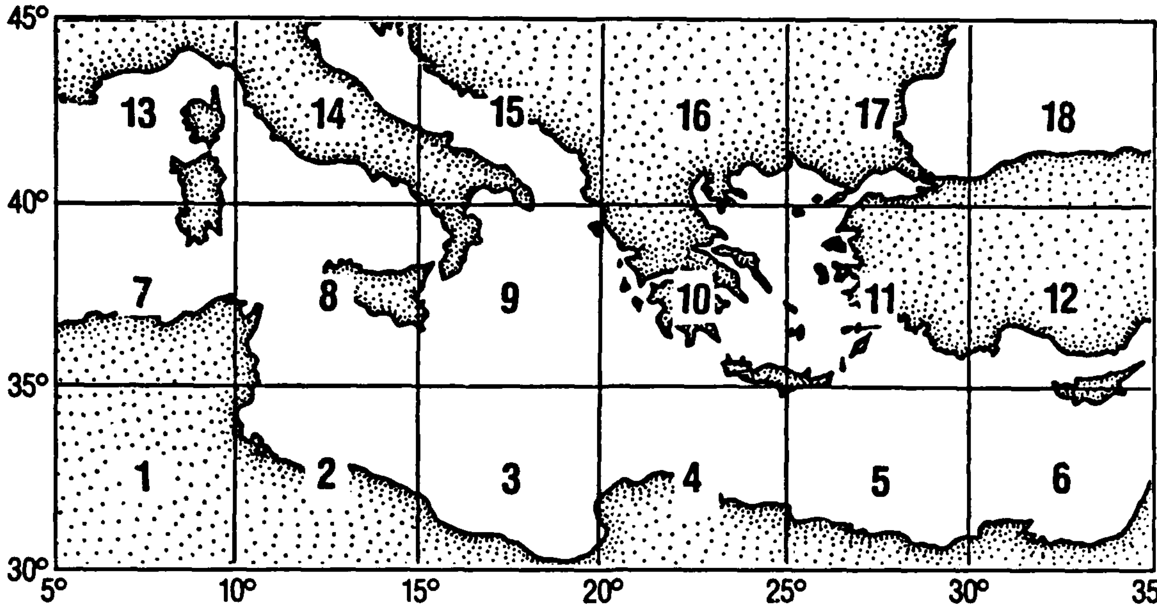


Figure 2-3. Example of cell-numbering scheme used in WVS. For clarity, cells are shown as 5-degree squares and are numbered with reference to the southwest corner. In the actual WVS file, however, cells are 1-degree squares and are numbered with reference to a global map origin (−180, −90).

latitude degrees = (LATCELL * 0.0001) deg
+ ((YSECONDS * 0.1) sec / 3600 sec/deg)
= (−750000 * 0.0001) deg
+ ((3480 * 0.1) / 3600) deg
= −75.0000 deg + 0.09667 deg
= −74.9033 degrees.

File Organization and Data Structure (Chain-Node)

Overall file organization is depicted in Table 2-2. A file header (4 records long, or 192 bytes) starts each WVS file. Explanatory text records, if needed, follow the file header (the number of text records is given in the file header). The rest of the file consists of data cells, each of which contains all features and segments for a given 1-degree-square area on the earth. A WVS file covering the globe contains 360 by 180 = 64,800 cells numbered sequentially from 1 to 64,800. The complete WVS format is listed in Appendix D.

The WVS data structure was designed to minimize redundant data storage. Since a sequence of vertices (from here on referred to as a segment) may outline more than one feature, features are cross-referenced to their segments (and vice versa) via unique alphanumeric feature and segment keys. In this way, a segment shared by more than one feature is stored only once. The segment header record contains a list of all the features that the segment delineates. Likewise, each feature in the map file is stored only once. The feature data record contains a list of all the segments that delineate that feature. This method of data storage was also utilized in SLF and is referred to as "chain-node." In sum, one feature may be comprised of many segments, or one segment may delineate several features, but in either case, each segment and feature is stored only once in chain-node.

To support advanced uses, all WVS segments define which feature types lie to the left and to the right of a segment (when moving forward through its vertices). For example, the left side of a shoreline segment may be land and the right side water. For boundary segments, the countries lying to the left and right are designated. For "left" and "right" to have any relative meaning, the order of vertices within a segment for a given feature must be provided. This has the added benefit of allowing a feature to be outlined with a continuous (unlifted) penstroke.

In describing a particular feature, WVS assigns one of six different directional indicators to each of the segments delineating that feature. For example, if a segment's vertices are to be read in reverse of the order in which they were stored, the direction would be "R" for reverse. Table 2-3 lists the six possible directional indicators, stored as one-character codes in the feature data records of the WVS file.

Figure 2-4 shows the basic differences between conventional "sequential" data storage and chain-node data structuring, together with this directional capability. Figure 2-4 (a) shows three adjacent areal features as they might appear on a map. In the sequential storage method, each feature would be treated as a separate polygon, digitized independently (b), and stored as a

Table 2-2. WVS file organization. Each record is 48 bytes

FILE_HEADER1 FILE_HEADER2 FILE_HEADER3 FILE_HEADER4
TEXT_RECORD1
:
TEXT_RECORDn
CELL #1: HEADER record FEATURE records . . . SEGMENT records . . .
CELL #2: HEADER record FEATURE records . . . SEGMENT records . . .
:
CELL #N: HEADER record FEATURE records . . . SEGMENT records . . .

Table 2-3. Directional indicators for WVS segments. The 1-character code is stored in Feature Data Records (see Appendix D).

Code	Direction	Description
F	Forward	Points are to be retrieved in the same order as stored.
R	Reverse	Points are to be retrieved in reverse of their stored order.
D	Disjoint Forward	Same as Forward, but the segment is in a disjointed part of the areal or linear feature, e.g., the continuation of a country that contains one or more islands.
E	Disjoint Reverse	Same as Disjoint Forward, but the points are to be retrieved in reverse of their stored order.
I	Inside Forward	Same as Forward, but the segment describes a hole or island within an areal feature; e.g., to delineate the shorelines of a lake containing islands, the island segments are designated as being inside the lake.
J	Inside Reverse	Same as Inside Forward, but the points are to be retrieved in reverse of their stored order.

string of (X, Y) coordinates. The common boundary between two abutting features would be digitized twice, and the two versions may not coincide. This often results in gaps or overlaps along the boundary (c).

In the chain-node method, the segments that comprise the features are independently digitized as shown in Figure 2-4 (d). The common boundary between adjacent features is digitized and stored only once, so that the features abut precisely. The coordinates of each segment are stored with a unique segment ID number and a list of the features associated with that segment. Likewise, each feature is stored with a unique ID and a list of the segments (with directional indicators) that define it. Therefore, features may be displayed either separately (e) or together (f) with no overlaps or gaps along the boundaries between them.

Cell Organization

Data are collected into cells to facilitate windowing, repartitioning, and future expansions of the data set. Each data cell starts with a one-record header containing a cell identification number, the cell origin, the number of features and segments in the cell, and the number of 48-byte feature and segment records the cell contains. The feature and segment record counts provided in the header can be used to skip directly to the cell's segment data for simple graphic applications, or to skip over one cell to the next cell (see Section 3).

The cell header also contains a code to indicate whether the cell contains all land, all water, or a combination of both. This code is valuable in performing color fills of land or water areas. Cells with no segment data are represented by a cell header record with no data records. Headers for empty cells are included for areal fills, algorithmic simplicity, and future data enhancements.

3. Reading the WVS Data

Reading the Complete File

Table 3-1 lists the Fortran format statements that may be used to read each record type in the WVS file. Appendix D gives a complete description of the variables in each WVS record type. Appendix E contains a Fortran program to read the complete WVS file from tape.

Table 3-1. Fortran format statements for each WVS record type.

FILE HEADER #1:	FORMAT (A20, I1, I2, 1X, A8, 1X, I4, 1X, 2I5)
FILE HEADER #2:	FORMAT (I8, I7, I3, 2(I8, I7))
FILE HEADER #3:	FORMAT (4(I7, 1X), I4, 1X, I3, 1X, I2, 5X)
FILE HEADER #4:	FORMAT (I7, 1X, I5, 1X, I9, 1X, 2(I4, 1X), I6, 1X, I4, 3X)
TEXT RECORDS:	FORMAT (A48)
CELL HEADER:	FORMAT (2A1, I1, I6, I8, I7, 2I5, 2I7)
FEATURE HEADER:	FORMAT (A3, I7, A1, A5, A24, 2I3, I2)
EXTRA FEA.ATTRIBUTES:	[varies with feature type]
FEATURE DATA:	FORMAT (6(I7, A1))
SEGMENT HEADER:	FORMAT (A3, I7, I5, 2I2, I5, 3(I7, A1))
EXTRA FEATURE REFERENCES:	FORMAT (6(I7, A1))
SEGMENT DATA:	FORMAT (8I6)

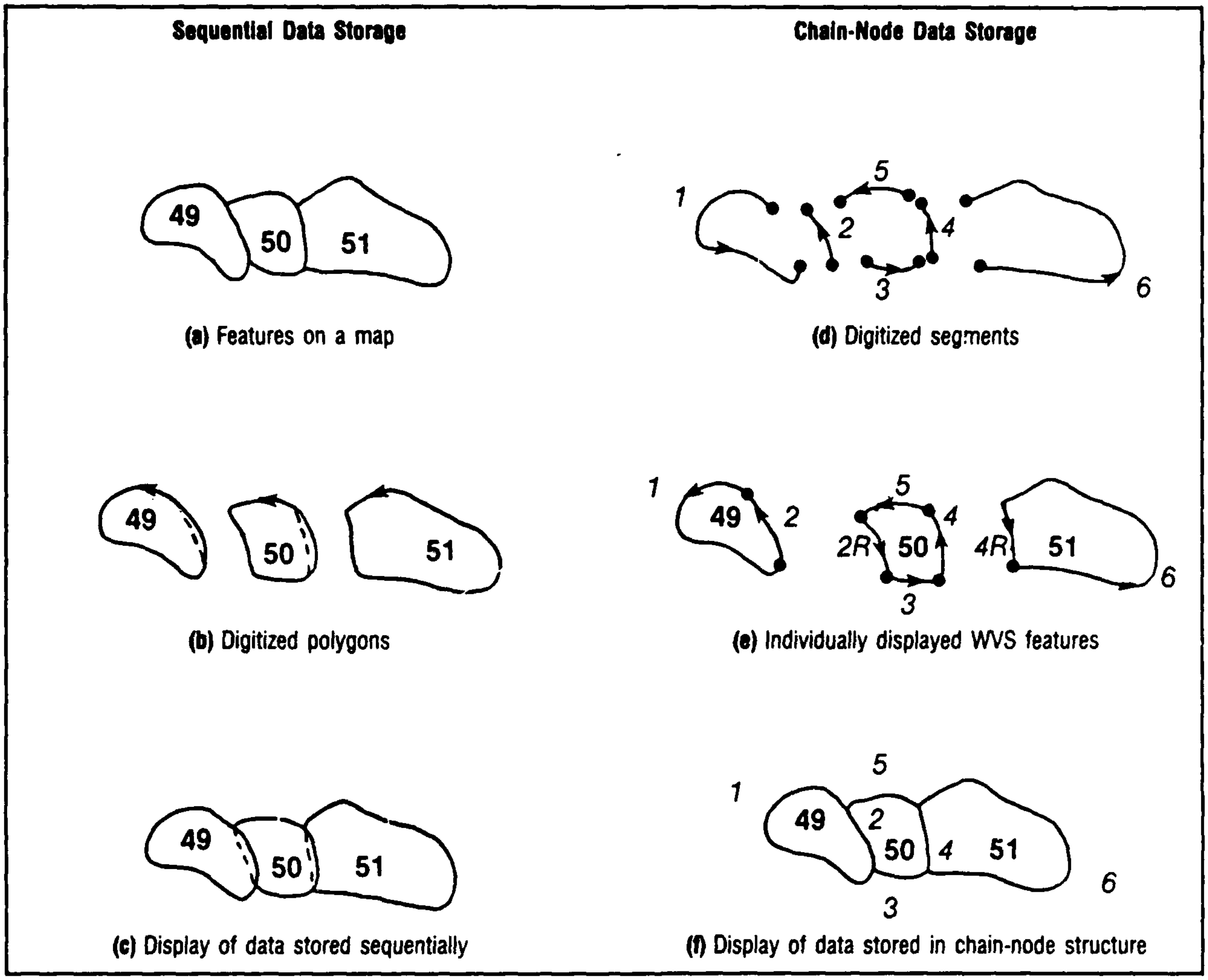


Figure 2-4. Sequential versus chain-node methods of cartographic data storage (Defense Mapping Agency, 1985). Arrows indicate the ordering of coordinates within a segment; solid lines represent digitized outlines; dotted lines represent true boundaries.

Extracting "Spaghetti" Data

Since counts of features, segments, cells, vertices, etc., are provided in the WVS file, users who require only a simple shoreline/political boundary plot may extract strings of segments (also referred to as "spaghetti" data) by selectively reading the file:

1. Read file headers.
2. Skip any text records after the file headers.
3. Read the cell header: If data in this cell is needed, continue. If not, skip to next cell and repeat this step.
4. Skip all feature records in the cell.
5. Read segment header record, skip feature references, and read vertices.
6. Repeat steps 3 through 5 until all needed vertices have been extracted.

Extracting a Window of Data

Rectangular subsets or windows of the WVS file may be selected by extracting and aggregating cells of data. If the window is to be selected directly from tape, the cells must be accessed sequentially as described below. A faster, direct access method is also decribed for those who will store their WVS data on disk.

Sequential Access

1. Compute the numbers of the cells needed for the desired window (see Appendix C).
2. Starting at the top of the file, read each record. When a cell header is encountered, determine if it is one of the needed cells.
3. If the cell is needed, collect its data. Otherwise, skip all the data records in that cell (to the next cell header).

Direct Access

This method assumes that the file can be loaded onto a disk or other media allowing rapid direct access.

1. Create a look-up table that lists the number of each cell's first record (i.e., the Cell Header Record) in the WVS file. Record 1 in the look-up table points to the location of Cell 1, Record 2 to Cell 2, etc. The look-up table is created once, when the file is initially loaded from tape.

2. Compute the numbers of the cells needed for the desired window (see Appendix C).

3. Access the look-up table by cell number, access the cell header, and collect the cell contents. For further illustration, see Table 3-2.

Table 3-2. Using a cell table look-up file.

Sample listing of a cell table look-up file	Example: Find the starting WVS record number of cell #5:
1	READ (ITABLE,REC=5) NREC
122	
345	where ITABLE is the logical unit number (LUN)
908	assigned to the table look-up file and NREC is
→1609	the starting record number of cell #5 in the WVS
2099	file; in this example, NREC = 1609. Now, to
3437	access the first record (the header record) of
3678	cell #5:
4980	READ (IWVS,REC=NREC) CELLHEADER
5002	
...	where IWVS is the LUN assigned to the WVS
...	file and CELLHEADER is the 48-byte cell
...	header record. From here, you can read the
	feature and segment records for cell #5.

4. User Responses to WVS Questionnaire

This section discusses users' responses to the questionnaire that was distributed in March 1987 with the prototype WVS data set. The discussion is divided into six main topics: user background, system constraints (memory, data storage and access), WVS applications, WVS documentation, WVS data quality, and future WVS requirements. The complete questionnaire is listed in Appendix F.

User Background

Table 4-1 shows the programs and systems that responded to the March 1987 WVS questionnaire (as of 1 September 1987). The wide variety of computer systems represented (also seen in Table 1-1) reemphasizes the need for standardization in data bases such as WVS. If more Navy computers can easily incorporate the data base, the data base will be that much more useful to the Navy. For example, users preferred ASCII to binary for the distributed WVS data base, even though binary data takes up less storage space than ASCII. Different computers store binary data differently, but ASCII is a standard data storage format used by most modern computers and is easily input by all Navy computer systems.

System Constraints: Memory, Data Storage and Access

The most critical system constraints cited were the following: size of system and display memory, disk storage, and display generation time. One respondent also mentioned array dimensioning (for storing segment vertices) as a potential problem. Array dimension limitations could impact the maximum number of vertices that can be stored for a single WVS segment.

The method of data storage used by WVS (i.e., 1-degree cells) reduces the chance that a segment array will become too large, since segments are split into smaller units along cell boundaries. Constraints on system and display memory and disk storage are also minimized by the use of data cells: if only certain cells are required for a particular application, then only those cells need to be copied from the distribution tape onto disk and displayed. If the whole world is needed, filtering should be done to reduce the size of the data set.

All respondents liked the 1-degree cell method of data storage for another reason as well: they found it very valuable in speeding up data access, and therefore display generation time, particularly in geographic windowing applications.

Table 4-1. List of users who responded to 1987 WVS questionnaire.

System and Subsystems	Computer Model(s)	Bits/Word
TOMAHAWK: *		
TWCS	ROLM-1666	16
TMPS	VAX 11/780, mVAX	32
TEPEE	HP-9020	32
CSS	not given	
STT at the Space and Naval Warfare System (SPAWAR)	ROLM-1666, HP-9350	16, 32
OPINTEL at the Naval Ocean Systems Center (NOSC)	VAX 11/780, SUN 3/160, IBM-PC/2-80	all 32
P3-UPDATE at the Naval Air Development Center (NADC)	CP-901, UYK-14	8, 32
AEGIS at the Naval Surface Weapons Center (NSWC)	CDC-865, IBM-PC, VAX Cluster, UYK-7, UYK-43B	60, 16, all 32
OTH at the Naval Underwater Systems Center (NUSC)	VAX 11/785	32

*Includes McDonnell Douglas Astronautics Company, Applied Physics Laboratory, Johns Hopkins University, and Lockheed Austin Division.

Data access can also be accelerated by directly accessing WVS data with a cell table look-up file, as described in this report and in the WVS users' guide. Both NOSC and NUSC used this direct access method successfully, liked it, and went on to adapt it to other functions such as feature extraction. Other users were planning to try the direct access method in the future. Meanwhile, they found the sequential access method easy to implement.

Overall, automated "one-pass" data entry was successful. Users cited record counts, feature/segment counts, and other file and cell header information, as well as thorough documentation, as contributing significantly to the ease of data entry.

Due to data storage limitations, most users said they would like DMA to provide the WVS data set at more than one data density (resolution). Systems with severe data storage limitations would be unable to store the full data set on-line. Providing users with several WVS resolutions would also ensure a higher degree of compatibility between systems that will filter their data, since they could start filtering at the data density closest to their need (instead of always starting from the maximum WVS data density). All respondents currently use or plan to use a critical-point data filter.

WVS Applications

All programs plan to use WVS as a map base on which to overlay other data. DMA's Digital Terrain Elevation Data (DTED) and Digital Feature Analysis Data (DFAD) were listed most frequently as the data to be overlayed on WVS. Other data types mentioned include bathymetric contours, Hydrographic Obstruction Data (HOD), cartographic data, and miscellaneous others.

User responses to questions of scale varied widely and were often confused. Some described scale as the number of nautical miles to be displayed without including the size of the output map; others specified the pixel resolution of a monitor without including the range of the map to be displayed. From the few coherent definitions of scale requirements came the following: WVS is needed in scales ranging from 1:7,300 (or about 0.1 nmi/inch) to nearly 1:33,000,000 (or about 450 nmi/inch). Obviously, this range is too large to be handled by a single data density (WVS is currently provided at a scale of about 1:250,000). As mentioned earlier, users would find it helpful if DMA provided several data densities from which to filter to the resolution and scale required. In any event, it should be noted that a scale of 1:7,300 is currently unreasonable for a global shoreline data set.

All respondents currently use or plan to use color fills on the WVS data. Common color fill applications include blocking out water areas to emphasize land features, or vice versa, and shading certain countries to point out various political, geographic or other relationships. For example, water areas may be colored blue; hostile nations may be red. WVS provides a "flag" in each cell header indicating whether that cell contains all land, all water, or a combination (i.e., a shoreline). Users indicated that these flags would be very useful for color land and water fills.

Similarly, WVS provides a flag to indicate what country lies to the left and to the right of each political boundary. None of the respondents needed this information currently, but several foresaw future applications for it. For example, the boundaries of France can easily be picked out of the data set, since each of its boundary segments is marked with a "France" flag.

WVS Documentation

All users liked the documentation and found it clear and easy to follow. The READWVS program that was included (also listed in Appendix E of this report) proved to be very helpful in the initial data input. There was one apparent flaw in the documentation however: some users felt that the explanation of the "chain-node" data structure was inadequate. This report explains the chain-node concept more clearly than the prototype WVS documentation, and the next WVS edition should provide more examples and discussion of chain-node.

A Comment on Data Quality

Most respondents did not have enough time to assess data quality for the prototype WVS before answering the questionnaires. However, NOSC respondents did find "numerous gaps" in political boundary data. This indicates a need to further investigate the WVS data quality before the final data set is distributed to Navy users.

Future WVS Requirements

Asking users to identify future WVS requirements resulted in many good suggestions. For example:

- WVS should be made available in several resolutions (as mentioned earlier).
- A computer program to aggregate cells for smaller scaled maps should be included in the documentation.
- Feature levels (e.g., an island within a lake within a continent) should be identified.
- Features such as islands should be ranked by size, similar to the ranking used in WDB II.
- Nodes (segment endpoints) should be stored explicitly, and should include the points at which different features meet (e.g., where a political boundary meets a shoreline).
- Additional feature types are needed, including navigable rivers, major cities, lakes, airfields, major roads, railroads, ports; there should be a distinction between river mouths and shorelines.
- Software to translate WVS to WDB II should be developed.

• Instructions for converting programs from WDB II to WVS input should be provided.

• Data to the north and south of 65° latitude are needed.

• Links between segments broken at cell boundaries are needed.

5. Summary

DMA and NORDA combined forces to design a data structure and format for the WVS that would best meet users' needs. The new prototype WVS data set meets Navy requirements as listed in the 1985 CNO OP-096 memo, ser 006C/50322906, and as summarized in Table 1-2 of this report. Potential Navy users were given a copy of the new WVS prototype to evaluate, along with a questionnaire that addressed such issues as computer system limitations, WVS applications, and future WVS requirements. Users responded very favorably to the new WVS prototype, and offered many suggestions for future work. The most criticai needs are listed:

1. The WVS data should be checked (and corrected if necessary) for data quality: gaps have been noted in the political boundaries.

2. Chain-node documentation should be expanded in future WVS editions.

3. Several resolutions of WVS are needed, since 1:250,000 is too dense for some applications and many users have data storage limitations that prevent them from utilizing the full resolution. If more than one data set cannot be distributed, then standard filtering techniques and algorithms should be provided to users.

4. Pointers are needed in the WVS file to link segments that span more than one cell (but are broken at the cell boundaries). An additional feature attribute record would be sufficient to store this information.

5. Nodes (segment endpoints) are needed where different features meet (e.g., where a political boundary meets a shoreline).

6. Feature levels should be identified (e.g., an island within a lake within a continent) and features ranked by size (e.g., island features ranked from smallest to largest), similar to the ranking provided by WDB II.

7. Some users need data in the far northern and southern latitudes (currently used shoreline data bases are scarce above 65°N and below 65°S).

6. References

Central Intelligence Agency (1977). *WDB II General Users Guide*. PB 271 869, Washington, DC, July.

Defense Mapping Agency (1985). *Standard Linear Format for Digital Cartographic Feature Data*. Second Edition, Washington, DC, 18 March.

Defense Mapping Agency (1985). *Feature Analysis Coding Standard*. Appendix VII I, First Edition, Washington, DC, 30 July.

International Hydrographic Organization, Committee on the Exchange of Digital Data (1986). *Format for the Exchange of Digital Hydrographic Data*. November.

Langran, Gail, Maura Connor, and R. Kent Clark (1986). *Recommendations on DMA's Standard Linear Format*. Naval Ocean Research and Development Activity, NSTL, MS, NORDA Report 146, May.

Langran, Gail and Maura Connor (1986). *Recommendations on the World Vector Shoreline*. Naval Ocean Research and Development Activity, NSTL, MS, NORDA Report 154, July.

Lohrenz, Maura Connor (1987). *WVS Format for Defense Mapping Agency's World Vector Shoreline Database*. Prototype and First Edition, Naval Ocean Research and Development Activity, NSTL, MS, 6 March.

Appendix A: Feature Attribute Coding Standard (FACS) Codes

FEATURE ATTRIBUTE CODING STANDARD (FACS) CODES
USED IN WVS EDITION ONE

The following list defines the FACS codes used in WVS Edition 1. For a complete FACS list, refer to the Feature Attribute Coding Standard documentation (Defense Mapping Agency, 1985).

FACS code	FACS2 ext.	Description of feature
2A010		Coastal Shoreline
6A000	23	International Boundary (Recognized)
6A000	38	International Boundary (Disputed)
6A000	39	International Boundary (Indefinite)
6A020		Armistice Line
6A030		Ceasefire Line
6A050		Treaty/Convention Line
6A060		DeFacto Boundary
6A070		Demilitarized Zone
6A170		Occupied Area
6A180		Neutral Zone
6C080		Limit of Sovereignty
9A010		Position of text (for country name annotation)

Appendix B: Sample WVS File Listing

SAMPLE WVS FILE LISTING

```
MED_PROTOTYPE         1 0 USDMAHTC     0 10000     10
-1800000-900000360 -100000 250000  420000 500000      -- File Header Records
   8864        32    8832     .    0      8   1   1
   8864  3881              0     1     1   1300     0
CL0 41571 -100000 250000     1     1        2       8   -- First Cell Header
FEA        1L6A000                      00MRWI   1   0 1   -- Boundary feature (FACS = 6A000)
       1F        0        0        0        0        0
SEG         1    27 1 0    7        1C        0        0
      0 35914   1130 35968   2351 35968   3568 35968
   4730 35968   5947 35968   7110 35968   8330 35968
   9490 35968  10710 35968  11930 35968  13090 35968   -- Vertices in segment define
  14310 35968  15469 35968  16690 35968  17849 35968        the boundary
  19069 35968  20228 35968  21449 35968  22669 35968
  23828 35968  25049 35968  26208 35968  27428 35968
  28649 35968  29808 35968  30542 36000       0       0
CL0 41572   -90000 250000    0     0        0       0   -- Empty land cells
CL0 41573   -80000 250000    0     0        0       0
              .         .         .         .
              .         .         .         .
              .         .         .         .
CL0 41614   330000 250000    0     0        0       0
CC0 41615   340000 250000    8     8       16     252
FEA        1L2A010                      0001WI   1   0 1   -- First cell with shoreline data
       1F        0        0        0        0        0
              .         .         .         .
              .         .         .         .             -- 8 shoreline features in
              .         .         .         .                  this cell
FEA        8L2A010                      0001WI   1   0 1
       8F        0        0        0        0        0
SEG         1   868 1 0   217        1C        0        0
  12225 36000  12285 35940  12285 35910  12315 35880
  12315 35850  12345 35820  12345 35760  12375 35730
              .         .         .         .             -- 8 segments define the shorelines
              .         .         .         .
              .         .         .         .
SEG         8    21 1 0     6        8C        0        0
  33555  2700  33555  2760  33600  2805  33630  2805
  33660  2775  33690  2775  33825  2640  33825  2610
  33780  2565  33750  2565  33735  2580  33735  2610
  33720  2625  33690  2625  33660  2655  33630  2625
  33615  2640  33615  2670  33600  2685  33570  2685
  33555  2700      0      0      0      0      0      0
CW0 41616   350000 250000    0     0        0       0   -- Empty water cell
              .         .         .         .
              .         .         .         .
              .         .         .         .
CC0 44112   110000 320000    2     2        4      27
FEA        1L6A000                      00TSLY   1   0 1   -- Boundary and shoreline features
       1F        0        0        0        0        0        stored in the same cell
FEA        2L2A010                      0001LY   1   0 1
       2F        0        0        0        0        0
SEG         1    83 1 0    21        1C        0        0
      0  6854    504  6959   1022  7063   1480  7294
   1969  7607   2488  7945   3006  8046   3467  8230
              .         .         .         .
              .         .         .         .
              .         .         .         .
```

```
 18050 33466 18230 33880 18353 34315 18508 34805
 18688 35240 18871 35629 18925 36000     0     0
SEG      2    16 1 0    4      2C       0     0
 35505 36000 35535 35970 35535 35940 35610 35865
 35640 35865 35670 35835 35700 35835 35730 35805
 35760 35805 35790 35775 35850 35775 35880 35745
 35910 35745 35940 35715 35970 35715 36000 35685
CC0 44113   120000 320000     1     1     2   141
            .           .           .          .
            .           .           .          .
            .           .           .          .
```

-- Total of 1300 cells define the prototype file (Mediterranean)

Appendix C: Determining Cell Codes for Extracting Windows of Data

DETERMINING CELL CODES FOR EXTRACTING WINDOWS OF DATA

The number of the cell in which any point is located may be determined by the algorithm shown in Equation (1):

```
Eq.(1)   CELL_CODE = 1 + ( INT(ABS(Xn - Xo) / XCELL) )
                       + ( INT(ABS(Yn - Yo) / YCELL) * (XMAP / XCELL) )
```

where:
- INT indicates truncation to integer
- ABS indicates absolute value (both are functions in the standard Fortran 77 library)
- CELL_CODE number of cell to find
- Xn = longitude of the point in question
- Yn = latitude of the point
- Xo = longitude of the map origin (LONGOR, in File Header #2)
- Yo = latitude of the map origin (LATOR, in File Header #2)
- XCELL = width of cell in degrees longitude (in File Header #4)
- YCELL = height of cell in degrees latitude (in File Header #4)
- XMAP = width of the map in degrees longitude (in File Header #2)

For example, in the full global WVS shoreline file, the map origin is -180°, -90° (lower left corner of a "rolled out" world map split on the $+/-180^{\circ}$ meridian), cell size is 1 (= XCELL = YCELL), and XMAP is 360. Figure C-1 illustrates this concept with a world map of 30-degree cells (for simplicity).

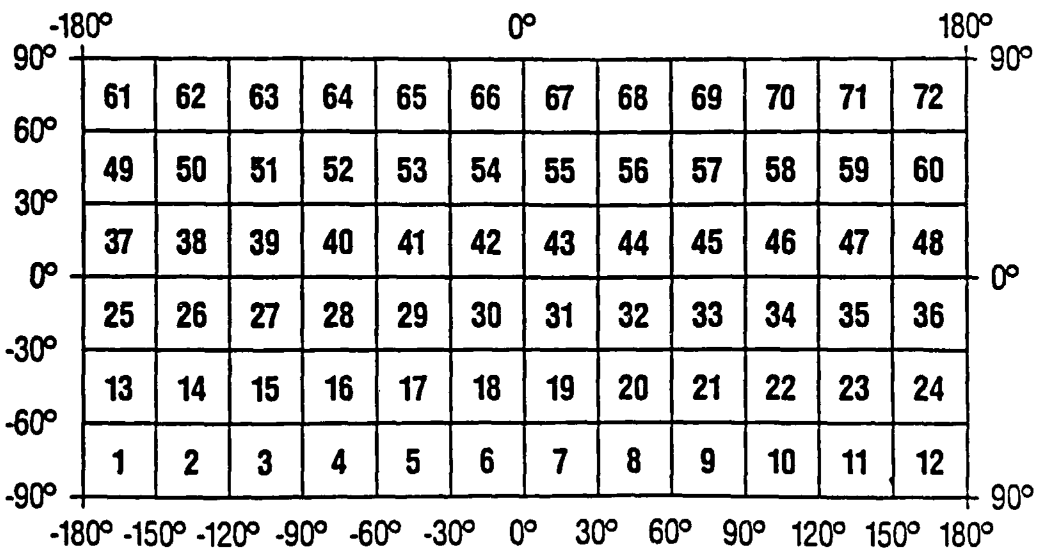


Figure C-1. Diagram of a global map, without data, divided into 30-degree cells. Note that the actual WVS map would be divided into 1-degree cells, numbered from 1 to 64,800.

To find the cell code of a point whose coordinates are (-152.4°Long, 30.2°Lat), proceed as follows from Equation (1):

```
CELL_CODE  = 1 + INT(ABS(-152.4 - (-180)))  + (360 * (INT(ABS(30.2 -(-90)))))
           = 1 + INT(27.6) + (360 * (INT(120.2)))
           = 1 + 27 + (360 * 120)
           = 43228
```

You now know that point (-152.4, 30.2) is in cell #43228. From this basic step, you can window in on a particular area by getting the cell numbers for the upper and lower LONG, LAT boundaries and filling in from there. For example: find the cells needed to work with the following area:

```
        45.2  +-------------------+
              |C                 D|
              |                   |
    Latitude  |                   |
              |                   |
              |A                 B|
        42.4  +-------------------+
              -4.0  Longitude   -1.2
```

First, using Equation (1), find the cell numbers for the corner points:

A: (-4.0, 42.4) is in cell # 47697 = CellA
B: (-1.2, 42.4) is in cell # 47699 = CellB
C: (-4.0, 45.2) is in cell # 48777 = CellC
D: (-1.2, 45.2) is in cell # 48779 = CellD

Since cells wrap the globe sequentially in the X (longitude) direction, you know that the area is 3 cells wide: 1 + (47699 - 47697)
and 4 cells high: 1 + ((48777 - 47697) / 360)

and all the cells you need are as follows:
(CellA .. CellB), (CellA+360 .. CellB+360), .. , (CellC .. CellD)
or, in this case, cells

48777 48778 48779
48417 48418 48419
48057 48058 48059
47697 47698 47699

These cell numbers can easily be calculated in a FORTRAN function, such as the following example:

```
      INTEGER FUNCTION GETCELL(FLONGOR, FLATOR, IXCELL, IYCELL,
     +                               XMAP, FLONGPT, FLATPT)
C
      XCELL = REAL(IXCELL)
      YCELL = REAL(IYCELL)
      GETCELL = 1 +  INT(ABS(FLONGPT - FLONGOR) / XCELL)
     +            + ( (INT(ABS(FLATPT - FLATOR) / YCELL)) * (XMAP / XCELL) )
      RETURN
      END
```

Appendix D: Detailed Description of WVS Format

I. FILE HEADER RECORDS

A. FILE HEADER #1: **Title, source, and coordinate conversions.**

Name	Format	Description
TITLE	A20	Title of this file (names are recorded without spaces).
FILENUM	I1	File number (refer to Table D-1).
EDITION	I2,1X	Edition number (= 0 in prototype and 1st ed.)
PRODUCER	A8,1X	USDMAHTC
COMPDATE	I4,1X	Compilation date (YYMM) of this data set (0 in prototype and 1st ed.)
ORIGDEC	I5	Inverse of implied decimal for LNG, LAT origin points given in file and cell headers. If ORIGDEC = 10,000, divide LNG and LAT origins by 10,000.
DATADEC	I5	Inverse of implied decimal for LNG, LAT data points in segment records. If DATADEC = 10, divide LNG and LAT data points by 10.

Table D-1. *WVS filesets to be distributed by DMA. If space permits, each fileset will be distributed on one magnetic 6250 bpi tape.*

File #	Ocean area
0.	Southern Ocean
1.	Eastern North Atlantic
2.	Indian Ocean
3.	Western North Pacific
4.	Eastern North Pacific
5.	South Pacific
6.	Western North Atlantic
7.	Eastern South Atlantic
8.	Western South Atlantic
9.	Arctic

B. FILE HEADER #2: Geographic boundaries of file.

Name	Format	Description
LNGOR	I8	Longitude of map origin (global origin: -180)*
LATOR	I7	Latitude of map origin (global origin: -90)*
XMAP	I3	Width of file in degrees longitude (for full world: 360)*
LNGSW	I8	Longitude of southwest file corner (in prototype: -10)*
LATSW	I7	Latitude of southwest file corner (in prototype: 25)*
LNGNE	I8	Longitude of northeast file corner (in prototype: 42)*
LATNE	I7	Latitude of northeast file corner (in prototype: 50)*

* Values stored as degrees x ORIGDEC.

C. FILE HEADER #3: Feature-specific counts.

Name	Format	Description
NFEA	I7,1X	Total number of features in file.
NPOINTS	I7,1X	Number of point features.
NLINES	I7,1X	Number of line features.
NAREAS	I7,1X	Number of area features.
NCODES	I4,1X	Number of FACS codes in file (e.g., # feature types).
MSEGperFEA	I3,1X	Maximum number of segments comprising a feature.
MFEAperSEG	I2	Maximum number of features depicted by a segment.
BLANK	5X	

D. FILE HEADER #4: Segment-specific counts, scale, and text record counts.

Name	Format	Description
NSEG	I7,1X	Total number of segments in file.
MVRTperSEG	I5,1X	Maximum number of vertices in a segment.
ISCALE	I9,1X	Scale reciprocal (e.g., 250000 = 1:250,000).
XCELL	I4,1X	Width (X) of cells in degrees LNG (implied dec 0.1). 0010 for WVS Edition One.
YCELL	I4,1X	Height (Y) of cells in degrees LAT (implied dec 0.1). 0010 for WVS Edition One.
NCELLS	I6,1X	Number of cells in this file.
NTEXT	I4	Number of text records to follow.
BLANK	3X	

E. TEXT RECORDS: Format = A48. The number of text records is NTEXT.

II. DATA CELL RECORDS

Each of the NCELLS contained in the file is represented by a cell header. If a cell is empty (NSEGinCELL = 0), there are no data records following its header. If a cell is not empty (NSEGinCELL > 0), the header is followed by the cell's feature records, which are followed by a series of segment records (refer back to Table 2-2). The formats of the cell header feature records, and segment records follow.

A. CELL HEADER

Name	Format	Description
CELLFLAG	A1	New cell marker = "C" (constant).
CELLTYPE	A1	Key to cell contents: "L" = all land, "W" = all water, "C" = combination.
NCELLHEAD	I1	Number of additional cell header records that immediately follow this one (for WVS Edition One, NCELLHEAD = 0).
CELLNUM	I6	Cell number (1-64800).
LNGCELL	I8	Longitude of lower left cell corner. *
LATCELL	I7	Latitude of lower left cell corner. *
NFEAinCELL	I5	Number of features in cell.
NSEGinCELL	I5	Number of segments in cell.
NFEAREC	I7	Number of 48-byte feature records in cell.
NSEGREC	I7	Number of 48-byte segment records in cell. Note: NFEAREC + NSEGREC + NCELLHEAD - 1 = number of records preceding the next cell.

* Cell origins stored as degrees; divide by ORIGDEC (from FILE HEADER #1).

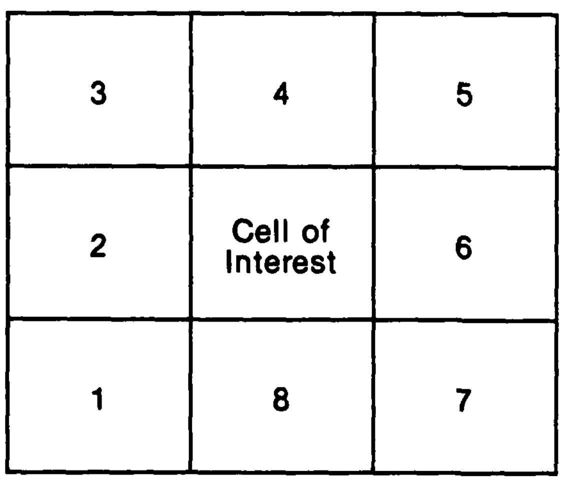


Figure D-1. *Edge codes for continuation of line segments beyond cell boundaries (CONT1, CONT2 in Feature Header).*

B. FEATURE RECORDS

Each cell contains NFEAinCELL features. Each feature has a header record, extra attribute records (if required by the feature type), and data records that contain references to the feature's component segments.

1) **FEATURE HEADER.** Each feature has at least one header record to identify the feature and its associated attributes, segment records, and (feature) data records. If NATTR > 0, additional 48-byte attribute records immediately follow the feature header.

Name	Format	Description
FEAFLAG	A3	New feature marker = "FEA" (constant).
FEANUM	I7	Feature identification number (unique in current cell) used to key from segment records.
FEATYPE	A1	"P" point, "L" line, "A" area feature.
FACS	A5	Code indicating feature type (see Appendix A).
ATTRIB	A24	Attribute buffer; format based on feature type. o Shorelines and boundaries: ATTVAL 3A6 Up to three attributes, each with a three-letter attribute code and three digit value code. CONT1 I1 Edge code where feature enters cell from adjacent cell. For features that begin within the cell, CONT1 = 0 (see Figure D-1). CONT2 I1 Edge code where feature exits cell to adjacent cell. For features that terminate within the cell, CONT2 = 0 (see Figure D-1). FEALAB A4 2-character country code to left and right of feature, respectively. If feature is a shoreline, then the water side will be represented by the file number instead of a country code. o Countries names: CCODE A2 2-character country code. CNAME A20 Country name. SCODE A2 2-character sovereignty code.
NSEGinFEA	I3	Number of segments comprising this feature.
NATTR	I3	Number of extra attribute records that immediately follow this header record (i.e., following the ATTRIB buffer). For shorelines and boundaries, NATTR = 0; for country names, NATTR = 1.
NFDATAREC	I2	Number of feature data records that follow any extra attribute records, after this header record.

2) **EXTRA ATTRIBUTE RECORD(S).** For WVS Edition One, only countries will use additional attribute records. They will describe the country's extent and the position of the country's name (approximate center). Future WVS editions may have more complex and varied attribute requirements, in which case the NATTR variable will state how many extra attribute records are allocated for the feature. Each attribute record is 48 bytes LNG. The following is an example of the country attribute format.

Name	Format	Description
MINLNG	I8	Minimum longitude (first integer meridian west of the country). *
MINLAT	I7	Minimum latitude (first integer parallel south of the country). *
MAXLNG	I8	Maximum longitude (first integer meridian east of the country). *
MAXLAT	I7	Maximum latitude (first integer parallel north of the country). *
CENTLNG	I8	Approximate center longitude. *
CENTLAT	I7	Approximate center latitude. *
BLANK	3X	

* Coordinates stored as degrees x ORIGDEC (from FILE HEADER #1).

3) **FEATURE DATA RECORDS.** All component segments of the current feature are listed here. Segments are identified by unique numeric keys (SEGNUM); the ordering of points is provided (SEGDIR). Up to six pairs of (SEGNUM, SEGDIR) can be stored in a feature data record. As described in Feature Header Record, total number of pairs = NSEGinFEA; total number of feature data records = NFDATAREC.

Name	Format	Description
SEGNUM	I7	Segment identification number (unique in current cell) used to key to a segment record in this cell.
SEGDIR	A1	Direction of points in segment: "F" = Forward, or same order as stored "R" = Reverse of stored order "D" = Disjointed, same order as stored "E" = Disjointed, reverse order "I" = Inside, same order as stored "J" = Inside, reverse order (see Table 2-3 for more detailed explanation).

C. SEGMENT RECORDS

Each cell contains NSEGinCELL segments. Each segment has a Header, optional Extra Feature Reference Records that key to features using the current segment, and Data Records that contain the segment vertices.

1) **SEGMENT HEADER.** All features delineated by the segment are referenced here. Features are referenced by unique numeric keys (FEANUM) and feature orientation (FEAORI). Up to 3 pairs of (FEANUM, FEAORI) fit in a single Segment Header Record. If NFEAinSEG > 3, the overflow feature references are stored in additional 48-byte Feature Reference Records.

Name	Format	Description
SEGFLAG	A3	New segment marker = "SEG".
SEGNUM	I7	Segment key: segment identification number (unique in current cell).
NVERTS	I5	Number of vertices (x,y data points) in segment.
NFEAinSEG	I2	Number of features to which this segment belongs.
NXFEARECS	I2	Number of extra feature reference records immediately following this header record.
NSDATAREC	I5	Number of segment data records following any extra feature ref. records after this header record.
FEANUM	I7	Feature key: a number identifying a feature to which this segment belongs.
FEAORI	A1	Feature orientation: "L" = feature is left of line segment, "R" = feature is right of line segment, "C" = feature coincident with line segment. (Point features are coincident "C" with segments).
FEANUM	I7	
FEAORI	A1	
FEANUM	I7	
FEAORI	A1	

2) **EXTRA FEATURE REFERENCE RECORDS.** The number of extra feature reference records used is NXFEARECS. The format is a repeating sequence of up to six pairs of (FEANUM, FEAORI), continued from the SEGMENT HEADER RECORD.

3) **SEGMENT DATA RECORDS.** A listing of NVERTS segment vertices, with coordinates expressed in longitude, latitude (XSECONDS, YSECONDS) pairs, up to four pairs per data record. Total number of data records is NSDATAREC, listed in the Segment Header Record. Coordinates are stored as seconds of offset from the current cell origin, with an implied decimal point of DATADEC. To transform the vertices to global coordinates:

longitude = (LNGCELL / ORIGDEC) + [(XSECONDS / DATADEC) / 3600]
latitude = (LATCELL / ORIGDEC) + [(YSECONDS / DATADEC) / 3600]

Name	Format	Description
XSECONDS	I6	Longitude in seconds of offset from cell origin.
YSECONDS	I6	Latitude in seconds of offset from cell origin.

Appendix E: Fortran Program to Read WVS File

SAMPLE FORTRAN PROGRAM TO READ IN WVS FILE

The following five pages list a Fortran-77 program and associated subroutines designed to be a guide to reading a WVS file. The WVS file is assumed to be on disk, with fixed 48-byte records. If the file is still on tape, the user may have to edit this program to incorporate certain tape reading commands specific to the computer being used. An include file ('WVSFORM.INC') is listed on the last 2 pages of this Appendix; it is used to declare all WVS variables needed by the program. Note the introductory comments in WVSFORM.INC: they pertain to changes that will need to be made to the include file for future WVS releases.

```
      PROGRAM READWVS
C
C     Reads in entire WVS file and makes a cell table look-up file.
C     Table look-up file is direct access, with a fixed record length
C     of 4 bytes (1 word).
C     This program is only meant as a guide, and should be adapted
C     to user's individual applications.
C
C     Programmer: Maura C. Lohrenz
C     Computer:   VAX 11/780
C     Language:   FORTRAN 77
C     Date:       13 February 1987
C
C
      CHARACTER*48 BUFF48         ! Generic 48-byte WVS buffer
      CHARACTER*24 FEABUFF        ! Buffer to hold FEANUM,FEAORI fields
C                                 !   in Segment Header records
      INTEGER*4 NRECS             ! Record Count, output to ITABLE as
C                                 !    each Cell Header is reached
      INCLUDE 'WVSFORM.INC'       ! Include file of WVS data types
      DATA INWVS  /11/,           ! Input WVS file
     +      ITABLE /12/           ! Output cell table-lookup file
C
C
      PRINT *,' Program READWVS, version of 4 May, 1988'
C
C  *** Format statements for each of the 12 WVS record types ***
C
 1001 FORMAT (A20,I1,I2,1X,A8,1X,I4,1X,2I5)                 ! File Header #1
 1002 FORMAT (I8,I7,I3,2(I8,I7))                            ! File Header #2
 1003 FORMAT (4(I7,1X),I4,1X,I3,1X,I2,5X)                   ! File Header #3
 1004 FORMAT (I7,1X,I5,1X,I9,1X,2(I4,1X),I6,1X,I4,3X)       ! File Header #4
 1005 FORMAT (A48)                                          ! Text Records
 1006 FORMAT (2A1,I1,I6,I8,I7,2I5,2I7)                      ! Cell Header
 1007 FORMAT (A3,I7,A1,A5,A24,2I3,I2)                       ! Feature Header
C1008        [ Not used in WVS prototype ]                  ! Extra Fea.Attributes
 1009 FORMAT (6(I7,A1))                                     ! Feature Data
 1010 FORMAT (A3,I7,I5,2I2,I5,A24)                          ! Segment Header
 1011 FORMAT (6(I7,A1))                                     ! Extra Feature Ref.s
 1012 FORMAT (8I6)                                          ! Segment Data
C
      NRECS = 0                   ! Initialize record count
C
C  *** Format statements for output ***
C
 1101 FORMAT(' WVS file: ',A20,', File # ',I2,', Ed.# ',I3,
     +          ', produced by ',A8,'.')
 1102 FORMAT(' Map origin      = ',F10.4,' LNG, ',F10.4,' Lat.',/,
     +          ' Map width       = ',I4,' LNG degrees.',/,
     +          ' SW file corner = ',F10.4,' LNG, ',F10.4,' Lat.',/,
     +          ' NE file corner = ',F10.4,' LNG, ',F10.4,' Lat.',/)
```

```
 1103     FORMAT(' Cell width, height = ',I4,',',I4,',',/,
     +           ' Number of cells = ',I6,',',/)
 1104     FORMAT(' Number of features = ',I7,', including: ',
     +           /,I7,' point',I7,' line, and',I7,' area features.',
     +           /, ' Total no. FACS types = ',I4,',',/)
 1105     FORMAT(' Max.# of segments per feature = ',I3,',',/,
     +           ' Max.# of features per segment = ',I2,',',/)
 1106     FORMAT(' Number of segments = ',I7,',',/,
     +           ' Max.# of vertices per segment = ',I5,',',/,
     +           ' Intended scale = 1:',I9,',',/)
C
C    *** Open files ***
C
          OPEN (INWVS,STATUS='OLD',ACCESS='SEQUENTIAL',
     +                  RECORDTYPE='FIXED',RECL=48)
C
          OPEN (ITABLE,STATUS='NEW',ACCESS='DIRECT',
     +                  RECORDTYPE='FIXED',RECL=1)
C
C    *** Read first 4 file header records ***
C
          READ (INWVS,1001) TITLE,FILENUM,EDITION,PRODUCER,
     +                      COMPDATE,ORIGDEC,DATADEC
          NRECS = NRECS + 1
          READ (INWVS,1002) LNGOR,LATOR,XMAP,
     +                      LNGSW,LATSW,LNGNE,LATNE
          NRECS = NRECS + 1
          READ (INWVS,1003) NFEA,NPOINTS,NLINES,NAREAS,
     +                      NCODES,MSEGperFEA,MFEAperSEG
          NRECS = NRECS + 1
          READ (INWVS,1004) NSEG,MVRTperSEG,ISCALE,
     +                      XCELL,YCELL,NCELLS,NTEXT
          NRECS = NRECS + 1
C
C    *** Print out important information from File Header ***
C
          FORIGDEC = FLOAT(ORIGDEC)
          DLNGOR   = FLOAT(LNGOR) / FORIGDEC
          DLATOR   = FLOAT(LATOR) / FORIGDEC
          DLNGSW   = FLOAT(LNGSW) / FORIGDEC
          DLATSW   = FLOAT(LATSW) / FORIGDEC
          DLNGNE   = FLOAT(LNGNE) / FORIGDEC
          DLATNE   = FLOAT(LATNE) / FORIGDEC
C
          WRITE (6,1101) TITLE, FILENUM, EDITION, PRODUCER
          WRITE (6,1102) DLNGOR, DLATOR, XMAP,
     +                   DLNGSW, DLATSW, DLNGNE, DLATNE
          WRITE (6,1103) XCELL, YCELL, NCELLS
          WRITE (6,1104) NFEA, NPOINTS, NLINES, NAREAS, NCODES
          WRITE (6,1105) MSEGperFEA, MFEAperSEG
          WRITE (6,1106) NSEG, MVRTperSEG, ISCALE
C
```

```
C     *** Read and print NTEXT text records ***
C
 1107     FORMAT(1X,A48)
          IF (NTEXT.GT.0) THEN
             WRITE (6,*) ' Text records: '
             ILOC = 1
             DO 100 I=1,NTEXT
                READ (INWVS,1005) TEXT(I)
                WRITE (6,1107) TEXT(I)
                NRECS = NRECS + 1
  100        CONTINUE
          END IF
C
C     *** Start main read loop, one cell at a time ***
C
          NC = 0
          NCELLRECS = 0
          DO 200 I=1,NCELLS
C                                                 ! Read Cell Header
             READ (INWVS, 1006) CELLFLAG,CELLTYPE,NCELLHEAD,CELLNUM,
     +                          LNGCELL,LATCELL,NFEAinCELL,NSEGinCELL,
     +                          NFEAREC,NSEGREC
             NC = NC + 1
             DLNGCELL = FLOAT(LNGCELL) / FORIGDEC
             DLATCELL = FLOAT(LATCELL) / FORIGDEC
C
             NRECS = NRECS + 1 + NCELLRECS        ! Get starting rec# of current cell
             ILOC = 2                             !     and output to ITABLE
             WRITE (ITABLE,REC=CELLNUM) NRECS
C
             ! Get #recs in this cell, to find next cell's starting rec#
             NCELLRECS = NCELLHEAD + NFEAREC + NSEGREC
C
             DO 110 J=1,NFEAinCELL
C                                                 ! Read Feature Header
                READ (INWVS,1007) FEAFLAG,FEANUM,FEATYPE,FACS,
     +                            ATTRIB,NSEGinFEA,NATTR,NFDATAREC
                IF (NATTR.GT.0) THEN
                   DO 105 K=1,NATTR               ! Skip if any; not used in prototype
                      READ (INWVS,1005) BUFF48
  105              CONTINUE
                END IF
C                                                 ! Read Feature Data Records
                CALL RFEADATA (INWVS,NFDATAREC,SEGNUM_ARR,SEGDIR_ARR,NSEGinFEA)
C
  110        CONTINUE
C
             DO 120 J=1,NSEGinCELL
C                                                 ! Read Segment Header
                READ (INWVS,1010) SEGFLAG,SEGNUM,NVERTS,NFEAinSEG,
     +                            NXFEARECS,NSDATAREC,FEABUFF
C
                N = 3 + (NXFEARECS*6)             ! Get all feature references
                CALL RXFEAREF (INWVS,FEABUFF,FEANUM_ARR,FEAORI_ARR,
     +                         N,NXFEARECS)
```

```
C                                                 ! Read Segment Data (returns
            N = NSDATAREC * 4                     ! LNG/Lat points in dec.degrees)
C
            CALL RSEGDATA (INWVS,DLNGCELL,DLATCELL,DATADEC,
     +                     DATALNG,DATALAT,N,NSDATAREC)
 120      CONTINUE
 200    CONTINUE
C
 900    PRINT *,' End of program READWVS; number of cells read = ',NC
        CLOSE (INWVS)
        CLOSE (ITABLE)
        STOP
        END
C
C
        SUBROUTINE RFEADATA (INWVS,NFDATAREC,SEGNUM_ARR,SEGDIR_ARR,N)
C       Reads Feature Data Records (ID and direction of
C       all segments making up current feature).
C
        INTEGER  SEGNUM_ARR(N)
        CHARACTER*1  SEGDIR_ARR(N)
 1009   FORMAT ( 6 (I7,A1) )
C
        NUM = 1
        DO 100 I=1,NFDATAREC
          READ (INWVS,1009) ((SEGNUM_ARR(J),SEGDIR_ARR(J)),
     +                       J=NUM,NUM+3)
          NUM = NUM + 4
 100    CONTINUE
        RETURN
        END
C
C
        SUBROUTINE RXFEAREF (INWVS,FEABUFF,FEANUM_ARR,FEAORI_ARR,N,
     +                       NXFEARECS)
C       Reads Feature Reference Data for current segment, including
C       up to 3 in FEABUFF, taken from Segment Header Record.
C
        INTEGER  FEANUM_ARR(N)
        CHARACTER*1  FEAORI_ARR(N)
        CHARACTER*24 FEABUFF
 1000   FORMAT ( 3 (I7,A1) )
 1011   FORMAT ( 6 (I7,A1) )
C
        READ (FEABUFF,1000) ( (FEANUM_ARR(I),FEAORI_ARR(I)) ,I=1,3)
C
        IF (NXFEARECS.GT.0) THEN
           I = 4
           DO 100 NF=1,NXFEARECS
              READ (INWVS,1011) ( (FEANUM_ARR(II),FEAORI_ARR(II)) ,II=I,I+5)
              I = I + 6
 100       CONTINUE
        END IF
        RETURN
        END
```

```
      SUBROUTINE RSEGDATA (INWVS,DLNGCELL,DLATCELL,DATADEC,
     +                     DATALNG,DATALAT,N,NSDATAREC)
C
C     Reads X,Y vertices from Segment Data Records, and converts to
C     geographic coordinates (+/- LNG,Lat in decimal degrees).
C
      INTEGER XSECONDS(4), YSECONDS(4), DATADEC
      REAL    DATALNG(N), DATALAT(N)
 1012 FORMAT (8I6)
C
      FACT = FLOAT(DATADEC) * 3600.
      NUM = 0
      DO 100 I=1,NSDATAREC
         READ (INWVS,1012) ( (XSECONDS(J),YSECONDS(J)) ,J=1,4)
         DO 50 K=1,4
            NUM = NUM + 1
            DATALNG(NUM) = DLNGCELL + (FLOAT (XSECONDS(K)) / FACT)
            DATALAT(NUM) = DLATCELL + (FLOAT (YSECONDS(K)) / FACT)
   50    CONTINUE
  100 CONTINUE
      RETURN
      END
```

```
C     WVSFORM.INC:    Include file for all WVS data types; Used by READWVS.FOR
C                     NOTE: arrays may need re-dimensioning once chain-node is
C                     put into effect. At present, with only 1 feature/segment
C                     and 1 segment/feature, some arrays (SEGNUM_ARR, etc) are
C                     only dimensioned to 1. Also, data points (DATALNG, DATALAT)
C                     are dimensioned to 5000; if the maximum points/segment
C                     exceeds 5000, these must be redimensioned. Finally, no
C                     extra feature attribute fields have been defined here.
C
C     FILE HEADER #1: Title, source, and coordinate conversions.
C
      CHARACTER*20 TITLE          ! (A20)    Title of this file
      INTEGER      FILENUM        ! (I1)     File numbe-
      INTEGER      EDITION        ! (I2,1X)  Edition number
      CHARACTER*8  PRODUCER       ! (A8,1X)  Usually USDMAHTC
      INTEGER      COMPDATE       ! (I4,1X)  Compilation date
      INTEGER      ORIGDEC        ! (I5)     Inverse of implied decimal
                                  !            for LNG, LAT origin points
      INTEGER      DATADEC        ! (I5)     Inverse of implied decimal
                                  !            for LNG, LAT data points
C
C     FILE HEADER #2: Geographic boundaries of file.
C
      INTEGER      LNGOR          ! (I8)   LNG. of file origin in deg.xORIGDEC
      INTEGER      LATOR          ! (I7)   Lat. of file origin in deg.xORIGDEC
      INTEGER      XMAP           ! (I3)   Width of file in whole degrees
      INTEGER      LNGSW          ! (I8)   LNG. of Southwest map corner xORIGDEC
      INTEGER      LATSW          ! (I7)   Lat. of Southwest map corner    "
      INTEGER      LNGNE          ! (I8)   LNG. of Northeast map corner    "
      INTEGER      LATNE          ! (I7)   Lat. of Northeast map corner    "
C
C     FILE HEADER #3: Feature-specific counts.
C
      INTEGER      NFEA           ! (I7,1X) Total number of features in file
      INTEGER      NPOINTS        ! (I7,1X) Number of point features
      INTEGER      NLINES         ! (I7,1X) Number of line features
      INTEGER      NAREAS         ! (I7,1X) Number of area features
      INTEGER      NCODES         ! (I4,1X) Number of FACS codes in file
      INTEGER      MSEGperFEA     ! (I3,1X) Max.# segments comprising a feature
      INTEGER      MFEAperSEG     ! (I2,5X) Max.# features depicted by a segment
C
C     FILE HEADER #4: Segment-specific counts, scale, and text record counts.
C
      INTEGER      NSEG           ! (I7,1X) Total number of segments in file
      INTEGER      MVRTperSEG     ! (I5,1X) Max.# vertices in a segment
      INTEGER      ISCALE         ! (I9,1X) Scale reciprocal (250000 = 1:250,000)
      INTEGER      XCELL          ! (I4,1X) Width (X) of cells in deg.LNG x10
      INTEGER      YCELL          ! (I4,1X) Height (Y) of cells in deg.lat x10
      INTEGER      NCELLS         ! (I6,1X) Number of cells in this file
      INTEGER      NTEXT          ! (I4,3X) Number of text records to follow
C
C     TEXT RECORDS:
C
      CHARACTER*48 TEXT(1)        ! (A48) Array of text. # text records = NTEXT
```

```
C    CELL HEADER
C
       CHARACTER*1    CELLFLAG         ! (A1) New cell marker = "C"
       CHARACTER*1    CELLTYPE         ! (A1) Key to cell contents (L, W, or C)
       INTEGER        NCELLHEAD        ! (I1) #additional cell header records after this
       INTEGER        CELLNUM          ! (I6) Cell number (1-64800)
       INTEGER        LNGCELL          ! (I8) Low.left cell corner in deg. LNG x ORIGDEC
       INTEGER        LATCELL          ! (I7) Low.left cell corner in deg. lat x ORIGDEC
       INTEGER        NFEAinCELL       ! (I5) Number of features in cell
       INTEGER        NSEGinCELL       ! (I5) Number of segments in cell
       INTEGER        NFEAREC          ! (I7) # of 48-byte feature records in cell
       INTEGER        NSEGREC          ! (I7) # of 48-byte segment records in cell
C
C   FEATURE HEADER
C
       CHARACTER*3    FEAFLAG          ! (A3)  New feature marker = "FEA"
       INTEGER        FEANUM           ! (I7)  Unique feature ID
       CHARACTER*1    FEATYPE          ! (A1)  "P" point, "L" line, "A" area feature
       CHARACTER*5    FACS             ! (A5)  Code indicating feature type
       CHARACTER*22   ATTRIB           ! (A24)Feature-dependant Attribute buffer
       INTEGER        NSEGinFEA        ! (I3)  # segments comprising this feature
       INTEGER        NATTR            ! (I2)  # extra attribute records to follow
       INTEGER        NFDATAREC        ! (I2)  # feature data records
C
C   FEATURE DATA RECORDS: info about segments making up this feature
C
       INTEGER        SEGNUM_ARR(1)    ! (I7)  Array of Segment IDs
       CHARACTER*1    SEGDIR_ARR(1)    ! (A1)  Array of Segment Directions
       CHARACTER*1    SEGDIR           ! (A1)  Single segment direction code
C
C   SEGMENT HEADER AND EXTRA FEATURE REFERENCE RECORDS
C
       CHARACTER*3    SEGFLAG          ! (A3)  New segment marker = "SEG"
       INTEGER        SEGNUM           ! (I7)  Segment key
       INTEGER        NVERTS           ! (I5)  Number of vertices in segment
       INTEGER        NFEAinSEG        ! (I2)  # features to which this segment belongs
       INTEGER        NXFEARECS        ! (I2)  # extra feature reference records
       INTEGER        NSDATAREC        ! (I5)  # segment data records
C                                      !  For this segment:
       INTEGER        FEANUM_ARR(1)    ! (I7)  Array of Feature IDs
       CHARACTER*1    FEAORI_ARR(1)    ! (A1)  Array of Feature orientations
       CHARACTER*1    FEAORI           ! (A1)  Single feature orientation code
C
C   SEGMENT DATA RECORDS
C                                   !  XSECONDS, YSECONDS stored 4/record:
       INTEGER        XSECONDS(4)      ! (I6) LNG: sec x DATADEC offset from LNGCELL
       INTEGER        YSECONDS(4)      ! (I6) Lat: sec x DATADEC offset from LATCELL
C
C                                   !  DATALNG, DATALAT stored for a whole segment:
       REAL           DATALNG(5000)    ! (F10.4) LNG. in dec. degrees
       REAL           DATALAT(5000)    ! (F10.4) Lat. in dec. degrees
```

Appendix F: WVS Questionnaire to Users

QUESTIONNAIRE SENT TO POTENTIAL USERS
TO EVALUATE PROTOTYPE WORLD VECTOR SHORELINE

USER BACKGROUND

1. Program or system name:

2. Program sponsor:

3. Name and phone number of user or person answering this questionnaire:

4. Date complete WVS data-set is needed:

5. Would you like to see a WVS users' group formed?

DATA STORAGE AND TRANSPORT

1. List the make, model, and word-size for each computer in your system that will be using WVS:

2. Does your system have memory or storage constraints?
   If so, please describe:

3. What is the limit of array dimensioning on your system?
   Was the maximum number of vertices/segment too large in the WVS file?

4. Comment on the ease or difficulty of the automated data entry of WVS.
   Did you find the feature and segment record counts useful?
   If you required one-pass input, were there any problems?

5. Will WVS data be transmitted over communications links or a network?

WVS FORMAT AND APPLICATIONS

1. Are you displaying WVS as a map?
   If so, at what resolutions or scales (e.g., 60 n.mi./inch ; 1:50,000) ?

2. Do you plan to use generalization or point reduction to decrease the size of the data set or for small scale displays?
   Describe methods and applications anticipated:

3. Please comment on the method of data storage used by WVS (i.e., dividing the world into 1-degree cells).
   Have you found this helpful in windowing or re-partitioning the data?
   Was the documentation satisfactory in describing these methods?

4. Do you intend to perform color-fills of areas (i.e., a land mask) using WVS? Did you find useful the tagging of cells (including "empty cells") as all land, all water, or a combination of land/water?

5. Will you be using the country-left/country-right tags? Were they clear and easy to extract?

6. Do you intend to display WVS with other data types such as DTED, bathymetric contours, etc.? Please describe. If you are already doing so, have you encountered any difficultes?

7. Have you tried sequential and/or direct methods of accessing specific data cells, as described in the documentation? Please comment on usefulness, ease of use, and possible improvements.

8. The WVS data format been designed to accommodate "chain-node" data. That is, the direction in which the digitized points were stored will be evident, and features will be cross-referenced to the segments which define them. This will allow, for example, the automatic identification of land/water sides of a shoreline, countries on either side of a political boundary, etc. Chain-node also reduces the amount of storage required to represent geographic features, since a segment shared by more than one feature is stored only once in the WVS file. (For more description of "chain-node" as used in WVS, see the WVS documentation). The prototype WVS data does not yet contain the information necessary to benefit from chain-node (e.g., the direction of points within a segment, and the orientation of features with respect to their segments, have yet to be encoded correctly in the dataset). Future editions of WVS will contain this information.

   Will your applications require chain-node data? Please list any problems or short-comings you forsee in applying chain-node to the WVS format.

DOCUMENTATION

1. Were there enough examples of format and file access?

2. Was the READWVS program helpful? How long did it take you to write a working program to read in the file?

3. Comment on the consistency and interpretability of variable names in the documentation.

4. Was the documentation clear and understandable, in general? Please point out any weak areas.

DATA QUALITY

1. Have you found any errors or discrepancies in the data itself? Comment if any shorelines were found to cross (they should not), or if points that are on cell boundaries are not duplicated in adjacent cells (they should be duplicated), etc.

FUTURE REQUIREMENTS

1. Do you forsee a need for nodes (segment endpoints) where different features meet (e.g., where a political boundary meets a shoreline)?

2. Would your system need to identify feature levels (e.g., an island w/in a lake w/in a land mass w/in the ocean) or to rank features by size (e.g., island features ranked from 1 - 10 : smallest to largest)?

3. List any additional feature types your system may need in future WVS editions (e.g., navigable rivers, lakes, major roads, cities and placenames, ports, airfields, etc).

4. Does your system currently use the CIA World Data Bank II (WDB II) dataset? If so, would translation software between WDB II and WVS data be useful for your applications?

5. Will you need more than one data density?

6. Are the FACS and feature attribute descriptions thorough enough? Would you like to see more feature description in the format? List for each current feature type:
   Shorelines:
   Boundaries:
   Countries:

7. Would you prefer a binary version of WVS? If so, do you forsee any problems converting foreign binary data to your system (may require some byte/bit manipulation)?

FURTHER COMMENTS: Please attach additional sheets if necessary.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NORDA Report 194	5. MONITORING ORGANIZATION REPORT NUMBER(S) NORDA Report 194
6. NAME OF PERFORMING ORGANIZATION Naval Ocean Research and Development Activity	7a. NAME OF MONITORING ORGANIZATION Naval Ocean Research and Development Activity
6c. ADDRESS (City, State, and ZIP Code) Ocean Science Directorate NSTL, Mississippi 39529-5004	7b. ADDRESS (City, State, and ZIP Code) Ocean Science Directorate NSTL, Mississippi 39529-5004
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Ocean Research and Development Activity — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Ocean Science Directorate NSTL, Mississippi 39529-5004	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. 980101; PROJECT NO. 00101; TASK NO. KKK; WORK UNIT NO. 93518X

11. TITLE (Include Security Classification)
Design and Implementation of the Digital World Vector Shoreline Data Format

12. PERSONAL AUTHOR(S)
Maura Connor Lohrenz

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	From ____ To ____	May 1988	36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	shores; data management; data bases; digital maps; computer applications.

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

This report defines the data structure and format that NORDA designed for the Defense Mapping Agency's (DMA) digital World Vector Shoreline (WVS) data set. Methods of accessing the data set, current and future applications for WVS, and suggested future additions to the data set are also discussed. Applications and suggestions are based on responses to a questionnaire prepared by NORDA to survey DoD users who evaluated the latest WVS prototype, the Mediterranean Sea area.

NORDA's role in designing the WVS data format and structure was a direct result of our recommendations to DMA on their original WVS prototype data set and format, presented in NORDA Reports 146 and 154. In these reports, NORDA recommended that specific changes be made to DMA's original WVS to be sure that it would meet the Navy's requirements for a digital global shoreline map (according to CNO OP-006 memo, ser 006C/50322906, 5 August 1985). In response, DMA requested that NORDA assist them in the design and implementation of a new WVS format and data structure. This new product was completed and distributed to potential Navy and other DoD users in March 1987.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ■ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Maura Connor Lohrenz	22b. TELEPHONE NUMBER (Include Area Code) (601) 688-4611 — 22c. OFFICE SYMBOL Code 351

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE