METRIC

MIL-PRF-32118
20 March 2003

PERFORMANCE SPECIFICATION

VECTOR PRODUCT FORMAT (VPF) DATABASE UPDATE (VDU)

This specification is approved for use by all Departments and Agencies of the Department of Defense

1. SCOPE

1.1 Scope. This specification defines requirements for the National Imagery and Mapping Agency's (NIMA) Vector Product Format (VPF) Database Update (VDU). This specification further defines the tables that are necessary to build an "update layer" for NIMA Vector Product Format (VPF) data sets that require tracking and accountability for changes made to the underlying VPF data set.

1.2 Purpose. The purpose of this specification is to assure uniformity of treatment among all elements engaged in a coordinated production program for generating VPF update layers for those NIMA VPF products that require them.

1.3 Classification. Requirements for updating Vector Product Format (VPF) data sets can be classified as follows:

- No update to VPF data is required (this specification is not applicable in this situation).

- Update of VPF geospatial data is required, but no tracking of changes is required (section 3.2 applies/VDU update layer is not required).

- Update of VPF geospatial data and tracking of changes are required (sections 3.2 and 3.3 both apply/VDU update layer is required).

2. APPLICABLE DOCUMENTS

2.1 General. The documents listed in this section are specified in Sections 3 and 4 of this specification. This section does not include documents cited in other sections of this specification or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in Sections 3 and 4 of this specification whether or not they are listed.

Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: Director, National Imagery and Mapping Agency, ATTN: ETA, Mail Stop P-24, 4600 Sangamore Road, Bethesda, MD 20816-5003 by using the Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

AMSC N/A AREA MCGT

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. Unless otherwise specified, the issues of these documents are those listed in the Department of Defense Index of Specifications and Standards (DoDISS) and the supplement thereto, cited in the solicitation (see 6.2).

STANDARDS

DEPARTMENT OF DEFENSE

MIL-STD-2407 - Vector Product Format

(Unless otherwise indicated, copies of the above specifications, standards, and handbooks are available from the DoD Single Stock Point for Military Specifications, Standards and Related Documents (DODSSP), 700 Robbins Avenue, Building 4D, Philadelphia, PA 19111-5094, or at the DODSSP web site at www.dodssp.daps.mil)

2.3 Order of precedence. In the event of a conflict between the text of this document and the references cited herein (except for related associated specifications or specification sheets) the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

3. REQUIREMENTS

3.1 First Article When specified (see 6.2), a sample shall be subjected to first article inspection in accordance with 4.2.

3.2 Binary patch update methodology. Vector Product Format (VPF) data shall be updated by means of binary patch methodology. The binary patch contains instructions on how to update the data files in the old data set to be a binary equivalent of the new data set. The generated binary patch can then be disseminated over the Internet or wrapped into a Web-based control that automates the update process.

3.2.1 Library level patches. Binary patches shall be distributed at the VPF library level, so individual libraries in a VPF data set may be updated to suit a user's requirements, and keep the binary patches to a manageable size.

3.2.2 Cumulative update. To avoid the situation where new binary patches are applied to previously applied binary patches, the binary patch shall be cumulative and contain all of the updates made to the VPF data since the original data set was issued by NIMA. Some users may have missed applying a previously issued patch and a cumulative patch will ensure that the binary patch is applied to a consistent, baselined data set in the field. Therefore it is important that the binary patches only be applied to the original base data set and not to versions of the data sets to which previous patches have already been applied.

3.3 VPF Database Update (VDU) Update Layer. For libraries that require tracking and accountability for changes made to the base data, an update layer shall be created and distributed along with the patch update. A VDU update layer is a VPF coverage that can be added to the underlying VPF data set. It contains features and attribution that provide indications of the changes that have been made to the geospatial data, and the sources of the changes. This stand-alone update layer is not intended to be a self-contained VPF data set because although it contains the tables and relations typically found in a VPF data set at the coverage level, it lacks the data base tables and library structure mandated in MIL-STD-2407 (see Section 3.6.2). When an update layer is required, it shall be based on the geographical extent of the initial data library. Not all VPF products have a requirement for maintaining a separate record of changes made to the data set and these do not require a VDU update layer, even though the base geospatial data may be updated by binary patch methodology.

3.3.1 Significance of changes in VDU Update Layer . Generally, a VDU record of changes is required for systems used for safety of navigation applications, such as the ECDIS-N that uses DNC data. The VDU update layer contains only those changes that are significant for the user, for example, safety of navigation information - announced by Notices to Mariners, Notices to Airmen (NOTAM), etc. The VDU update layer has been designed to be as small as possible because many users are operating under constrained bandwidth environments.

3.3.2 Changes not included in VDU Update Layer. Minor changes are frequently made to the underlying data files during VPF product maintenance. Although contributing to the overall quality of the VPF product, these changes do not in themselves constitute a significant change. To minimize the size of the VDU update layer, these minor changes should not be included in the VDU update layer.

3.4 Operating systems supported. The binary patch file update of VPF base data and the VDU update layer shall support hosting on Windows (NT, 2000, XP), Solaris, and HP-UX platforms.

3.5 Security.

3.5.1 Security Classification. The security classification of the products generated by the use of these specifications will be the lowest category practicable. When it is necessary to assign a security classification to the product, it shall be in accordance with established national security procedures.

3.6 Relationship of base product to VDU update layer. When a VDU update layer is implemented in a VPF data set, the following changes are required to the base VPF product.

3.6.1 Coverage attribute table (cat). A coverage attribute table (cat) identifies the coverage name, description, and level of topology for each coverage found in a VPF library. Only those coverages that actually exist within a specific library are listed in that library's coverage attribute table. Products implementing the VDU update layer will not have a VDU Update (upd) coverage in the coverage attribute table when the VPF data set is initially issued by NIMA. However, if an update layer is required for a product, the following changes will need to be made to the base VPF data set:

a. Edit the coverage attribute table to add a row for the upd coverage at topology level 3.

b. Add the update.doc table containing the appropriate edition number information at the library level

c. Add the upd coverage itself.

3.6.2 Other mandatory tables in VPF data base. Since the upd coverage is embedded in an existing VPF library when the binary patch is applied, other tables required for VPF compliance at the database level and library level are not included in the VDU update layer, because they already exist in the data set (exception see section 3.15).

3.7 VDU Update (upd) Coverage. In addition to the coverages defined for a particular VPF data set by the product specification, an additional layer is required in the product to support user review of data updates.

3.7.1 The upd coverage The upd coverage is a VPF coverage distributed with the binary patch update information for each VPF library, and is designed to flag and provide additional information about significant changes made since the last database edition of the baseline product. Typically this will include features that are significant for safety of navigation. It will contain information such as Notice to Mariners number and accompanying text, current and previous feature IDs, and coverage and feature class information. The update layer is expected to change with each VDU update, so changes to the update layer shall be contained within the library patch. In order to minimize the impact of the upd coverage on library patch sizes, an efficient VPF implementation of the Update Layer is required.

3.8 Update coverage tiling. The upd coverage is untiled and 2-dimensional (2D).

3.9 Topology. The upd coverage has level 3 topology (see MIL-STD-2407).

3.10 Update coverage change indicators. The upd coverage shall contain both minimum bounding rectangles (MBR) and points to indicate features that have changed. This allows application software to key on either the MBR or point feature. User display systems utilizing VPF data sets will then be able to read this layer and step users through a review process, highlighting spatial and feature updates.

3.10.1 Minimum bounding rectangle. In order to display changes to a user for review, the update layer shall provide a MBR around line and area features that have changed.

3.10.2 Change points. In addition to the MBR around area and line features, point features shall be used as location indicators for area, line, and point feature changes. Therefore, changes to area and line features are indicated by both an MBR and a point. The point shall be placed within the area feature or on the line feature.

3.11 Complex feature construct. The upd coverage is implemented as a VPF-compliant coverage within each VPF data set library. In order to support an efficient VPF implementation, the VDU update layer uses the VPF complex feature implementation. This allows producers of VPF update layers to retain attributes at the complex feature level and use both an update area feature table (updatea.aft) and an update point feature table (updatep.pft) to point to multiple spatial components.

The complex feature type is not strictly required to indicate the relationship between two features, but rather is implemented in order to minimize the size of the data set, given the requirement that both area and point features are required for the same change, and data set size is a critical design limitation. FIGURE 1 shows the relationship between the complex feature table, area and point feature tables, and their required join tables.

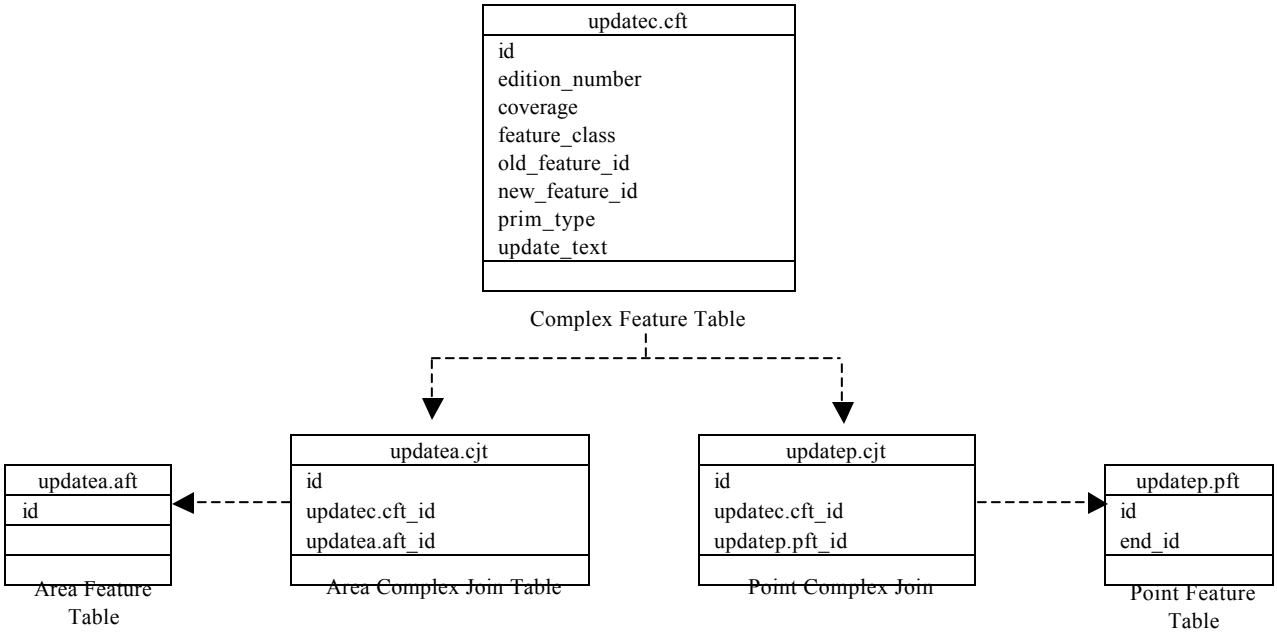

FIGURE 1 Feature-level table architecture.

3.12 Links between upd coverage features and data features. To facilitate tracking of change information contained in the upd coverage with the features in the data coverages affected by the change, the updatec.cft contains the coverage, feature class, and feature ID information of the changed feature, both before and after the change has been made.

3.13 Types of change indicators. The type of change shall be indicated by the old and new feature IDs as follows:

Action	Old ID	New ID
Addition:	null	populated
Deletion:	populated	null
Change:	populated	populated

3.14 Update coverage tables. The following tables are required only if an update coverage is required for the specific VPF product being updated. The table headers depicted in the following tables show all possible columns allowed by MIL-STD-2407. The actual structure of the table headers implemented in the upd coverage shall conform to the product to which the upd coverage is being added.

TABLE 1 Update Character value description table (char.vdt).

Thematic Layer: Update
Coverage Name: upd
Feature Table Description: Update Character Value Description Table
Table Name: char.vdt

{Header length}L; Update Value Character Value Description Table;-; id=I,1,P,row identifier,-,-,-,: table=T,12,N,name of the feature table,-,-,-,: attribute=T,16,N,column name,-,-,-,: value=T,5,N,unique value of attribute,-,-,-,: description=T,*,N,description of value,-,-,-,:				
1	updatec.cft	prim_type	cnd	Connected Node
2	updatec.cft	prim_type	end	Entity Node
3	updatec.cft	prim_type	edg	Edge
4	updatec.cft	prim_type	fac	Face
5	updatec.cft	update_text	N_P	Unpopulated
6	updatec.cft	update_text	N_A	Not Applicable
7	fca	type	P	Point/Node Feature
8	fca	type	A	Area Feature
9	fca	type	C	Complex Feature

TABLE 2 Update complex feature table (updatec.cft).

Thematic Layer:	Update
Coverage Name:	upd
Feature Table Description:	Update Complex Feature Table
Table Name:	updatec.cft
Thematic Index ID Number	1

```
{Header length}L;
Update Complex Feature Table;-;
id=I,1,P,Row Identifier,-,-,-,:
edition_number=F,1,N,Edition number of library,-,-,-,:
coverage=T,*,N,Name of coverage where changed feature resides,-,-,-,:
feature_class=T,8,N,Name of feature class of changed feature,-,-,-,:
old_feature_id=I,1,N,ID number of changed feature in base data set,-,-,-,:
new_feature_id=I,1,N,ID number of changed feature in updated data set,-,-,-,:
prim_type=T,3,N,Type of primitive affected by change,char.vdt,-,-:
update_text=T,*,N,Description of change,-,-,-,:;
```

Column	Description	Value
id	Row Identifier	sequential beginning at 1
edition_number	Edition number of library	Real number indicating the current edition number and patch of this library. Example: 23.1
coverage	Name of coverage where changed feature resides	Character string Example: “Obstructions” or “nav”
feature_class	Name of feature class of changed feature	Character string Example: “limbndyl”
old_feature_id	ID number of changed feature in base data set	Unique ID of the changed feature in the base data set
new_feature_id	ID number of changed feature in updated data set	Unique ID of the changed feature in the updated data set
prim_type	Type of primitive affected by change	Character string One of end, cnd, edg or fac
update_text	Description of change	Null, N_P, N_A or actual string

TABLE 3 Update point feature table (updatep.pft).

Thematic Layer:	Update
Coverage Name:	upd
Feature Table Description:	Update Point Feature Table
Table Name:	updatep.pft
Thematix Index ID Number	2

```
{Header length}L;
Update Point Feature Table;-;
id=I,1,P,Row Identifier,-,-,-,:
end_id,=I,1,N,Entity Node Primitive ID,-,end2_id.pti,-,:;
```

Column	Description	Value
id	Row Identifier	sequential beginning at 1

TABLE 4 Update point complex join table (updatep.cjt).

```
{Header length}L;
Update Point Complex Join Table;-;
id=I,1,P,Row Identifier,-,-,-,:
updatec.cft_id=I,1,N,Complex Feature Key,-,cft1_2.jti,-,:
updatep.pft_id=I,1,N,Base Feature Key,-,pft2_1.jti,-,:;
```

TABLE 5 Update area feature table (updatea.aft).

Thematic Layer:	Update
Coverage Name:	upd
Feature Table Description:	Update Area Feature Table
Table Name:	updatea.aft
Thematic Index ID Number	3

```
{Header length}L;
Update Area Feature Table;-;
id=I,1,P,Row Identifier,-,-,-,:;
```

Column	Description	Value
id	Row Identifier	sequential beginning at 1

TABLE 6 Update area feature join table (updatea.ajt).

Thematic Layer:	Update
Coverage Name:	upd
Feature Table Description:	Update Area Feature Join Table
Table Name:	updatea.ajt

```
{Header length}L;
Update Area Feature Join Table;-;
id=I,1,P,Row Identifier,-,-,-,:
updatea.aft_id=I,1,N,Feature Key,-,updatea.jti,-,:
fac_id=I,1,N,Face Primitive ID,-,fac3_id.jti,-,:;
```

TABLE 7 Update area complex join table (updatea.cjt).

```
{Header length}L;
Update Area Complex Join Table;-;
id=I,1,P,Row Identifier,-,-,-,:
updatec.cft_id=I,1,N,Complex Feature Key,-,cft1_3.jti,-,:
updatea.aft_id=I,1,N,Base Feature Key,-,aft3_1.jti,-,:;
```

TABLE 8 Format for an entity node primitive table (end).

```
{Header length}L;
Entity Node Primitive Table;-;
id=I,1,P,Row Identifier,-,-,-,:
containing_face=I,1,N,Foreign Key to Face Table,-,-,-:
coordinate=C/H[1],1,N,Coordinate of Entity Node,-,-,-,:;
```

NOTE:

1. The type of coordinate shall be dependent on the requirements of the base data. The type of coordinate used shall be defined in the contract guidance.

TABLE 9 Format for a connected node primitive table (cnd).

```
{Header length}L;
Connected Node Primitive Table;-;
id=I,1,P,Row Identifier,-,-,-,:
first_edge=I,1,N,Foreign Key to Edge Table,-,-,-:
coordinate=C/H[1],1,N,Coordinate of Connected Node,-,-,-,:;
```

NOTE:

1. The type of coordinate shall be dependent on the requirements of the base data. The type of coordinate used shall be defined in the contract guidance.

TABLE 10 Format for an edge primitive table (edg).

```
{Header length}L;
Edge Primitive Table;-;
id=I,1,P,Row Identifier,-,-,-,:
start_node=I,1,N,Start Node,-,-,-:
end_node=I,1,N,End Node,-,-,-:
right_face=I,1,N,Right Face,-,-,-:
left_face=I,1,N,Left Face,-,-,-:
right_edge=I,1,N,Right Edge from End Node,-,-,-:
left_edge=I,1,N,Left Edge from Start Node,-,-,-:
coordinates=C/H[1],*,N,Coordinates of Edge,-,-,-,:;
```

NOTE:

1. The type of coordinate shall be dependent on the requirements of the base data. The type of coordinate used shall be defined in the contract guidance

TABLE 11 Format for a face table (fac).

{Header length}L; Face Primitive Table;-; id=I,1,P,Row Identifier,-,-,-,: ring_ptr=I,1,N,Foreign Key to Ring Table,-,-,-,:;

TABLE 12 Format for a ring table (rng).

{Header length}L; Ring Table;-; id=I,1,P,Row Identifier,-,-,-,: face_id=I,1,P,Foreign Key to Face Table,-,-,-,: start_edge=I,1,N,Foreign Key to Edge Table,-,-,-,:;

TABLE 13 Format for a bounding rectangle table (ebr,fbr).

{Header length}L; Bounding Rectangle Table;-; id=I,1,P,Row Identifier,-,-,-,: xmin=F/I[1],1,P,Minimum X Coordinate,-,-,-,: ymin=F/I[1],1,P,Minimum Y Coordinate,-,-,-,: xmax=F/I[1],1,P,Maximum X Coordinate,-,-,-,: ymax=F/I[1],P,Maximum Y Coordinate,-,-,-,:;

NOTE:

1. The type of coordinate shall be dependent on the requirements of the base data. The type of coordinate used shall be defined in the contract guidance

TABLE 14 Update feature class attribute table.

Thematic Layer: Update
Coverage Name: **upd**
Feature Table Description: Update Feature Class Attribute Table
Table Name: **fca**

```
{Header length}L;
Update Feature Class Attribute Table;-;
id=I,1,P,Row Identifier,-,-,-,:
fclass=T,8,U,Feature Class Name,-,-,-,:
type=T,1,N,Feature type,char.vdt,-,-:
descr=T,*,N,Description,-,-,-,:;
```

Column	Description	Value	Value Meaning	Feature Class
id	Row Identifier	Sequential beginning with 1		
fclass	Feature Class Name			
		updatec		
		updatep		
		updatea		
type	Feature Type			
		C	Complex Feature	updatec
		P	Point/Node Feature	updatep
		A	Area Feature	updatea
descr	Description			
		Update Complex Feature		updatec
		Update Point Feature		updatep
		Update Area Feature		updatea

TABLE 15 Update feature class schema table.

Thematic Layer: Update
Coverage Name: **upd**
Feature Table Description: Update Feature Class Schema Table
Table Name: **fcs**

{Header length}L; Update Feature Class Schema Table;-; id=I,1,P,Row Identifier,-,-,-,: feature_class=T,8,N,Name of Feature Class,-,-,-,: table1=T,12,N,First Table,-,-,-,: table1_key=T,16,N,Column Name in First Table,-,-,-,: table2=T,12,N,Second Table,-,-,-,: table2_key=T,16,N,Column Name in Second Table,-,-,-,:;					
1	updatec	updatec.cft	id	updatep.cjt	updatec.cft_id
2	updatec	updatep.cjt	updatep.pft_id	updatep.pft	id
3	updatec	updatep.pft	end_id	end	id
4	updatec	end	id	updatep.pft	end_id
5	updatec	updatep.pft	id	updatep.cjt	updatep.pft_id
6	updatec	updatep.cjt	updatec.cft_id	updatec.cft	id
7	updatec	updatec.cft	id	updatea.cjt	updatec.cft_id
8	updatec	updatea.cjt	updatea.aft_id	updatea.aft	id
9	updatec	updatea.aft	id	updatea.ajt	updatea.aft_id
10	updatec	updatea.ajt	fac_id	fac	id
11	updatec	fac	id	updatea.ajt	fac_id
12	updatec	updatea.ajt	updatea.aft_id	updatea.aft	id
13	updatec	updatea.aft	id	updatea.cjt	updatea.aft_id
14	updatec	updatea.cjt	updatec.cft_id	updatec.cft	id
15	updatea	updatea.aft	id	updatea.ajt	updatea.aft_id
16	updatea	updatea.ajt	fac_id	fac	id
17	updatea	fac	id	updatea.ajt	fac_id
18	updatea	updatea.ajt	updatea.aft_id	updatea.aft	id
19	updatep	updatep.pft	end_id	end	id
20	updatep	end	id	updatep.pft	end_id

3.15 Library-level VDU file. The VDU update information shall also include a file at the library level, one directory above the UPD coverage. This file contains a structured data field that specifies the date that the library was last updated. If libraries are updated asynchronously, it becomes logistically difficult to selectively update the library's metadata tables to reflect the dates of the libraries actually updated on the end user's system. Adding an update edition file to a library allows each library to retain its own edition date and currency information. Table 16 shows the structure of the update.doc file that stores currency information for a library.

TABLE 16 Update documentation table.

{Header length}L; Update Documentation Table;-; id=I,1,P,Row Identifier,-,-,-,: update_text=T,*,N,Update Text Information,-,-,-,: library_name=T,8,U, The library name,-,-,-,:;

4. VERIFICATION

4.1 Classification of inspection. The inspection requirements specified herein are classified as follows:

a. First article inspection (see 4.2)

b. Conformance inspection (see 4.3)

4.2 First article inspection. When a first article inspection is required (see 3.1), the product shall be examined as specified in 4.3.1 and 4.3.2.

4.3 Conformance inspection. Conformance inspection shall include file comparison procedures described below.

4.3.1 Updated data file comparison. Update patches generated by patch generation software shall be applied to the old version of the VPF data set. Software routines that detect differences between two files shall then be executed. There shall be no differences between the new version of the VPF product, and the old version of the VPF product, after the old version has been updated using the binary patch methodology.

4.3.2 Validation of update layer data.

4.3.2.1 Matching points. The point change indicators for area and line features shall be checked to ensure that they fall within a changed area feature or on a changed line feature.

5. PACKAGING

5.1 Packaging**.** For acquisitions purposes, the packaging requirements shall be as specified in the contract or order (see 6.2). When actual packaging of material is to be performed by DoD personnel, these personnel need to contact the responsible packaging activity to ascertain requisite packaging requirements. Packaging requirements are maintained by the Inventory Control Point's packaging activity within the Military Department or Defense Agency, or within the Military Department's Systems Command. Packaging data retrieval is available from the managing Military Department's or Defense Agency's automated packaging files, CD-ROM products, or by contacting the responsible packaging activity.

6. NOTES

(This section contains information of a general or explanatory nature that may be helpful, but is not mandatory).

6.1 Intended use. The intended use of the VDU is to allow users to update VPF data sets using a binary patch methodology. The intended use of the VDU update layer is to provide a format for additional data to be added to the VPF data set to allow tracking and accountability of change data, including marking of changed features and other information, and date of last update of a data library. VDU is not necessarily a military-unique process, but the data sets being updated by VDU, and

consequently modified by this specification are for the most part DoD-generated, and have restricted distribution to DoD only.

6.2 Acquisition Requirements. Acquisition documents must specify the following:

a. Title, number and date of this specification.

b. Issue of the DoDISS to be cited in the solicitation, and if required, the specific issue of individual documents referenced (see 2.2.1 and 2.3).

c. When a first article is required (see 3.1 and 4.2).

6.3 Supersession. This section is not applicable to this specification.

6.4 Definitions.

6.4.1 Binary patch methodology. An update process that compares two data sets and produces a patch file that contains the differences between the old and the new data set.

6.4.2 Complex feature. A single feature that relates directly to other features rather than to a primitive. A single feature composed of other features, either simple or complex. (MIL-STD-2407)

6.4.3 Coverage. A set of feature classes that has a specified spatial extent and in which the primitives interconnect as described by the coverage's topology. (MIL-STD-2407)

6.4.4 Library. A collection of one or more coverages contained within a specified spatial extent, all of which share a single coordinate system. (MIL-STD-2407)

6.5 Subject term (key word) listing.

Binary patch
Digital Nautical Chart (DNC)
Electronic chart
Navigation
Vector Product Format (VPF)

6.6 Standardization agreements. This section is not applicable to this specification.

6.7 Changes from previous issue. This section is not applicable to this specification.

6.8 NIMA Operational Help Desk. For questions concerning this or other NIMA-prepared standards, specifications, or products, please telephone the NIMA Operational Help Desk at 1-800-455-0899, Commercial 314-263-4864, or DSN 693-4864.

VPF DATABASE UPDATE (VDU) METHODOLOGY

A.1 SCOPE

A.1.1 Scope. This Appendix provides an overview of a methodology and the functional requirements for VPF Database Update (VDU). This Appendix is not a mandatory part of the specification. The information contained herein is intended for guidance.

A.2 APPLICABLE DOCUMENTS

(This section is not applicable to this appendix.)

A.3 VDU Design and Methodology

A.3.1. System Components. The three main system components are Patch Build, Patch Dissemination, and Patch Apply and they are described below:

A.3.1.1 Patch Build:

The patch build process compares two datasets and produces a patch file that contains the differences between the old and new dataset. Several COTS tools build patch files between old and new datasets. Additionally, a developer may opt to develop a custom patch build application. In either case, the user is provided with a Patch Apply utility that is able to apply the patch file to a base dataset to generate a binary-equivalent new dataset. The underlying patch build utilities and algorithms should be transparent to the end user, as the user is provided with an apply utility, or the apply utility is built directly into a navigation or display system.

The general steps for building a patch file are:

1. Obtain copy of current base data Edition 1.0 from NIMA Database
2. Obtain copy of new version Edition 1.1 from NIMA Database
3. Run differencing routine on base Edition 1.0 and new version Edition 1.1 to create the patch file
4. Apply patch file using the application utility to the original base Edition 1.0 to validate the patch file in Step 5
5. Run the differencing routine on the newly updated original base Edition 1.0 “prime” and the new version Edition 1.1 as a quality assurance (QA) step and verify and validate that the patches are valid (i.e., differences are zero, and therefore the two editions are identical)
6. Proceed to patch dissemination

A.3.1.2 Patch Dissemination:

Patches can be distributed through conventional channels such as CDROMs and electronic channels including the Internet. Patch files generated in the patch build phase should be signed with a digital certificate to insure authentication. Digital signatures can be applied using a number of publicly available signing tools. Signed patch files are then distributed via standard Web servers for download.

The general steps for disseminating a patch file are:

1. Obtain patch file from patch build process
2. Digitally sign patch file
3. Post patch file for user download

A.3.1.3 Patch Apply:

Users require a patch application utility that is compatible with the algorithm used to build the disseminated patch. COTS patch vendors usually distribute patch apply software royalty-free, so apply utilities should be made freely downloadable and distributable to end-users.

The general steps for applying a patch file are:

1. Obtain "patch file" from Website or CDROM
2. Provide access (directory, path) to copy (load base Edition 1.0 CDROM or make copy on hard disc) of current base Edition 1.0
3. Verification by "patch file" application software that the correct directories, paths and versions of base Edition 1.0 and "patch files" 1.1 are validated prior to patching
4. Apply "patch file" 1.1 to copy of current base Edition 1.0 which now becomes Edition 1.1
5. Verification by "patch file" application utility software that the "patch" was applied correctly
6. Edition 1.1 now verified and validated to be imported to replace previous Edition 1.0
7. User's Edition system logs date/time and content of applied "patch file" change information and replacement of Edition 1.0 with Edition 1.1
8. Obtain copy of new "patch file" 1.2 and repeat steps (2 through 5 above using new "patch file" 1.2).
9. Edition 1.2 now verified and validated to be imported to replace previous Edition 1.1
10. User's Edition system logs date/time and content of applied "patch file" change information and replacement of Edition 1.1 with Edition 1.2

A.4 Component Interactions

A.4.1 System Components.

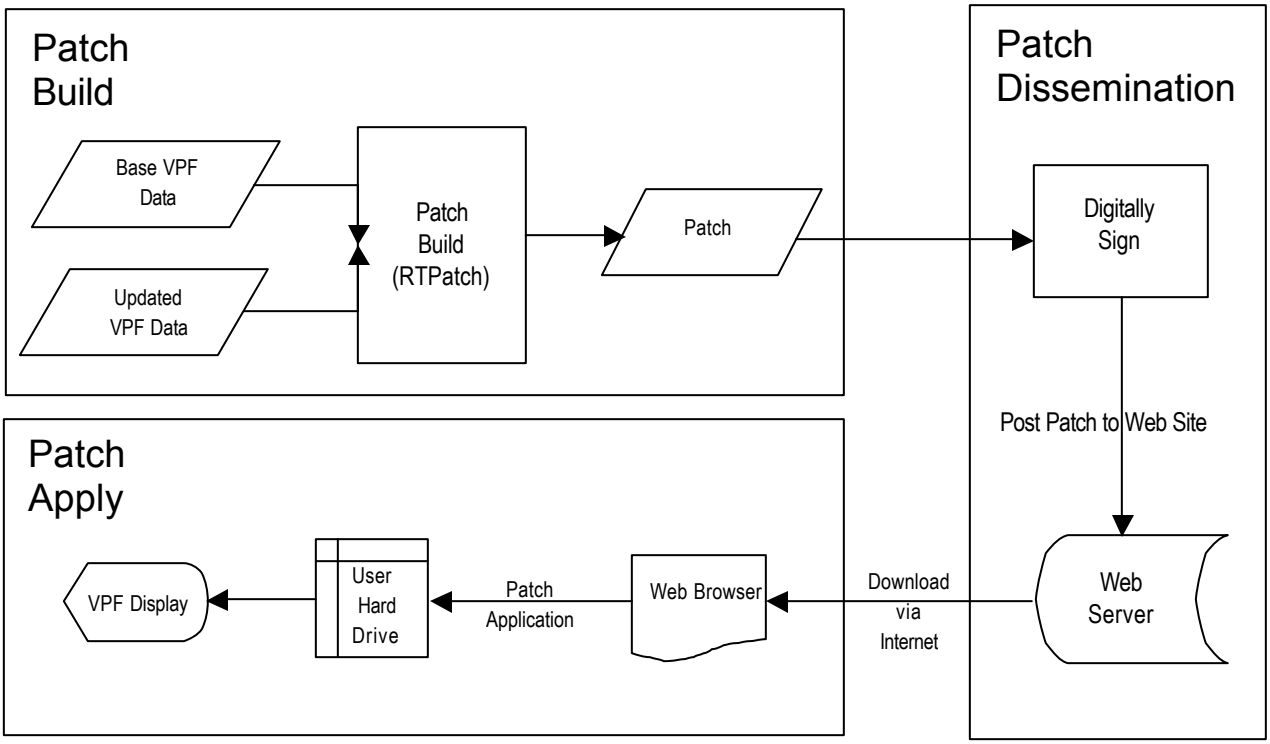

FIGURE 2 Patch component interactions.

The patch component interactions can be clearly defined:

A.4.1.1 Patch Build

Input: Old and new VPF datasets
Action Performed: Patch generation between old and new dataset
Output: Patch between old and new dataset
Exit Action: Post patch to Web site for distribution

A.4.1.2 Patch Dissemination

Input: Patch built from Patch Build process
Action Performed: Digitally sign patch file, provide user access to patch file

Output: Patch downloaded to user
Exit Action: Patch Apply process

A.4.1.3 Patch Apply

Input: Patch downloaded from Patch dissemination process and old dataset
Action Performed: Patch applied to old dataset
Output: Patched dataset, binary equivalent to new dataset from Patch Build process
Exit Action: Notify user of successful application

A.4.2 Interface Interactions

The component interfaces are network based. Once a patch is generated with the Patch Build process, it is copied or posted to a Web server over the network. Users access the patch file during the Patch Dissemination and Patch Apply phases over standard HTTP requests, via a choice of network communication paths (SIPRN, OSIS, JWICS, and NIPR/EXTRA Nets) to NIMA's Gateway/www servers.

INDEX

Subject Page

INDEX

CONCLUDING MATERIAL

Custodians:
Navy - NO

Preparing activity:
NIMA - MP
(Project MCGT-0333)

STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL

INSTRUCTIONS

1. The preparing activity must complete blocks 1, 2, 3, and 8. In block 1, both the document number and revision letter should be given.
2. The submitter of this form must complete blocks 4, 5, 6, and 7.
3. The preparing activity must provide a reply within 30 days from receipt of the form.

NOTE: This form may not be used to request copies of documents, nor to request waivers, or clarification of requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

I RECOMMEND A CHANGE:	**1. DOCUMENT NUMBER** MIL-PRF-32118	**2. DOCUMENT DATE** *(YYMMDD)* 030320

3. **DOCUMENT TITLE**
VECTOR PRODUCT FORMAT (VPF) DATABASE UPDATE (VDU)

4. **NATURE OF CHANGE** (Identify paragraph number and include proposed rewrite, if possible. Attach extra sheets as needed.)

5. REASON FOR RECOMMENDATION

6. SUBMITTER		
a. NAME *(Last, First, Middle Initial)*	b. ORGANIZATION	
c. ADDRESS *(Include Zip Code)*	d. TELEPHONE *(Include Area Code)* (1) Commercial (2) *AUTOVON* *(If applicable)*	**7. DATE SUBMITTED** *(YYMMDD)*

8. PREPARING ACTIVITY	
a. NAME National Imagery and Mapping Agency	b. TELEPHONE *(Include Area Code)* (1) Commercial (703) 262-4409 (2) AUTOVON 898-8539
c. ADDRESS *(Include ZIP Code)* National Imagery and Mapping Agency ATTN: ETA, MS P-24 12310 Sunrise Valley Drive Reston, VA 20191-3449	**IF YOU DO NOT RECEIVE A REPLY WITHIN 45 DAYS, CONTACT:** Defense Standardization Program Office 8725 John J. Kingman Road, Suite 2533 Fort Belvoir, VA 22060-6879 Telephone: (703) 767-6888, DSN: 427-6888