The Common Operations and Development Environment (CODE) for the WSR-88D Open RPG

CODE B17.0r1.13: May 2016

Includes ORPG Build 17.0r1.13

Volume 4. CODE Utility Guide

- Instructions for using CVT, CVG, and other CODE utilities -

The **U.S. Government Edition** of CODE is the complete version. Distribution is limited to within the United States Government.

The **Public Edition** of CODE is intended for public release. Certain Copyrighted material has been removed to permit release outside the U.S. Government.

CODE provides:

- Instructions for setting up the development environment (includes ORPG source code)
- Guidance for compiling software and configuring new ORPG tasks & products
- Instructions for definition and use of algorithm adaptation data and algorithm dependent parameters
- API Programming Guide and the structure of WSR-88D algorithms (with sample algorithms)
- WSR-88D specific analysis tools
- A set of WSR-88D Archive II Data files and other special test case data.

CODE User provides:

- An Intel PC with Red Hat Enterprise Workstation.

CODE Guide Volume 1. Guide to Setting Up the Development Environment
- Document 1. CODE Specific ORPG Installation Instructions
  - I - Preparation for Installation
  - II - Installation Instructions
  - III - Supplemental Information
  - IV - Running the ORPG
- Document 2. Installing CODE Software
  - I - Software Requisites for CODE Utilities
  - II - Instructions for CODE Utilities
  - III - Instructions for Sample Algorithms

CODE Guide Volume 2. ORPG Application Software Development Guide
- Document 1. The ORPG Architecture
- Document 2. The ORPG Development Environment
  - I - Integrating Development Software with ORPG Source Code
  - II - Compiling Software in the ORPG Environment
  - III - ORPG Configuration for Application Developers
  - IV - Configuring Site Specific Adaptation Data
- Document 3. WSR-88D Final Product Format
  - I - Product Block Structure
  - II - Traditional Product Data Packets
  - III - Generic Product Components
  - IV - ORPG Application Dependent Parameters
- Document 4. ORPG Internal Data for Algorithm Developers
  - I - Base Data Format
  - II - Algorithm Adaptation Data - Configuration & Use
  - III - Other Data Inputs

CODE Guide Volume 3. WSR-88D Algorithm Programming Guide
- Document 1. The WSR-88D Algorithm API Overview
- Document 2. The WSR-88D Algorithm API Reference
  - I - API Service Registration / Initialization
  - II - Control - Input/Output - Abort Services
  - III - Final Product Construction
  - IV - API Convenience Functions
- Document 3. The WSR-88D Algorithm Structure and Sample Algorithms
  - I - WSR-88D Algorithm Structure
  - II - Sample Algorithms
  - III - Writing Product Data Fields
- Document 4. Special Topics
  - I - Topics Related to Using the Development Environment
  - II - Topics Related to Reading Radial Base Data
  - III - Topics Related to Writing Algorithms

CODE Guide Volume 4. CODE Utility Guide
- Document 1. CODEview Text (CVT) - ASCII Product Display
- Document 2. CODEview Graphics (CVG) - Graphic Product Display
  - I - Displaying Products with CVG
  - II - Configuring Products for Display by CVG
- Document 3. Archive II Disk File Ingest - play_a2 Tool
- Document 4. Product Distribution with the nbtcp Tool
- Document 5. Additional CODE / ORPG Tools

Volume 4. CODE Utility Guide

Document 1. CODEview Text (CVT)

The CODEview Text (CVT) utility provides a means for evaluating the contents of ICD formatted graphic products. It provides the capability to print an inventory of the current contents of the product database linear buffer and can selectively display (in ASCII format) various components of the product including all header information at one time.

Document 2. CODEview Graphics (CVG)

CODEview Graphics (CVG) is an interactive tool providing a graphic display of ICD formatted WSR-88D final products.

Section I Displaying Products with CVG
Section II Configuring Products for Display by CVG

Document 3. Archive II Disk File Ingest - play_a2 Tool

The ORPG utility **`play_a2`** provides a capability for running an ORPG using Archive II disk files as a source of radar data. This permits application development without a connection to a radar and can be used to provide a convenient source of specialized input data for testing.

Document 4. Product Distribution with the nbtcp Tool

This document describes how to use **`nbtcp`** in a CODE ORPG environment. **`nbtcp`** simulates a Class 1 user and communicates with an ORPG to initiate product distribution to a selected directory. The result is a collection of individual product binary files for selected products from a running ORPG. This is one method of testing the request parameters for selected products.

Document 5. Additional CODE / ORPG Tools

Several additional CODE utilities and ORPG tools are documented here. The ORPG tools are a site adaptation utility **`change_radar`**. The CODE tools are a color palette editor (for **`CVG`**), and a utility to make additional background maps for use with **`CVG`**

Appendices

Volume 4. CODE Utility Guide

Document 1. CODEview Text (CVT)

CODEview Text Utility 4.4.3

This Guide covers CODEview Text (**CVT**) CVT 4.4.3 which is included with the source code for ORPG Build 12 which is installed when compiling the ORPG following the CODE instructions.

IMPORTANT NOTE: In order to use **CVT** with multiple ORPG builds, the environmental variable **CV_ORPG_BUILD** must be set to the build number (e.g., **9**). This is most important when reading from a database produced by an ORPG before Build 6.

Part A. Introduction

CODEview Text (CVT) is a command line utility used to explore the inner components of ICD formatted products and the system requests that are used to create them. ICD product formats are defined in the Interface Control Document (ICD) for the RPG to Class 1 User, document 2620001.

CVT 4.4.3 (integrated with ORPG Build 12) - WHAT'S NEW?

CVT 4.4.3 Changes:

- A Build 12 change in the radial header for the generic radial component changed the azimuth from center azimuth to beginning azimuth.

CVT 4.4.2 Changes:

- Preconfigured the Scale / Offset parameters for the DP test products to the decoding logic, the user does not have to add custom decode parameters.

CVT 4.4.1 Changes:

- **Corrected an error in parsing data packet 9. Not all packets were displayed**.
- Modified the outputs of the volume and product generation times in several commands so all outputs are in the same format.

- The product generation time in the internal 96-byte header was labeled incorrectly as local time.
- Clarified the output of the generation time in the generic product structure header.

CVT 4.4 Changes:

- **Added the ability to decode data level in packet 16 and the generic radial using Scale-Offset parameters either in the product or via user supplied configuration file**. Included Scale-Offset decoding parameters for existing digital products in CVT. The scale selector commands (**`scaler`**, **`scalev1`**, **`scalev2`**, and **`scalesw`**) are deprecated but are currently retained for backward compatibility.
- **Significantly enhanced the output of the generic data packet**.
    - CVT can provide a simple listing of all components or components of a particular type.
    - In addition to printing all components, CVT can print out components of a particular type or a specific single component.
    - The printing of parameters associated with the product and with components can be turned off.
    - The format of the output has been improved and generic product fields having meaning encoded into an integer value are decoded.
    - The output format of data arrays (generic radials) has been improved along with the ability to decode data levels.
    - Added the capability to select a single radial or groups of radials to the display of the generic radial.
- Modified BSCAN output to handle more than 400 radials to support super resolution products.
- Fixed a bug in BSCAN output when viewing historical data before ORDA that did not have exactly 360 radials.
- **Added the capability to print out all layers of the symbology block by using the following command: '`layer 0`'**.
- **Modified the '`radial`' and '`row`' commands to be used with the 'layer' command**.
- Modified the display of radial and raster packets to permit use of the basic display command '**`layer C`**'. The content selector is now optional. Before the '**`radial`**' and '**`row`**' commands had to be used to explicitly state which radials / rows to display. If the content selector is not used, all radials/rows is assumed.

CVT 4.3.3 Changes:

- Added the volume date time as the last column of output from the inventory commands and the database search command.

Product Database Functions

All final products are stored in the central product database. The primary purpose of **CODEview Text (CVT)** is to display products contained within the main product database linear buffer. **CVT** can display inventories and search for specified products in the product database.

Product Buffer Functions

Intermediate products are generally stored in individual "product-specific" linear buffers. **CVT** can be used to display inventories of products within "product-specific" linear buffers. If the structure of intermediate products were standardized, **CVT** could be extended to display intermediate product data.

> Click Here for an explanation of the term *linear buffer* and a description of the contents of the "product-specific" linear buffers and the product database linear buffer.

Other Functions

CVT is able to display the queue within the ORPG Product Request linear buffer to enable developers to monitor ICD product generation requests. In addition, **CVT** can dump an entire product to an ASCII file formatted in Hexadecimal notation for analysis. Also, a binary image of an ICD formatted product can be extracted from the database and written to a file for further processing. The CODEview Graphics utility can display these binary files.

Compressed Products

CVT transparently handles final products that have been compressed by the ORPG. To accomplish this **CVT** looks at the 8th product dependent parameter and checks for the presence of the block divider following the *Product Description Block*.

Display Capabilities

CVT offers many ways to output product data. For example, run length encoded (RLE) data in 'traditional' radial and raster packets can be viewed decoded in Decimal notation or encoded (RLE) in Hexadecimal notation. For radial products, a tabular formatted output called BSCAN can be used. BSCAN is simply a way to output an entire radar screen of data in a structured format (360+ degrees (rows) by 'n' range bins (columns)). For base data products, data scaling functions are available to convert raw moment data into dBZ (if reflectivity) or m/s (if velocity or spectrum width).

CVT is able to decode and display most of the 30+ data packet types. A data packet is a type specification for a particular data structure or display format within the ICD framework.

***Traditional Products*.** A traditional WSR-88D product can consist of multiple data layers (each layer containing specific packet types) creating sophisticated product displays. These data layers are within that portion of the ICD product called the Symbology Block. **CVT** can decode all packet types except Cell Trend Data (packet 21) and Cell Trend Volume Scan Time (packet 22).

In addition to decoding and displaying the contents of the Symbology Block layers, **CVT** can also display the contents of the Tabular Alphanumeric Block (TAB), the Graphic Alphanumeric Block (GAB), and the Stand Alone Tabular Alphanumeric Product (SATAP).

***Generic Products*.** A WSR-88D Generic Product consists of one generic data packet (packet 28) in layer 1 of the symbology block, and no GAB or TAB. Currently the only option CVT provides is an ASCII output of all product parameters and generic components contained in the generic data packet.

	Name	Packet Code(s)
Vector Packets	Text & Special Symbols	1, 2, 8
	Wind Barb Data	4
	Linked Vector (No Value) Unlinked Vector (No Value) Linked Vector (Uniform Value) Unlinked Vector (Uniform Value)	6 7 9 10
	Vector Arrow Data	5
Special Graphic Symbols	Mesocyclone Data Correlated Shear MESO TVS Data Positive Hail Data Probable Hail Data Storm ID Data HDA Hail Data Point Feature Data SCIT Past Position Data SCIT Forecast Position STI Circle Data ETVS Data	3 11 12 13 14 15 19 20 23 24 25 26
Contour Packets	256-level Digital Radial Data Array	16
	Digital Precipitation Data Array	17
	Precipitation Rate Data Array	18
	Color Contour Vector Linked Contour Vector Un-linked Contour Vector	0802x 0E03x 3501x
	Run Length Encoded Radial Data	AF1Fx
	Run Length Encoded Raster Data	BA0Fx, BA07x
	SuperOb Wind Data Packet	27
	Generic Data Packet supported components: area, radial, & text	28

This includes all ICD defined packets except packets 21 and 22.

Limitations

- While **`CVT`** provides diagnostics to help catch parsing errors (structure errors) in the TAB portion of the product, this assistance is not yet provided for the GAB and for displaying packets in the symbology block.
- **`CVT`** does not yet support decompression of an intermediate product.

Known Problems

- A **`CVT`** '**`check`**' function does not work with product ID 49 (code 62), Storm Structure. This is a unique product that is a stand-alone tabular alphanumeric product (SATAP) that contains additional data not meant for display. **`CVT`** does display the TAB portion with the '**`satap`**' command.
- **`CVT`** does not support the Radar Coded Message (RCM) product. The **`CVT`** summary command incorrectly reports that a SATAP is present when attempting to display the RCM.

Part B. Command Line Parameters

The contents of Part B can be displayed on the terminal by executing the command '**cvt help**'. An abbreviated listing of command line parameters can be displayed with '**cvt h**'.

Key to use of Letters within the following help listing

```
L        Represents a Linear Buffer ID
LBNAME   The name of the Linear Buffer file in the form of a relative
         path under $ORPGDIR (.lb extension optional)
X        Represents the message sequence number of a product found within
         a Linear Buffer. A message sequence number can be found in the
         left-most column of an inventory / database search output
FNAME    Is the name of the a single product binary file. This can be a
         path relative to the current working directory or a full path name.
A B      Represent radial numbers/degrees and row numbers
C        Is a layer number. Used when displaying messages where more
         than one layer is present.
```

NOTE: The Linear Buffer ID is not the same as the legacy product code or message code. For example, product 19, base reflectivity, has a linear buffer number of 2. The product attribute table in the **product_attr_table** configuration file is one source for determining the linear buffer number associated with a product.

- CODEview Text Informational Commands

```
cvt help            Listing of CODEview Text command line arguments and options
cvt h               An abbreviated version of cvt help
cvt version         Display version and release date information
cvt packets         Listing of ICD Packet Codes and Definitions
```

- Linear Buffer Inventory Commands

```
cvt i               Provide an inventory of the main product
                         database linear buffer
cvt i LBNAME        Provide an inventory of the specific product
                         linear buffer with the path "LBNAME" (under $ORPGDIR)
cvt slb L           Search the products database linear buffer for a specific
                         linear buffer ID "L" (product ID)
cvt req LBNAME      Show the inventory of one of the request linear
                         buffer with the path "LBNAME" (under $ORPGDIR)
```

- Two Basic Sources of Product Messages

1. Normally product messages are loaded from the product database. The location of the product, that is its message sequence number '**X**', is determined with one of the inventory commands. This form of the **cvt** command is:

```
cvt msg X <additional command arguments>
```

By default, cvt uses the product database linear buffer configured for the ORPG account currently in use. An alternate database can be temporarily substituted with the CVT_DB environmental variable. Set the value of this variable to the complete path of the alternate database linear buffer filename.

NOTE: If reading a product database created with an ORPG prior to Build 6, the environmental variable CV_ORPG_BUILD must be set to the build number.

2. A second source of product messages are single product binary files. The product in file '**FNAME**' is loaded with the command:

```
cvt load FNAME [-nohdr|-wmohdr] <additional command arguments>
```

By default, it is assumed that the product in the file includes the internal 96-byte header.

- If the product was received by an external user (AWIPS or the nbtcp tool) it will not contain the internal header. In this case the command must include the '**-nohdr**' option.
- If the file was obtained from the central server (e.g., the NCDC web site), the WMO header must be stripped off with the '**-wmohdr**' option.

NOTE: The two extraction commands ('**extract**' and '**hexdump**') do not work when loading a product from a binary file. The 'summary' option must be used to get the summary output when loading from a binary file.

- CVT Product Consistency Checks

When CVT begins any display command two checks are performed.

1. A check is made of the first 120 bytes to determine if the product appears to be a valid final product (ICD format). The results are displayed and the command terminated if a valid ICD product is not detected.

   The **-f** option (force load) can be used to continue with the desired CVT command.

```
cvt msg X -f <additional command arguments>
cvt load FNAME -f <additional command arguments>
```

2. If the product is a final product (ICD), CVT compares specific product data fields for consistency. These are: number of blocks, block offsets, product length, block lengths, and layer / page lengths. The results are displayed.

   A more verbose output can be obtained with the '**cvt msg X check**' command.

- Product Information & Header Display Commands

```
cvt msg X summary    Access message "X" in the products database linear
                          buffer then print a message summary. This includes
                          a list of all Symbology Block layers in the
                          product and the presence of a TAB / GAB.
cvt msg X            Same as cvt msg X summary
                          (this shortcut does not work with cvt load FNAME)

cvt msg X check      Prints a verbose product consistency check

cvt msg X hdr        Print the Pre-ICD product header of message "X"
cvt msg X mhb        Print the message header block of message "X"
cvt msg X pdb        Print the product description block of message "X"
cvt msg X sym        Print the symbology block header of message "X"
cvt msg X fullhdr    Print all header blocks of message "X"
```

- Basic Data Display Commands

The first step in displaying the data is to determine how many layers are present in the symbology block and the presence of a tabular alphanumeric block (TAB) or graphic alphanumeric block (GAB). This can be accomplished with the summary command for a particular message "X":

```
cvt msg X summary
```

A particular data layer "C" can be displayed with the basic command (using a value of 0 for C will result in all layers being displayed):

```
cvt msg X layer C
```

For ***Traditional Products***, special command modifiers, '**`radial`**' and '**`row`**', control the display of layers containing radial and raster data packets.

For ***Generic Products*** the entire product can be displayed using the '**`layer 1`**' command. The command '**`generic`**' permits use of additional command modifiers to control the display.

```
cvt msg X generic
```

- TAB, GAB, and Stand Alone Product Display Commands (Traditional Products)

The commands used to display the GAB, TAB, and the Stand Alone Tabular Alpha Product are:

```
cvt msg X tab    Print the contents of the Tabular Alphanumeric Block
          tabv   Print the MHB and PDB included within the TAB
cvt msg X gab    Print the contents of the Graphic Alphanumeric Block
cvt msg X satap  Print the contents of the Stand Alone Tabular Alpha Product
```

- Radial / Raster Data Display Commands

If the data layer contains radial or raster data packets, the following forms of the display command are used to print the radial (or rows) contained in message "X" layer "C":

```
cvt msg X layer C radial <modifier list>
cvt msg X layer C row <modifier list>
```

For simple products containing one data layer with radial or raster packets, an abbreviated form of the command can be used:

cvt msg X radial <modifier list> Print contents of radials from message "X"
cvt msg X row <modifier list> Print contents of rows from message "X"

The **<modifier list>** provides flexibility in displaying the data. The entire data packet is displayed with default behavior if no **<modifier list>** is provided.

- Radial / Raster Modifier List Structure

A **Radial/Raster Modifier list** consists of an optional **<content selector>** followed by an optional **<format selector>** and an optional **<decode selector>** (**CVT** 4.4 and later) or an optional **<scale selector>**. **Note: Beginning with CVT 4.4, the <scale selector> is deprecated and being replaced by the <decode selector>.**

Parameters within braces **{}** are required. Parameters within brackets **[]** are optional. Possible values of a parameter are separated by "**|**". Default behaviors are implied by a lack of a parameter and are defined in the discussion below.

Modifiers for Radial and Raster Based Products:

```
            [<content selector>]  [<format selector>]    [<decode selector>]
           ---------------------  -------------------  ------------------------
   radial  {A | A B | all} [deg]     [rle | bscan]     {fdecode | pdecode} [D]
      row  {A | A B | all}           [rle]
```

The **<format selector>** (**rle** and **bscan**) are only used with traditional products (not used with the generic radial component).

If no <content selector> is provided, all radials / rows are selected. The default behavior for the <content selector> is to select radials and rows by their ordinal value. For radial data, a **deg** option is provided to select the radials by degrees (the input is in whole degrees and radials are selected inclusively).

If a <format selector> is not used, the data are displayed in Decimal notation and RLE data are decoded (if appropriate). The **rle** option prevents RLE data from being decoded and displays the values in Hexadecimal notation (this parameter has no effect on data that are not RLE). The **bscan** option (for radial based products only) formats the output in a tabular format (azimuth x range) and displays the data in Decimal notation. SPECIAL NOTE: The **bscan** option overrides the

<content selector> and displays all radials. The **bscan** option does not work with a generic radial component.

The **<scale selector>** is being deprecated and being replaced by the **<decode selector>**.

The <decode selector> is used with digital radial data (packet 16) and as described later with the generic area and generic radial components. If a **<decode selector>** (**CVT** 4.4 and newer) is not used, the data (including reflectivity, velocity, and spectrum width) remain encoded as in the product. The optional modifier **D** states the number of digits after the decimal point (0-6). If the modifier is not used the default is 3 digits.

The fdecode option uses Scale-Offset parameters that are either already included in CVT or are contained in a user provided file. The parameter file must be placed into the **~/.cvt/** directory with a name format of '**decode_params.<product id>**'.

The pdecode option uses Scale-Offset parameters within the product itself.

If a **<scale selector>** (prior to **CVT** 4.4) is not used, the data (reflectivity, velocity, and spectrum width) remain scaled as in the product. The **scaler** option rescales reflectivity data to dBZ. The **scalev1** and **scalev2** options rescale velocity to meters/second for data obtained from velocity mode 1 and 2 respectively. The **scalesw** option rescales spectrum width data to meters/second.

- Example Modifier Lists

```
radial A                Data within radial A
radial A deg            Data within radial A degrees
radial A B              Data from radials A to B inclusive
radial A B deg          Data from radial A degrees to B degrees inclusive
radial A B deg rle      Data from radial A degrees to B degrees inclusive
                             leaving data in run length encoded format
radial all bscan        Data from all radials using the bscan format
radial all              All radial data in default (decimal) format
row A                   Data within row A
row A B                 All data from row A to B inclusive
row all                 Data from all rows in default (decimal)format
row A B rle             Data from row A to B inclusive in run length
                             encoded (rle) format
layer C radial all      All radial data in the layer number C (layer
                             numbers can be found using the summary option)
```

- Generic Product Display Commands

Current guidance limits generic products to a single data packet 28 in layer 1 and no GAB and TAB. The basic form for display of a generic product is:

cvt msg X generic <generic modifier list>

If a generic product modifier list is not used then the entire product is displayed (with no decoding of any unsigned integer data arrays).

If the generic product contains a generic radial component, the `radial` modifier can be used with a `<content selector>` to display a subset of the radials:

```
cvt msg X generic radial <content selector>
```

- Generic Product Modifier List Structure

A **`Generic Product Modifier list`** consists of optional selectors: **`<component list>`** selector, **`<component print>`** selector, **`<parameter print>`** selector, and **`<decode selector>`**. These can be used in any order though use of a **`<component list>`** selector will render the **`<component print>`** selector non-functional. The possible values for the selectors are:

```
                              [<component list>]
-------------------------------------------------------------------------------
list_all | list_area | list_rad | list_text | list_table | list_grid | list_event
```

The `component list` selector provides a simple listing of components with an index into the component array. **`list_all`** causes all components to be listed. The other commands create a listing of all components of a particular type. Using any list selector disables all component print selectors.

```
                              [<component print>]
-----------------------------------------------------------------------------
print_area | print_rad | print_text | print_table | print_grid | print_event

print X
```

By default, all components are printed unless a `component print` selector or `component list` selector is used. The **`component print`** selector has two forms. The first causes all components of a particular type to be displayed. The second form **`print X`** causes the component with an index **`X`** to be printed.

```
                              [<parameter print>]
                  ----------------------------------------------
                  no_pparams | no_cparams | no_pparams no_cparams
```

The **`parameter print`** selector controls whether the product parameters and parameters associated with individual components are printed. The default is all parameters printed. Both values can be used together to eliminate the printing of all parameters.

```
                           [<radial content selector>]
                          -----------------------------
                          radial  {A | A B | all} [deg]
```

The `<radial content selector>` can be used with generic radial components. If no selector is provided, all radials are selected. The default behavior for the `<radial content selector>` is to select radials by their ordinal value. A **`deg`** option is provided to select the radials by degrees (the input is in whole degrees and radials are selected inclusively).

```
                              [<decode selector>]
```

```
-----------------------
{fdecode | pdecode} [D]
```

The `<decode selector>` causes the data values in the unsigned integer arrays (associated with radial and grid components) to be decoded using Scale-Offset parameters. The optional modifier **`D`** states the number of digits after the decimal point (0-6). If the modifier is not used the default is 3 digits.

The `fdecode` option uses Scale-Offset parameters that are either already included in CVT or are contained in a user provided file. The parameter file must be placed into the **`~/.cvt/`** directory with a name format of '**`decode_params.<product id>`**'.

The `pdecode` option uses Scale-Offset parameters within the product itself.

- Example Modifier Lists

```
generic                          Print entire generic product
generic no_pparams no_cparams    Print entire generic product, omitting both
                                     product parameters and component parameters
generic list_all                 Print a listing of all components
generic list_area                Print a listing of all area components
generic list_area no_pparams     Print a list of all area components, omit product
                                      parameters
generic print X                  Print the component with the index 'X' in the
                                      component listing
generic print_area               Print all area components in the generic product
generic print_area no_cparams    Print all area components, omit area component
                                      parameters
generic print_radial             Print all radial components
generic print_radial radial A    Print all radial components but only print data
                                      for radial A
generic radial A deg             Print entire generic product except only print
                                      data for radial A degrees in radial component
generic radial A B               Print entire generic product except only print
                                      data from radial A to radial B inclusive
```

For generic products containing radial components, the modifier '**`generic`**' can be omitted if the '**`radial`**' modifier is used. However without the '**`generic`**' modifier the **`[<component list>]`**, **`[<component print>]`**, and **`<parameter print>]`** selectors will not function.

- Product Extraction Commands

`CVT` can extract ICD formatted final product messages from the product database linear buffer. In addition, **`CVT`** can extract intermediate product messages from either the product database linear buffer (if warehoused) or the individual linear buffer. The basic form of the product extraction commands are:

```
cvt msg X extract    Extracts the contents of message "X" from the
                          product database linear buffer (default) into the
                          file $HOME/graphic/buffer.out (binary output)
cvt msg X hexdump    Extracts the contents of message "X" from the
                          product database linear buffer (default) into the
                          file $HOME/graphic/hexdump.out (ASCII dump in Hex)
```

The default output file is **$HOME/graphic/buffer.out** for the **extract** command and **$HOME/graphic/hexdump.out** for the **hexdump** command.

The basic extraction commands use the product database linear buffer (**$ORPGDIR/pdist/product_data_base.lb**) as the default input source.

- Redirecting Extraction input and output

The **-in** and **-out** options can be used to redirect the input and output from their default locations.

```
-in LBNAME          Extracts contents of message "X" from the linear buffer
                         $ORPGDIR/LBNAME rather than the product database.
                         (path provided by LBNAME is relative to $ORPGDIR)

-out FILENAME       Outputs the results of extract / hexdump to a file named
                         $HOME/FILENAME rather than the default filename.
                         (path provided by FILENAME is relative to $HOME)
```

The default extraction includes a 96 byte header created by the ORPG. To remove this "Pre-ICD" header, use the following flag:

```
-nohdr              Causes the extract command to skip over the first 96 byte
                         header (extracts just the ICD compatible product)

-nodc               CVT automatically decompresses compressed products. This
                         option skips decompression during extraction.
```

The complete form of the product extraction commands are (parameters within brackets **[]** are optional):

```
cvt msg X extract [-nohdr] [-nodc] [-in LBNAME] [-out FILENAME]
cvt msg X hexdump [-nohdr] [-nodc] [-in LBNAME] [-out FILENAME]
```

Part C. Examples of Use of Parameters

Show product summary information for message 3

```
cvt msg 3 summary
```

Override a failure of the consistency checks (either ICD format or block length) by forcing the output of the **fullhdr** command

```
cvt msg 3 -f fullhdr
```

Loading a binary file:

Correctly load a product binary file having no internal 96-byte header. This is the normal form of a product binary file received by an external user (AWIPS or the **nbtcp** tool).

```
cvt load prod_filename -nohdr
```

Correctly load a product binary file obtained from the NCDC web site. This file includes a WMO header to be stripped off.

```
cvt load prod_filename -wmohdr
```

Extraction Commands:

Extract a hexadecimal output of message 3 from the product database linear buffer (default) to the file **$HOME/graphic/x.out**

```
cvt msg 3 hexdump -out graphic/x.out
```

Extract a binary output of message 3 from the linear buffer **$ORPGDIR/base/bref19.lb** to the default location

```
cvt msg 3 extract -in base/bref19.lb
```

Radial and Raster Products:

Show the data (in default/decimal format) from the 14th radial in the buffer of message 3

```
cvt msg 3 radial 14
```

Show the data (in default/decimal format) from radials ranging from 200 to 202 degrees inclusive.

```
cvt msg 3 radial 200 202 deg
```

Show all radial data (in default/decimal format) from message 3

```
cvt msg 3 radial all
```

Show all radial data (in run length encoded hexadecimal format) from message 3

```
cvt msg 3 radial all rle
```

Show radial data from 350 degrees to 2 degrees, inclusive, from message 3

```
cvt msg 3 radial 350 2 deg
```

Show radial data from 1 degree to 5 degrees, inclusive, using the run length encoded format from message 3

```
cvt msg 3 radial 1 5 deg rle
```

Show all radial data in the first data layer in the **BSCAN** format from message 3

```
cvt msg 3 layer 1 radial all bscan
```

Show all data from raster row 3 in default/decimal format from message 10

```
cvt msg 10 row 3
```

Show all data from raster rows 10 through 20 inclusive in default/decimal format from message 10

```
cvt msg 10 row 10 20
```

Show all raster data in default/decimal format from message 10

```
cvt msg 10 row all
```

Show all raster data in run length encoded hexadecimal format from rows 200 through 210 inclusive in message 10

```
cvt msg 10 row 200 210 rle
```

Show all raster data (from all rows) in the run length encoded hexadecimal format from layer 1 of message 10

```
cvt msg 10 layer 1 row all rle
```

Generic Products:

List all of the components contained in a generic product. This list includes an index which can be used to select a single component for display.

```
cvt msg 12 generic list_all
```

Print the generic product header and the component at index 5 in the component array.

```
cvt msg 12 generic print 5
```

Print all radials in the generic radial component and decode the data levels using a user supplied configuration file. The file must be in the **~/.cvt/** directory. The product ID is 176 which requires a filename of decode_params.176. If this file does not exist, the decoding can continue since this product is pre-configured in CVT.

```
cvt load Sample_176_DPR -nohdr radial fdecode
```

Part D. Illustrations of CODEview Text output

1. Inventory of product-specific linear buffer

Command: **cvt i base/bref19**

This command lists those products having a message in the selected single product linear buffer. The **MSG#** refers to the message sequence number within the product database linear buffer that contains this individual product (this is not the internal Message ID). If the **LBMSGLEN** is 96 bytes, the product is being stored in the product database linear buffer.

Note: The product-specific linear buffers are configured to contain references a specific number (in this case 40) of the most recent products (of that type) produced. The main data base retains products on a time-configured basis.

The output was greatly truncated for this illustration.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

      ORPG Products Linear Buffer Inventory Utility
full Linear Buffer Path:  /home/cd11r1_2/data/base/bref19.lb

 Inventory of Linear Buffer: base/bref19.lb
  MSG#  LBMSGLEN LBMSGNUM PRODLEN  VOLNUM   ELEV   VOL DATE    TIME
    1    000096    3264    018160    022     02   1999/05/04 00:12:23
    2    000096    3298    017884    022     03   1999/05/04 00:12:23
    3    000096    3312    016574    022     04   1999/05/04 00:12:23
    4    000096    3326    015944    022     05   1999/05/04 00:12:23


   26    000096    4165    016966    027     02   1999/05/04 00:37:11
   27    000096    4202    016432    027     03   1999/05/04 00:37:11
   28    000096    4215    015354    027     04   1999/05/04 00:37:11
   29    000096    4226    014720    027     05   1999/05/04 00:37:11

NOTE: THE MESSAGE LIST IS NOT SORTED, internal message order is used.
Key:
  MSG#= The message sequence number as listed within the specific linear buffer
  LBMSGLEN=Linear Buffer Message Length
  LBMSGNUM=Message Sequence Number of this product in the main database
  PRODLEN=Product Message Length (includes 96 byte internal header)
```

```
  VOLNUM=Volume Number associated with the product
  ELEV= The Elevation Index associated with the product


    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.
program complete
```

2. Inventory of product database linear buffer

Command: **cvt i**

Shows the inventory listing of the main product database. Unlike CVG, CVT does not sort the product database listing. The list is ordered by the internal message sequence number, **MSG#** with expired messages removed from the list. The identity of the product is determined by the linear buffer number, **LBID**, also referred to as "product id". This id can be used to find the product name and other attributes defined in the **product_attr_table** configuration file.

The output is an example of using the variable **CVT_DB** to access an alternate product database file named **prod_db_b7v1_most_wrap_linux.lb**. This overrides the value of the environmental variable **ORPG_PRODUCTS_DATABASE**.

Message Sequence Number 1 is not displayed. The first message in the product database linear buffer is an internal message and does not contain a product.

Note: The output was greatly truncated for this illustration. This database has reached maximum capacity of 7500 messages. More recent messages (volume 34 and up) are replacing older product messages. Expired product messages are not included in the list as indicated by the break in message sequence numbers in the center of the list (for example the large gap at the 'wrap point' having no messages between 2879 - 6046).

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using an alternate product database:
        /code_data/products/prod_db_b7v1_most_wrap_linux.lb
     defined by the CVT_DB variable


      ORPG Products Database Inventory Utility
Inventory of the Product Database Linear Buffer CVT PATCH ADD VOL TIME
  MSG#      LBID    PRODLEN   VOLNUM     ELEV    VOL DATE    TIME
     2    ID   15    016130    V 034     E 03   1999/05/04 01:11:56
```

```
     3    ID   99    050030    V 034     E 03   1999/05/04 01:11:56
     4    ID  101    331672    V 034     E 03   1999/05/04 01:11:56
     5    ID   58    015072    V 034     E 03   1999/05/04 01:11:56
     6    ID   57    028848    V 034     E 03   1999/05/04 01:11:56
     7    ID  139    003074    V 034     E 03   1999/05/04 01:11:56
     8    ID  290    002836    V 034     E 03   1999/05/04 01:11:56
     9    ID  105    012704    V 034     E 03   1999/05/04 01:11:56
    10    ID  106    011354    V 034     E 03   1999/05/04 01:11:56



  2867    ID   51    010768    V 045     E 14   1999/05/04 02:06:29
  2868    ID   49    009642    V 045     E 14   1999/05/04 02:06:29
  2869    ID   21    016356    V 045     E 14   1999/05/04 02:06:29
  2870    ID  141    002248    V 045     E 14   1999/05/04 02:06:29
  2871    ID  124    023268    V 045     E 14   1999/05/04 02:06:29
  2872    ID  126    009540    V 045     E 14   1999/05/04 02:06:29
  2873    ID  123    026238    V 045     E 14   1999/05/04 02:06:29
  2874    ID  125    010470    V 045     E 14   1999/05/04 02:06:29
  2875    ID  129    002456    V 045     E 14   1999/05/04 02:06:29
  2876    ID   24    027726    V 045     E 14   1999/05/04 02:06:29
  2877    ID   26    009896    V 045     E 14   1999/05/04 02:06:29
  2878    ID   23    035576    V 045     E 14   1999/05/04 02:06:29
  2879    ID   25    011064    V 045     E 14   1999/05/04 02:06:29
  6046    ID    6    007608    V 028     E 04   1999/05/04 00:42:08
  6047    ID    8    028776    V 028     E 04   1999/05/04 00:42:08
  6048    ID    9    023988    V 028     E 04   1999/05/04 00:42:08
  6050    ID   94    021999    V 028     E 04   1999/05/04 00:42:08
  6051    ID   12    013340    V 028     E 04   1999/05/04 00:42:08
  6052    ID   16    011820    V 028     E 04   1999/05/04 00:42:08
  6053    ID   11    018200    V 028     E 04   1999/05/04 00:42:08
  6054    ID   13    016486    V 028     E 04   1999/05/04 00:42:08
  6055    ID   15    014878    V 028     E 04   1999/05/04 00:42:08
  6056    ID   87    036746    V 028     E 04   1999/05/04 00:42:08
  6057    ID 1991    015468    V 028     E 04   1999/05/04 00:42:08
  6058    ID 1990    086858    V 028     E 04   1999/05/04 00:42:08



  7489    ID 1991    015946    V 034     E 03   1999/05/04 01:11:56
  7490    ID 1990    088510    V 034     E 03   1999/05/04 01:11:56
  7491    ID 1993    000386    V 034     E 03   1999/05/04 01:11:56
  7492    ID 1994    000386    V 034     E 03   1999/05/04 01:11:56
  7493    ID    8    026802    V 034     E 03   1999/05/04 01:11:56
  7494    ID    9    022430    V 034     E 03   1999/05/04 01:11:56
  7495    ID   10    019748    V 034     E 03   1999/05/04 01:11:56
  7496    ID   94    022988    V 034     E 03   1999/05/04 01:11:56
  7497    ID   12    013850    V 034     E 03   1999/05/04 01:11:56
  7498    ID   16    013146    V 034     E 03   1999/05/04 01:11:56
  7499    ID   11    018922    V 034     E 03   1999/05/04 01:11:56
  7500    ID   13    016670    V 034     E 03   1999/05/04 01:11:56

NOTE: THE PRODUCT LIST IS NOT SORTED, internal message order is used.
Key:
  MSG#= Message Sequence Number within the Products Database Linear Buffer
  LBID=Linear Buffer ID associated with each product
    Use this value to find the product name within the configuration
    file: product_attr_table
  PRODLEN=Product Message Length (includes 96 byte internal header)
```

```
  VOLNUM=Volume Number associated with the product
  ELEV= The Elevation Index associated with the product


    Using an alternate product database:
        /code_data/products/prod_db_b7v1_most_wrap_linux.lb
     defined by the CVT_DB variable

program complete
```

3. Search of product database

Command: **cvt slb L**

Searches the product database and returns a listing for all products of a specified linear buffer id. This example shows the listing for all Base Reflectivity products, BREF19, buffer id 2, currently in the database.

This output is an example of using the account's configured database file via the value of the **ORPG_PRODUCTS_DATABASE** environmental variable.

Note: The output was truncated for this illustration.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

      ORPG Products Database Search Utility: LB ID=2
 Search was a Success!
  MSG#      LBID    PRODLEN   VOLNUM     ELEV    VOL DATE    TIME
     3    ID    2    012846    V 001     E 01   1999/05/03 22:19:08
    23    ID    2    010198    V 001     E 02   1999/05/03 22:19:08
    57    ID    2    010300    V 001     E 03   1999/05/03 22:19:08
    71    ID    2    010126    V 001     E 04   1999/05/03 22:19:08
    84    ID    2    009862    V 001     E 05   1999/05/03 22:19:08
   187    ID    2    012956    V 002     E 01   1999/05/03 22:24:08
   204    ID    2    010450    V 002     E 02   1999/05/03 22:24:08
   242    ID    2    010490    V 002     E 03   1999/05/03 22:24:08
   259    ID    2    010376    V 002     E 04   1999/05/03 22:24:08
   268    ID    2    010140    V 002     E 05   1999/05/03 22:24:08
   364    ID    2    012916    V 003     E 01   1999/05/03 22:29:06
   383    ID    2    010524    V 003     E 02   1999/05/03 22:29:06
   417    ID    2    010702    V 003     E 03   1999/05/03 22:29:06
   433    ID    2    010624    V 003     E 04   1999/05/03 22:29:06
   446    ID    2    010236    V 003     E 05   1999/05/03 22:29:06
```

```
  4018    ID    2    016694    V 026     E 03   1999/05/04 00:32:14
  4033    ID    2    015560    V 026     E 04   1999/05/04 00:32:14
  4046    ID    2    014992    V 026     E 05   1999/05/04 00:32:14
  4145    ID    2    017610    V 027     E 01   1999/05/04 00:37:11
  4165    ID    2    016966    V 027     E 02   1999/05/04 00:37:11
  4202    ID    2    016432    V 027     E 03   1999/05/04 00:37:11
  4215    ID    2    015354    V 027     E 04   1999/05/04 00:37:11
  4226    ID    2    014720    V 027     E 05   1999/05/04 00:37:11

NOTE: THE PRODUCT LIST IS NOT SORTED, internal message order is used.
Key:
  MSG#=Message Sequence Number within the Products Database Linear Buffer
  LBID=Linear Buffer ID associated with this product
    Use this value to find the product name within the configuration
    file: product_attr_table
  PRODLEN=Product Message Length (includes 96 byte internal header)
  VOLNUM=Volume Number associated with the product
  ELEV= The Elevation Index associated with the product


    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.
program complete
```

4. Accomplish a verbose product consistency check

Command: **cvt msg X check**

Accomplishes a validation of portions of an ICD product and provides a verbose output. The number of blocks, block offsets, product message length, block lengths, layer / page lengths are all compared and any inconsistencies are reported.

The following is the result of a successful check on a product that had one layer in the symbology block and three pages in the GAB.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=4323
-> LB Message Sequence Number=1260
```

```
-> Product Info: LBuffer# 025 MSGLEN 007174 VOLNUM 08 ELEV 14
Binary File appears to contain an ICD Formatted Product

ENTERING PRODUCT CONSISTENCY CHECK MODULE

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---
    Product has 4 blocks:  the MHB and the PDB plus
    the Symbology Block   the GAB

--- Checking block offsets with block identifiers ---
    Symbology Block Offset is 60 halfwords (120 bytes)
    GAB Offset is 2703 halfwords (5406 bytes)

--- Checking Symbology Block layer lengths ---
    Symbology Block Length is 5286.
    Symbology block has 1 layers.
    Layer  1 Length = 5270 bytes

--- Checking GAB Page lengths ---
    GAB Length is 1672.
    Number of GAB pages is 3
    Page  1 Length = 550 bytes
    Page  2 Length = 550 bytes
    Page  3 Length = 550 bytes

--- Comparing block lengths with product message length.
    Product Message Length is 7078 bytes
    Standard headers (MHB & PDB): 120 bytes
    Symbology Block Length is 5286 bytes
    GAB Length is 1672 bytes

Finished Consistency Check

program complete
```

5. Display product summary

Command: **cvt msg X summary**

Shows the summary information about the product in message X in the database. This example uses the DHR Digital Hybrid Scan Reflectivity product associated with linear buffer 57, PUP product code 32. The product contains two layers in the symbology block. The product was compressed in the database and automatically decompressed by CVT. There is no graphic alphanumeric block (GAB) or tabular alphanumeric block (TAB).

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3
```

```
CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=1497
-> LB Message Sequence Number=338
-> Product Info: LBuffer# 057 MSGLEN 041405 VOLNUM 03 ELEV 02
Binary File appears to contain an ICD Formatted Product
Decompressing Product
Uncompressed message length is 85668 (excluding 96 byte internal header)

ENTERING DISPLAY SUMMARY INFO MODULE
Binary File appears to contain an ICD Formatted Product

PRODUCT SUMMARY INFORMATION
---------------------------
Message Code:                                     32
Linear Buffer ID:                                 57
Volume Scan Start Time                            Mar 13, 1993
                                                  09:36:40  (UTC)
Total Product Length (without 96 byte header)     85668
Volume Scan Sequence Number                       3
Elevation Count                                   0
Elevation Index                                   2
Weather Mode                                      2
VCP Number                                        11

 symbology offset = 60 (120 bytes)
Symbology Block Information:
  Block Length = 85548 bytes
  Number of Layers = 2
  Layer  1 Length=84974 bytes
           First Packet Code=10 Hex or 16 decimal
  Layer  2 Length=552 bytes
           First Packet Code=1 Hex or 1 decimal
Graphic Alphanumeric Block is NOT available
Tabular Alphanumeric Block is NOT available
program complete
```

6. Display Pre-ICD product header

Command: **cvt msg X hdr**

Displays the Pre-ICD header for product message X. This data is not part of the product that is distributed to users. In this example, the product is Base Reflectivity, BREF19, product id 2. The first part of the Pre-ICD header is the ORPG internal product generation message which includes the 6 product request parameters. Note that unused product dependent parameters are coded with a special value of PARAM_UNUSED or -32768 decimal.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=1633
-> LB Message Sequence Number=3
-> Product Info: LBuffer# 002 MSGLEN 027178 VOLNUM 01 ELEV 01
Binary File appears to contain an ICD Formatted Product


*******************************************************
**** ORPG INTERNAL PRODUCT HEADER (96 total bytes) ****
*******************************************************
        ------------------  Product Generation Message (52 Bytes) ---------------
Halfword#                                                Decimal       Hexadecimal
01      Product ID (linear buffer number)                2          or 0x0002
02      Replay or Realtime Product                       1          or 0x0001
03-04   Linear Buffer message ID                         0          or 0x00000000
05-06   Generation Time                                  1242244149 or 0x4A0B2435
        Decoded Generation Time:        May 13, 2009
                                        19:49:09
07-08   Volume Scan Start Time                           732014782  or 0x2BA1A8BE
        Decoded Scan Start Time:        Mar 13, 1993
                                        09:26:22
09-10   Total Product Length (includes 96 byte header)  27178      or 0x00006A2A
11      Product Request Number                           0          or 0x0000
12      Elevation Index                                  1          or 0x0001
13-14   Volume Scan Sequence Number                      1          or 0x00000001

                                            Signed    Unsigned        Hex
15      Request Product Parameter 1         -32768       32768     or 0x8000
16      Request Product Parameter 2         -32768       32768     or 0x8000
17      Request Product Parameter 3              5           5     or 0x0005
18      Request Product Parameter 4         -32768       32768     or 0x8000
19      Request Product Parameter 5         -32768       32768     or 0x8000
20      Request Product Parameter 6         -32768       32768     or 0x8000

21      Response Product Parameter 1        -32768       32768     or 0x8000
22      Response Product Parameter 2        -32768       32768     or 0x8000
23      Response Product Parameter 3             5           5     or 0x0005
24      Response Product Parameter 4        -32768       32768     or 0x8000
25      Response Product Parameter 5        -32768       32768     or 0x8000
26      Response Product Parameter 6        -32768       32768     or 0x8000

        ------------------  Product Header Message (44 Bytes) -------------------
Halfword#                                                Decimal       Hexadecimal
26-28   Elevation Time                                   732014782  or 0x2BA1A8BE
        Decoded Elevation Time:         Mar 13, 1993
                                        09:26:22
```

```
29      Elevation Count                                 1          or 0x0001
30      Archive 3 Flag                                  0          or 0x0000
31      Base Data Status                                2          or 0x0002
32      Spare                                           0          or 0x0000
33-34   Spot Blank Bitmap                               0          or 0x00000000
35      Weather Mode                                    2          or 0x0002
36      VCP Number                                      11         or 0x000B
37-38   Compression Method                              0          or 0x00000000
39-40   Uncompressed Size (includes 96-byte header)     27178      or 0x00006A2A
41-42   Reserved Word 1                                 0          or 0x00000000
43-44   Reserved Word 2                                 0          or 0x00000000
45-46   Reserved Word 3                                 0          or 0x00000000
47-48   Reserved Word 4                                 0          or 0x00000000

================================================================================
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

program complete
```

7. Display all header information

Command: **cvt msg X fullhdr**

Shows all header information for the specified product message. The first two sections, named Product Generation Message and Product Header message are the Pre-ICD header. The Next three sections, ORPG Message Header Block, ORPG Product Description Block, and Symbology Block Header, are part of the ICD product distributed to users.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
```

```
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=1726
-> LB Message Sequence Number=2
-> Product Info: LBuffer# 003 MSGLEN 018442 VOLNUM 01 ELEV 01
Binary File appears to contain an ICD Formatted Product


*******************************************************
**** ORPG INTERNAL PRODUCT HEADER (96 total bytes) ****
*******************************************************
        ------------------  Product Generation Message (52 Bytes) ---------------
Halfword#                                                Decimal       Hexadecimal
01      Product ID (linear buffer number)                3          or 0x0003
02      Replay or Realtime Product                       1          or 0x0001
03-04   Linear Buffer message ID                         0          or 0x00000000
05-06   Generation Time                                  1242244149 or 0x4A0B2435
        Decoded Generation Time:        May 13, 2009
                                        19:49:09
07-08   Volume Scan Start Time                           732014782  or 0x2BA1A8BE
        Decoded Scan Start Time:        Mar 13, 1993
                                        09:26:22
09-10   Total Product Length (includes 96 byte header)  18442      or 0x0000480A
11      Product Request Number                           0          or 0x0000
12      Elevation Index                                  1          or 0x0001
13-14   Volume Scan Sequence Number                      1          or 0x00000001

                                            Signed    Unsigned        Hex
15      Request Product Parameter 1         -32768       32768     or 0x8000
16      Request Product Parameter 2         -32768       32768     or 0x8000
17      Request Product Parameter 3              5           5     or 0x0005
18      Request Product Parameter 4         -32768       32768     or 0x8000
19      Request Product Parameter 5         -32768       32768     or 0x8000
20      Request Product Parameter 6         -32768       32768     or 0x8000

21      Response Product Parameter 1        -32768       32768     or 0x8000
22      Response Product Parameter 2        -32768       32768     or 0x8000
23      Response Product Parameter 3             5           5     or 0x0005
24      Response Product Parameter 4        -32768       32768     or 0x8000
25      Response Product Parameter 5        -32768       32768     or 0x8000
26      Response Product Parameter 6        -32768       32768     or 0x8000

        ------------------  Product Header Message (44 Bytes) -------------------
Halfword#                                                Decimal       Hexadecimal
26-28   Elevation Time                                   732014782  or 0x2BA1A8BE
        Decoded Elevation Time:         Mar 13, 1993
                                        09:26:22
29      Elevation Count                                  1          or 0x0001
30      Archive 3 Flag                                   0          or 0x0000
31      Base Data Status                                 2          or 0x0002
32      Spare                                            0          or 0x0000
33-34   Spot Blank Bitmap                                0          or 0x00000000
35      Weather Mode                                     2          or 0x0002
36      VCP Number                                       11         or 0x000B
37-38   Compression Method                               0          or 0x00000000
39-40   Uncompressed Size (includes 96-byte header)      18442      or 0x0000480A
41-42   Reserved Word 1                                  0          or 0x00000000
43-44   Reserved Word 2                                  0          or 0x00000000
```

```
45-46   Reserved Word 3                                  0          or 0x00000000
47-48   Reserved Word 4                                  0          or 0x00000000

==================================================================================
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check



*******************************************************
******** ORPG MESSAGE HEADER BLOCK (18 bytes) ********
*******************************************************
        ---------------------  ORPG Message Header Block  -----------------------
Halfword#                                                Decimal       Hexadecimal
01      NEXRAD Message Code                              16         or 0x0010
02      Date of Message                                  14378      or 0x382A
        Decoded Date of Message:        May 13, 2009
03-04   Time of Message                                  71342      or 0x000116AE
        Decoded Time of Message:        19:49:02
05-06   Product Size in Bytes (excludes internal header)18346      or 0x000047AA
07      ID of the Source                                 302        or 0x012E
08      ID of the Receiver                               0          or 0x0000
09      Number of Blocks                                 3          or 0x0003


*******************************************************
***** ORPG PRODUCT DESCRIPTION BLOCK (102 bytes) *****
*******************************************************
        ------------------ORPG Product Description Block  -----------------------
Halfword#                                                Decimal       Hexadecimal
10      PDB divider                                      -1         or 0xFFFF
11-12   Radar Latitude                                   28113      or 0x00006DD1
        Decoded Radar Latitude:           28.113 N
13-14   Radar Longitude                                  -80654     or 0xFFFEC4F2
        Decoded Radar Longitude:         -80.654 W
15      Radar Height (MSL)                               116        or 0x0074
16      Internal Product Code                            16         or 0x0010
17      Operational Weather Mode                         2          or 0x0002
18      VCP Number                                       11         or 000B
19      Request Sequence Number                          0          or 0x0000
20      Volume Scan Number                               1          or 0x0001

21      Volume Scan Date                                 8473       or 0x2119
        Decoded Volume Scan Date:       Mar 13, 1993
25-26   Volume Scan Start Time                           33982      or 0x000084BE
        Decoded Volume Scan Start Time: 09:26:22
```

```
24      Product Generation Date                          14378      or 0x382A
        Decoded Product Generation Date: May 13, 2009
25-26   Product Generation Time                          71342      or 0x000116AE
        Decoded Product Generation Time: 19:49:02

                                            Signed    Unsigned        Hex
27      Product Dependent Parameter 1            0           0     or 0x0000
28      Product Dependent Parameter 2            0           0     or 0x0000
                                                        Decimal       Hexadecimal
29      Volume Elevation Index                          1           or 0x0001
                                            Signed    Unsigned        Hex
30      Product Dependent Parameter 3            5           5     or 0x0005

31      Data Level Threshold 1              -32766       32770     or 0x8002
32      Data Level Threshold 2                   5           5     or 0x0005
33      Data Level Threshold 3                  18          18     or 0x0012
34      Data Level Threshold 4                  30          30     or 0x001E
35      Data Level Threshold 5                  41          41     or 0x0029
36      Data Level Threshold 6                  46          46     or 0x002E
37      Data Level Threshold 7                  50          50     or 0x0032
38      Data Level Threshold 8                  57          57     or 0x0039
39      Data Level Threshold 9              -32768       32768     or 0x8000
40      Data Level Threshold 10             -32768       32768     or 0x8000
41      Data Level Threshold 11             -32768       32768     or 0x8000
42      Data Level Threshold 12             -32768       32768     or 0x8000
43      Data Level Threshold 13             -32768       32768     or 0x8000
44      Data Level Threshold 14             -32768       32768     or 0x8000
45      Data Level Threshold 15             -32768       32768     or 0x8000
46      Data Level Threshold 16             -32768       32768     or 0x8000

47      Product Dependent Parameter 4           59          59     or 0x003B
48      Product Dependent Parameter 5            0           0     or 0x0000
49      Product Dependent Parameter 6            0           0     or 0x0000
50      Product Dependent Parameter 7            0           0     or 0x0000
51      Product Dependent Parameter 8        16697       16697     or 0x4139
52      Product Dependent Parameter 9       -13444       52092     or 0xCB7C
53      Product Dependent Parameter 10           0           0     or 0x0000

                                                        Decimal       Hexadecimal
54      Product Version (if available)                  0           or 0x0000
        Spot Blanking Bit                               0           or 0x0000

55-56   Symbology Offset (halfwords)                    60          or 0x0000003C
57-58   Graphic Block Offset (halfwords)                0           or 0x00000000
59-60   Tabular Block Offset (halfwords)                0           or 0x00000000


**************************************************
******** SYMBOLOGY BLOCK HEADER (10 bytes) ********
**************************************************
        ---------------------- SYMBOLOGY BLOCK HEADER  ------------------------
                                                        Decimal       Hexadecimal
61      divider                                         -1          or 0xFFFF
62      blockID                                         1           or 0x0001
63-64   blocklength                                     18226       or 0x00004732
65      num layers                                      1           or 0x0001

program complete
```

8. Display selected radials in Decimal notation

Command: **cvt msg X radial 2 3 deg**

Displays radials from 2 degrees to 3 degrees inclusive of product "X". This product is Base Reflectivity, BREF19, linear buffer id 2. The raw 16-level reflectivity data are displayed (levels 0 - 15) as extracted from the run-length encoded (RLE) data in the product.

***Note: CVT** does not support use of the command line radial modifiers **bscan** and **rle** with the generic radial component.*

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=1855
-> LB Message Sequence Number=3
-> Product Info: LBuffer# 002 MSGLEN 027178 VOLNUM 01 ELEV 01
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Set Processing for All Layers

packet code af1f found

---------------- Decoding Packet AF1Fx -----------------
Index of First Range Bin:                        0
Number of Range Bins:                            230
I center of sweep:                               256
J center of sweep:                               280
Scale Factor:                                    999
```

```
Number of Radials                                366

Radials Unpacked to show all Data Values
AFIF Decimal Output - Radial 279  Angle: 002.6  Delta:  1.0
00 06 06 07 06 07 06 06 06 06 06 06 05 06 06 06 06 06 07 07
07 07 07 07 08 07 07 07 07 07 07 07 07 07 07 07 07 07 08 07
07 08 08 08 07 08 07 07 07 08 08 08 08 07 07 07 07 07 07 07
07 07 07 08 07 07 07 07 07 07 08 08 07 07 08 07 07 07 08 07
07 07 07 07 07 07 06 07 07 06 06 06 06 05 05 05 06 06 06 06
06 06 05 05 05 05 05 06 05 06 06 06 05 05 06 06 07 07 07 07
07 07 07 07 07 07 07 07 08 07 07 07 07 08 08 08 08 07 07 07
07 07 07 07 07 07 07 07 07 07 08 07 07 07 07 07 07 07 07 07
07 07 07 08 07 07 07 07 07 07 07 07 07 07 07 07 08 07 07 07
07 07 07 06 07 07 07 08 08 08 07 07 07 07 08 08 08 08 08 08
08 08 08 08 08 08 08 08 08 08 08 08 07 07 08 08 07 07 07 07
06 06 06 06 06 05 05 05 05 05
AFIF Decimal Output - Radial 280  Angle: 003.6  Delta:  0.9
00 06 06 06 07 07 06 06 06 06 06 06 06 06 06 06 06 07 06 06
07 07 07 07 08 07 07 07 07 07 07 07 07 07 07 07 07 08 08 07
08 08 08 08 08 08 07 07 07 08 08 08 08 07 07 07 07 07 07 07
07 07 08 07 07 07 07 07 07 07 07 07 07 07 08 08 08 08 07 07
07 07 07 07 07 07 06 07 07 06 06 06 06 06 06 06 06 07 07 07
06 05 05 05 04 04 05 05 05 05 05 05 05 06 06 07 07 06 07 08
08 07 07 07 07 07 07 06 07 07 07 07 07 07 07 07 07 08 07 07
07 07 07 07 08 07 07 07 08 08 07 07 07 08 08 08 08 07 08 08
07 08 08 08 08 07 08 08 07 07 07 07 08 07 07 07 07 07 07 07
07 08 08 07 08 07 08 08 08 08 08 08 08 08 08 08 08 08 08 07
07 08 08 08 07 08 07 07 07 07 07 07 06 07 07 08 08 07 07 08
07 07 07 06 06 06 06 05 06 06
program complete
```

9. Display all radials in BSCAN format

Command: **`cvt msg X radial bscan`**

Displays all radial data in decimal format using the BSCAN output structure. This output includes the two radials in the Illustration 7 output in a tabular format.

All radials should be selected when using the BSCAN option.

The output is not formatted correctly when displayed directly on the screen. The output should be redirected to a file and the file opened with an editor.

Note: *The radial modifier* **`bscan`** *is not supported with the generic radial component.*

Note: The output was greatly truncated for this illustration.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3
```

```
    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=2633
-> LB Message Sequence Number=3
-> Product Info: LBuffer# 002 MSGLEN 027178 VOLNUM 01 ELEV 01
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Set Processing for All Layers

packet code af1f found

---------------- Decoding Packet AF1Fx -----------------
Index of First Range Bin:           0
Number of Range Bins:               230
I center of sweep:              256
J center of sweep:              280
Scale Factor:                   999
Number of Radials               366

Radials Unpacked to show all Data Values
BSCAN Format Output Selected
CVT using variable CV_ORPG_BUILD to set ORPG Build to 10
Begin BSCAN Generator
number of passes to print 4


  AZM      1    2    3    4    5    6    7    8    9   10   11   12   13   14
-------------------------------------------------------------------------------
  0.6      0    5    5    5    6    7    6    6    7    7    7    7    7    7
  1.6      0    6    6    7    6    7    6    6    6    6    6    5    6    6
  2.6      0    6    6    7    6    7    6    6    6    6    6    6    5    6
  3.6      0    6    6    6    7    7    7    6    6    7    7    7    7    8
  4.5      0    5    6    6    7    6    6    6    6    6    6    5    6    6
  5.5      0    6    7    6    6    7    6    6    6    6    5    6    6    5
  6.5      0    0    7    6    7    6    6    6    6    6    6    6    5    6
  7.5      0    0    6    7    7    6    6    6    5    6    5    5    6    6
```

```
   AZM     51   52   53   54   55   56   57   58   59   60   61   62   63   64
-------------------------------------------------------------------------------
  0.6       7    7    7    6    7    7    6    6    6    6    6    6    5    6
  1.6       8    8    8    8    7    7    7    8    8    7    8    8    6    6
  2.6       7    8    7    7    7    8    7    7    7    7    7    7    7    6
  3.6       5    5    5    5    6    6    7    7    6    7    8    8    7    7
  4.5       7    7    7    7    6    7    7    6    6    6    6    6    5    6
  5.5       7    6    6    7    7    6    6    6    6    6    6    6    7    6
  6.5       8    7    7    6    6    6    6    6    6    6    6    6    6    7
  7.5       7    7    6    6    6    6    5    6    6    5    6    6    6    6



  final length=30


  AZM    201  202  203  204  205  206  207  208  209  210  211  212  213  214
-------------------------------------------------------------------------------
  0.6       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1.6       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  2.6       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  3.6       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  4.5       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  5.5       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  6.5       0    0    0    0    0    0    0    0    0    0    0    0    0    0
  7.5       0    0    0    0    0    0    0    0    0    0    0    0    0    0


program complete
```

10. Display selected rows of a raster product in RLE values

Command: **`cvt msg X row 120 121 rle`**

Display the second and third rows of the raster data layer in run length encoded Hexadecimal format. The product is Composite Reflectivity, CRP38, linear buffer 25.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=4323
-> LB Message Sequence Number=1080
-> Product Info: LBuffer# 025 MSGLEN 007052 VOLNUM 07 ELEV 14
```

```
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Checking GAB Page lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Set Processing for All Layers

packet code ba07 found

---------------- Decoding Packet BA07x BA0Fx------------
Packet Type (Legacy internal code)              8000
Packet Type (Legacy internal code)              c0
I Coordinate Start                              1
J Coordinate Start                              1
X Scale INT                                     2
X Scale Fractional (N/A - for Legacy PUP)       0
Y Scale INT                                     2
Y Scale Fractional (N/A - for Legacy PUP)       0
Number of Rows                                  232
Packing Descriptor                              2

Rows Left in Run-Length Encoded Format
packet_BA07 - number of columns is 232
BA07 RLE Output - Row: 120
f0 f0 f0 f0 80 21 52 23 16 2a 19 18 17 26 14 13 16 17 16 24
16 17 1a 37 18 29 16 15 14 43 12 53 24 c3 12 13 32 21 f0 f0
f0 f0 f0 f0 70 0
BA07 RLE Output - Row: 121
f0 f0 f0 f0 a0 62 14 18 19 1a 19 17 35 24 33 16 15 17 16 19
16 15 29 18 17 16 25 14 33 22 63 14 e3 22 11 f0 f0 f0 f0 f0
f0 80
program complete
```

11. Display an Overlay Product Containing Special Symbol Packets

Command: **`cvt msg X layer 1`**

Display the data packets in the first layer of a product. This example is an overlay product which typically contain the special symbols and labels in the first layer of the symbology block. Overlays

usually have a GAB and TAB providing data about each depicted feature. The following product contains 3 TVS features (packet 20) and 3 Storm ID Labels (packet 15).

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=4323
-> LB Message Sequence Number=1426
-> Product Info: LBuffer# 143 MSGLEN 001758 VOLNUM 09 ELEV 14
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Checking GAB Page lengths ---

--- Checking TAB Page lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Set Processing ONLY for Layer Number 1

packet code 20 found

Packet 20: Generic Point Feature
Packet 20: Length=   8  Number Included=1
  I Pos: -295  J Pos: -132  Feature Type: 7  Attribute: 0

packet code 15 found

Packet 15: Storm ID Data
Packet 15: Length=   6  Number Included=1
  I Pos: -295  J Pos: -132  Storm ID: B0

packet code 20 found

Packet 20: Generic Point Feature
Packet 20: Length=   8  Number Included=1
  I Pos: -338  J Pos:   40  Feature Type: 7  Attribute: 0
```

```
packet code 15 found

Packet 15: Storm ID Data
Packet 15: Length=   6  Number Included=1
  I Pos: -338  J Pos:   40  Storm ID: L0

packet code 20 found

Packet 20: Generic Point Feature
Packet 20: Length=   8  Number Included=1
  I Pos: -323  J Pos:   41  Feature Type: 7  Attribute: 0

packet code 15 found

Packet 15: Storm ID Data
Packet 15: Length=   6  Number Included=1
  I Pos: -323  J Pos:   41  Storm ID: L0

program complete
```

12. Listing of Components in a Generic Product

Command: **cvt msg X generic list_all**

List all components in a Generic Product.. The product is the Mesocyclone Detection Data Array, DMDPROD, linear buffer 149. All components are area components.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=2633
-> LB Message Sequence Number=261
-> Product Info: LBuffer# 149 MSGLEN 005046 VOLNUM 02 ELEV 10
Binary File appears to contain an ICD Formatted Product
Decompressing Product
Uncompressed message length is 64424 (excluding 96 byte internal header)
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---
```

```
--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Generic Flag, Processing for Layer 1

packet code 28 found
Offset value = 234
Packet 28 offset of serial data is 240
Contents of Align HW = 0
Length of Serialized Data = 64280 bytes

product id 149, name: Digital Mesocyclone Detection,
desc: Data array product output of Mesocyclone Detection Algorithm.
    type: 2 (Elevation), prod gen time: 71613
NOTE: Class 1 ICD states all times are 'Unix Time' but some
      products have seconds after midnight for Generation time.
    Decoded Gen time is 19:53:33 (sec after midnight)

    Radar name: KMLB, lat  28.113, lon -80.654, height  35.4
    Vol time 1242293510 (Unix Time), elev time 1242293715 (Unix Time),
    vol number 2, elev number 10
    Decoded Vol time is May 14, 2009  9:31:50
    Decoded Elev time is May 14, 2009  9:35:15

    VCP 11, Op mode 2 (Clear Air), target elev  10.0
    compress type 0, size 0

This Product has 5 parameters
    param[ 0]: id: avg_dir, attrs:
               Name=Average Direction of Tracked Features; Type=float; Value=254.1;
               Units=deg;
    param[ 1]: id: avg_spd, attrs:
               Name=Average Speed of Tracked Features; Type=float; Value= 14.7;
               Units=m/s;
    param[ 2]: id: last_elev_flag, attrs:
               Name=Last Elevation Flag; Type=int; Value=0;Units=;
    param[ 3]: id: elev_angle, attrs:
               Name=Elevation Angle; Type=float; Value= 0.5, 1.5, 2.4, 3.4, 4.3,
                5.3, 6.2, 7.5, 8.7,10.0;Units=deg;
    param[ 4]: id: elev_time, attrs:
               Name=Elevation Time; Type=int; Value=34310,34351,34370,34394,
               34416,34438,34460,34483,34499,34515;Units=s;

This Product has 22 components
     Listing All Components
(Index  0) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  1) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  2) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  3) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  4) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  5) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  6) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  7) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index  8) AREA Component: (3), Geographical Point, Lat Lon Coordinates
```

```
(Index  9) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 10) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 11) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 12) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 13) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 14) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 15) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 16) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 17) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 18) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 19) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 20) AREA Component: (3), Geographical Point, Lat Lon Coordinates
(Index 21) AREA Component: (3), Geographical Point, Lat Lon Coordinates


Packet 28 Complete
program complete
```

13. Display a Selected Component in a Generic Product

Command: **cvt msg X generic print 3 no_pparams no_cparams**

Displays the fourth component in a Generic Product. The product is the Mesocyclone Detection Data Array, DMDPROD, linear buffer 149. The product parameters and the component parameters have been eliminated from the output

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=2633
-> LB Message Sequence Number=261
-> Product Info: LBuffer# 149 MSGLEN 005046 VOLNUM 02 ELEV 10
Binary File appears to contain an ICD Formatted Product
Decompressing Product
Uncompressed message length is 64424 (excluding 96 byte internal header)
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---
```

```
--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Generic Flag, Processing for Layer 1

packet code 28 found
Offset value = 234
Packet 28 offset of serial data is 240
Contents of Align HW = 0
Length of Serialized Data = 64280 bytes

product id 149, name: Digital Mesocyclone Detection,
desc: Data array product output of Mesocyclone Detection Algorithm.
    type: 2 (Elevation), prod gen time: 71613
NOTE: Class 1 ICD states all times are 'Unix Time' but some
      products have seconds after midnight for Generation time.
    Decoded Gen time is 19:53:33 (sec after midnight)

    Radar name: KMLB, lat  28.113, lon -80.654, height  35.4
    Vol time 1242293510 (Unix Time), elev time 1242293715 (Unix Time),
    vol number 2, elev number 10
    Decoded Vol time is May 14, 2009  9:31:50
    Decoded Elev time is May 14, 2009  9:35:15

    VCP 11, Op mode 2 (Clear Air), target elev  10.0
    compress type 0, size 0

This Product has 5 parameters
     Printing of Product Parameters not Selected

This Product has 22 components

(Index  3) AREA Component: (3), Geographical Point, Lat Lon Coordinates

This Component has 37 parameters
     Printing of Component Parameters not Selected

        $ 1 points:
            point[0]: lat =    26.91, lon =   -80.14


Packet 28 Complete
program complete
```

14. Display and Decode Selected Radials of a Generic Radial Product

Command: **`cvt load FNAME -nohdr generic radial 10 pdecode`**

Display radial 10 from a generic radial product.. The product is a test version of the future Digital Precipitation Rate Data, DPRPROD, linear buffer 176. Neither the product nor the radial component have parameters. The data levels were decoded using the Scale-Offset parameters in the product.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10
 filename from relative path is /home/cd11r1_12/DP_PROD/Sample_176_DPR
:loading product headers...
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Checking number of blocks vs. block offset values ---

--- Checking block offsets with block identifiers ---

--- Checking Symbology Block layer lengths ---

--- Comparing block lengths with product message length.

Finished Consistency Check

-> Generic Flag, Processing for Layer 1

packet code 28 found
Offset value = 234
Packet 28 offset of serial data is 240
Contents of Align HW = 0
Length of Serialized Data = 1346632 bytes

product id 176, name: Digital Precipitation Rate (DPR),
desc: Data array product output of QPE RATE algorithm.
    type: 1 (Volume), prod gen time: 59907
NOTE: Class 1 ICD states all times are 'Unix Time' but some
      products have seconds after midnight for Generation time.
    Decoded Gen time is 16:38:27 (sec after midnight)

    Radar name: KCRI, lat  35.238, lon -97.460, height 394.7
    Vol time 1187498596 (Unix Time), elev time 0 (Unix Time),
    vol number 20, elev number 0
    Decoded Vol time is August 19, 2007  4:43:16

    VCP 21, Op mode 3 (Severe Wx), target elev  14.6
    compress type 0, size 0

This Product has 0 parameters

This Product has 1 components
     Printing All Components

(Index  0) RADIAL Component (1): description: Rate Data array product output
    bin_size 250.0, first_range 125.0, 360 radials
```

```
This Component has 0 parameters

  Data Values Decoded using Product Parameters (pdecode)

   NOTE: if the product does not contain the Scale Offset
         parameters the decoded values will be incorrect.

   Product 176, Scale is 1000.000000, Offset is 0.000000
   max level is 65535, n lead flags is 0, n trail flags is 0

Radial 10    begin azimuth   9.5, width  1.0, elevation  0.0, n_bins 920
data attributes: type = ushort; Unit = inches/hour
Data:
   0.006    0.010    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.123    0.088    0.091    0.075    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.113    0.070    0.076    0.104    0.198    0.246    1.783    0.211    0.215    1.783
   1.783    0.116    0.118    0.109    0.098    0.107    0.132    0.118    0.118    0.123
   0.151    0.140    0.103    0.106    0.091    0.127    0.135    0.132    0.091    0.201
   0.193    0.159    0.166    0.246    0.464    0.435    0.303    0.106    0.113    0.124
   0.162    0.166    0.185    0.190    0.121    0.115    0.111    0.117    0.112    0.110
   0.120    0.130    0.189    0.206    0.192    0.164    0.126    0.094    0.115    0.134
   0.151    0.152    0.113    0.104    0.086    0.100    0.112    0.107    0.106    0.106
   0.126    0.138    0.134    0.106    0.065    0.055    0.077    0.101    0.117    0.128
   0.143    0.140    0.103    0.101    0.118    0.115    0.074    0.071    0.079    0.111
   0.087    0.085    0.087    0.093    0.081    0.097    0.113    0.126    0.126    0.121
   0.115    0.108    0.126    0.126    0.129    0.129    0.127    0.119    0.114    0.075
   0.073    0.057    0.072    0.064    0.064    0.088    0.137    0.142    0.130    0.068
   0.067    0.080    0.087    0.091    0.076    0.084    0.076    0.089    0.078    0.061
   0.065    0.073    0.090    0.077    0.096    0.103    0.112    0.091    0.106    0.108
   0.123    0.075    0.063    0.077    0.101    0.118    0.097    0.105    0.073    0.076
   0.093    0.124    0.132    0.097    0.085    0.069    0.060    0.053    0.055    0.050
   0.057    0.054    0.088    0.102    0.075    0.053    0.041    0.075    0.079    0.071
   0.056    0.062    0.064    0.061    0.059    0.054    0.045    0.032    0.043    0.061
   0.104    0.085    0.062    0.050    0.059    0.066    0.080    0.082    0.092    0.063
   0.065    0.048    0.057    0.045    0.055    0.071    0.101    0.096    0.069    0.053
   0.043    0.045    0.062    0.109    0.091    0.067    0.047    0.044    0.039    0.044
   0.040    0.049    0.060    0.065    0.055    0.041    0.031    0.044    0.032    0.035
   0.021    0.021    0.018    0.024    0.025    0.043    0.047    0.060    0.038    0.035
   0.020    0.025    0.027    0.034    0.036    0.035    0.026    0.025    0.019    0.013
   0.009    0.006    0.008    0.009    0.015    0.018    0.021    0.018    0.016    0.013
   0.012    0.015    0.025    0.025    0.021    0.015    0.014    0.017    0.015    0.011
   0.012    0.021    0.027    0.021    0.017    0.022    0.029    0.045    0.031    0.019
   0.019    0.022    0.029    0.015    0.015    0.016    0.026    0.030    0.031    0.020
   0.018    0.024    0.037    0.034    0.026    0.022    0.020    0.019    0.014    0.023
   0.032    0.024    0.014    0.015    0.027    0.026    0.017    0.015    0.019    0.024
   0.021    0.025    0.024    0.030    0.031    0.027    0.021    0.013    0.014    0.021
   0.029    0.031    0.029    0.033    0.044    0.039    0.044    0.036    0.026    0.038
   0.050    0.044    0.040    0.032    0.050    0.047    0.064    0.064    0.051    0.043
   0.033    0.040    0.038    0.044    0.040    0.029    0.020    0.019    0.024    0.014
   0.010    0.011    0.027    0.030    0.027    0.020    0.024    0.030    0.023    0.021
   0.018    0.022    0.028    0.014    0.010    0.007    0.004    0.004    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.005    0.006    0.007    0.006
   0.005    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
   0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
```

```
0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.000    0.007    0.009    0.009    0.010    0.010    0.010
0.011    0.010    0.011    0.010    0.013    0.011    0.013    0.013    0.007    0.006
0.008    0.012    0.010    0.007    0.006    0.008    0.010    0.010    0.011    0.007
0.006    0.006    0.009    0.008    0.008    0.018    0.012    0.012    0.013    0.010
0.012    0.015    0.000    0.000    0.000    0.000    0.000    0.000    0.007    0.008
0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.007    0.007    0.006    0.000    0.000    0.000    0.000    0.000    0.000
0.000    0.000    0.000    0.000
```

Command: **cvt msg X satap**

Display a Stand Alone Tabular Alphanumeric Product. The product is Supplemental Precipitation Data, HYSUPPLE, linear buffer 109. This is a rarely used product structure type.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

    Using the account's product data base:
        /home/cd11r1_12/data/pdist/product_data_base.lb
    defined by the ORPG_PRODUCTS_DATABASE environmental variable
        An alternate data base can be specified with the CVT_DB variable.

*** ORPG DATABASE PRODUCT LOAD UTILITY ***
-> Number of Products Available=4323
-> LB Message Sequence Number=765
-> Product Info: LBuffer# 109 MSGLEN 002930 VOLNUM 05 ELEV 02
Binary File appears to contain an ICD Formatted Product
Binary File appears to contain an ICD Formatted Product

Performing Consistency Check of Product Length with
Block Offsets and Block Lengths

--- Stand Alone Tabular Alphanumeric Product (SATAP) Detected ---

Finished Consistency Check


Stand Alone Tabular Alphanumeric Product
Block Divider =   -1
Number of Pages:  2

Stand Alone Tabular Alphanumeric Product - Page: 1
SUPPLEMENTAL PRECIPITATION DATA - RDA ID   302  05/03/99 22:39

VOLUME COVERAGE PATTERN =  11   MODE = A

          GAGE BIAS APPLIED               -      NO
               BIAS ESTIMATE              -     1.00
               EFFECTIVE # G/R PAIRS      -     0.00
               MEMORY SPAN (HOURS)        -     0.00
               DATE/TIME LAST BIAS UPDATE - 12/31/** 00:00
  TOTAL NO. OF BLOCKAGE BINS REJECTED     -        0
               CLUTTER BINS REJECTED      -     2355
               FINAL BINS SMOOTHED        -        0
   HYBRID SCAN PERCENT BINS FILLED        -    99.76
               HIGHEST ELEV. USED (DEG)   -     1.50
               TOTAL RAIN AREA (KM**2)    -   3682.2

        MISSING PERIOD:  NONE

Stand Alone Tabular Alphanumeric Product - Page: 2
                        GAGE-RADAR MEAN FIELD BIAS TABLE
```

```
LAST BIAS UPDATE TIME:  12/31/** 00:00                    BIAS APPLIED ?   NO

  MEMORY SPAN  | EFFECTIVE NO. |   AVG. GAGE   |   AVG. RADAR  |   MEAN FIELD  |
    (HOURS)    |   G-R PAIRS   |   VALUE (MM)  |   VALUE (MM)  |      BIAS     |
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000
       0.000           0.000           0.000           0.000           0.000

SATAP Message Complete
program complete
```

16. Display CODEview Text version

Command: **cvt version**

Show the version number and contact information for the CODEview Text utility.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

CODEview Text for the WSR-88D Open Systems RPG
----------------------------------------

CODEview Text (CVT) Version 4.4.3
Release Date: January, 2010

Provided to the NWS Office of Science & Technology by
     Noblis, Inc. of Falls Church, Virginia

Techncal Contact:  Brian Klein  brian.klein@noaa.gov

program complete
```

17. Display all defined packet types

Command: **cvt packets**

Display the ICD packet codes with name and a reference to the defined structure.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

=================== ICD Packet Code Listing =======================
From the ICD for RPG to Class 1 User, Document 2620001
Packet                                                      ICD
Code     Enumerated Structure Name                        Figure
-------------------------------------------------------------------
1        TEXT_AND_SPECIAL_SYMBOL_TEXT_NO_VALUE            3 - 8b
2        TEXT_AND_SPECIAL_SYMBOL_SYMBOL_NO_VALUE          3 - 8b
3        MESOCYCLONE_DATA                                 3 - 14
4        WIND_BARB_DATA                                   3 - 13
5        VECTOR_ARROW_DATA                                3 - 12
6        LINKED_VECTOR_NO_VALUE                           3 - 7
7        UNLINKED_VECTOR_NO_VALUE                         3 - 8
8        TEXT_AND_SPECIAL_SYMBOL_TEXT_UNIFORM_VALUE       3 - 8b
9        LINKED_VECTOR_UNIFORM_VALUE                      3 - 7
10       UNLINKED_VECTOR_UNIFORM_VALUE                    3 - 8
11       CORRELATED_SHEAR_MESO (3D)                       3 - 14
12       TVS_DATA                                         3 - 14
13       POSITIVE_HAIL_DATA                               3 - 14
14       PROBABLE_HAIL_DATA                               3 - 14
15       STORM_ID_DATA                                    3 - 14
16       DIGITAL_RADIAL_DATA_ARRAY                        3 - 11c
17       DIGITAL_PRECIP_DATA_ARRAY                        3 - 11a
18       PRECIP_RATE_DATA_ARRAY                           3 - 11b
19       HDA_HAIL_DATA                                    3 - 14
20       POINT_FEATURE_DATA                               3 - 14
21       CELL_TREND_DATA                                  3 - 15
22       CELL_TREND_VOLUME_SCAN_TIME                      3 - 15a
23       SCIT_PAST_POSITION_DATA                          3 - 14
24       SCIT_FORECAST_POSITION_DATA                      3 - 14
25       STI_CIRCLE_DATA                                  3 - 14
26       ETVS_DATA                                        3 - 14
27       SuperOb Wind Data Packet                         3 - 15b
28       Generic Data Packet                              3 - 15b
         Supported Components: area, radial, text, table - Appendix E
Hexadecimal Product Codes (must include the trailing 'x')
0802x    CONTOUR_VECTOR_COLOR                             3 - 8a
0E03x    CONTOUR_VECTOR_LINKED                            3 - 8a
3501x    CONTOUR_VECTOR_UNLINKED                          3 - 8a
AF1Fx    RADIAL_DATA_16_LEVELS                            3 - 10
BA07x    RASTER_DATA_7                                    3 - 11
BA0Fx    RASTER_DATA_F                                    3 - 11

program complete
```

18. Display a CVT command quick reference list

Command: **`cvt h`**

Displays the cvt commands with minimal explanation. A more extensive help output consisting of the contents of Part B of this document is provided by the '**cvt help**' command.

```
CVT binary was compiled for the PC Linux Platform.
This version of CVT must be run within an ORPG account.
CODEview Text (CVT) Version 4.4.3

CVT using variable CV_ORPG_BUILD to set ORPG Build to 10

======================= CVT Command Quick Reference ==========================

 More extensive information can be obtained with 'cvt help'

- CODEview Text Informational Commands
  -----------------------------------
cvt help             Listing of CODEview Text command line arguments & options
cvt h                An abbreviated version of 'cvt help'
cvt version          Display version and release date information
cvt packets          Listing of ICD Packet Codes & Definitions

- Linear Buffer Inventory Commands
  --------------------------------
cvt i                Provide an inventory of the main products
                         database linear buffer
cvt i LBNAME         Provide an inventory of the specific product linear
                         buffer with the path 'LBNAME' (under $ORPGDIR)
cvt slb L            Search the products database linear buffer for a
                         specific linear buffer ID 'L' (product ID)
cvt req LBNAME       Show the inventory of one of the request linear
                         buffer with the path 'LBNAME' (under $ORPGDIR)

 ------------------------------------------------------------------------------
                   TWO SOURCES OF PRODUCT MESSAGES:
cvt msg X <arguments>                       Access message 'X' from LB
                                              (product from the database)
cvt load FNAME [-nohdr|-wmohdr] <arguments> Load product from a binary file with
                                               relative/absolute path 'FNAME'

                           CONSISTENCY CHECKS
 Product display can be forced with a failure of either the block length check
 or the ICD product check by using the '-f' option (force load)
      cvt msg X -f <additional command line arguments>
      cvt load FNAME -f <additional command line arguments>
 ------------------------------------------------------------------------------

- Product Information & Header Display Commands
  ---------------------------------------------
cvt msg X summary    Access message 'X' in the products database linear buffer
                         then print a message summary.
cvt msg X            Same as the command: 'cvt msg X summary'
                         (this shortcut does not work with 'cvt load FNAME')

cvt msg X check      Print a verbose product consistency check

cvt msg X hdr        Print the pre-ICD product header
cvt msg X mhb        Print the message header block
cvt msg X pdb        Print the product description block
```

```
cvt msg X sym        Print the symbology block header
cvt msg X fullhdr    Print all header blocks

 ------------------------------------------------------------------------------

- Basic Data Display Commands
  ---------------------------
cvt msg X layer C    Print the contents of layer C
     Command modifiers, 'radial' and 'row' can be used to control the
     output of layers containing radial and raster data.

cvt msg X generic    Print the entire contents of a generic product
     Command modifiers can be used to control the output of generic products.

cvt msg X tab        Print the contents of the Tabular Alphanumeric Block
          tabv       Print the MHB and PDB included within the TAB
cvt msg X gab        Print the contents of the Graphic Alphanumeric Block

cvt msg X satap      Print the contents of the Stand Alone Tabular Alpha Product

- Radial/Raster Data Display Commands
  -------------------------------------
cvt msg X [layer C] radial <modifier list>       for radial products
cvt msg X [layer C] row <modifier list>          for raster products

- Radial/Raster Modifier List Structure
  -----------------------
           [<content selector>]   [<format selector>]    [<decode selector>]
           ---------------------  -------------------  ------------------------
   radial  {A | A B | all} [deg]     [rle | bscan]     {fdecode | pdecode} [D]
      row  {A | A B | all}           [rle]

 ------------------------------------------------------------------------------

- Generic Product Display Commands
  -------------------------------------
cvt msg X generic <generic modifier list>

- Generic Modifier List Structure
  -----------------------
  Any combination of the following optional modifiers in any order:
                               [<component list>]
-------------------------------------------------------------------------------
list_all | list_area | list_rad | list_text | list_table | list_grid | list_event

                               [<component print>]
  ---------------------------------------------------------------------------
  print_area | print_rad | print_text | print_table | print_grid | print_event
  print X

                               [<parameter print>]
                 ---------------------------------------------
                 no_pparams | no_cparams | no_pparams no_cparams

 For radial components:    [<radial content selector>]
                          ----------------------------
                          radial  {A | A B | all} [deg]

 For radial/grid components:   [<decode selector>]
```

```
                              ------------------------
                              {fdecode | pdecode} [D]

 ------------------------------------------------------------------------------

- Product Extraction Commands
  ---------------------------
cvt msg X extract [-nohdr] [-nodc] [-in LBNAME] [-out FILENAME]
cvt msg X hexdump [-nohdr] [-nodc] [-in LBNAME] [-out FILENAME]

 ------------------------------------------------------------------------------
program complete
```

Part E. Linear Buffer Background Information for CODEview Text

Linear buffers

A *linear buffer* is data storage facility that provides the primary mechanism for inter-task communication in the WSR-88D ORPG. Virtually all persistent internal data storage is accomplished via linear buffers. Most linear buffers are of type 'file' where the stored data exist in files which are persistent. For performance purposes, a few linear buffers are configured as 'shared memory' buffers which only contain data while the ORPG is running. All data is stored in a linear buffer in the form of "messages". ORPG service libraries hide the details of reading and writing linear buffer messages. However, the reader of a linear buffer message must know the structure of the data stored in the message.

Product storage

ORPG algorithm tasks output each product type to a unique linear buffer, which we will call a "product-specific" linear buffer for the sake of discussion. The ORPG prepends an internal 96-byte header to each product for storage in linear buffers. This "Pre-ICD" product header is used by the ORPG product distribution infrastructure. There are two classes of products. *Intermediate products* are not necessarily in any specified format and are not distributed to users. *Final products* are in an ICD format for distribution. The details of storage are not identical however.

Each product is stored individually in a separate linear buffer message. Generally, *intermediate products* are stored (along with the 96-byte internal header) in the corresponding product-specific linear buffer. *Final products* are stored (along with the 96-byte header) in the main product database linear buffer. In this case, the product-specific linear buffer message contains only the 96-byte header which includes a reference to the product database linear buffer message containing the product. The product-specific linear buffers are configured to retain several of the most recent product messages (typically 10 for volume final products and 40 for elevation final products). The retention of product messages in the product database linear buffer is centrally managed and is configured separately.

Note: Some intermediate products can be designated as *warehoused* data. This data is stored in the central product database in a manner identical to *final products*.

Volume 4. CODE Utility Guide

Document 2. CODEview Graphics (CVG)

CODEview Graphics Utility 9.2

`CODEview Graphics (CVG)` is an X Windows based product display tool that can be used to display ICD formatted products. ICD product formats are defined in the Interface Control Document (ICD) for the RPG to Class 1 User, document 2620001. In a manner similar to the **`CODEview Text (CVT)`** utility, **`CVG`** actually decodes the product. Beyond configuration of a few parameters, no modification to **`CVG`** is required to display new products, as long as the product contains the supported data packet types and follows the ICD format.

Section I Displaying Products with CVG

Section II Configuring Products for Display by CVG

Volume 4. Document 2. - CODEview Graphics (CVG)

Section I Displaying Products with CVG

CODEview Graphics 9.2

This Guide covers CODEview Graphics (**cvg**) 9.2 which is included with the source code for ORPG Build 13.

CAUTION: The **cvg** configuration files usually differ significantly from prior versions of CVG. Any locally developed products should be reconfigured from scratch. Do not attempt to reuse any configuration files from previous CVG installations other than locally developed color palette and digital legend files.

This document covers the following topics.

- **Part A. Introduction** A summary of capabilities, recent changes, limitations, and known problems.
- **Part B. Displaying a Product from the Database** Basic steps for selection and display of a product.
- **Part C. CVG Display Functions** A description of CVG product screen and main window functions.
- **Part D. Displaying Products not in the ORPG Database** Selection of products from binary disk files.
- **Part E. CVG Preferences** - Basic CVG functional preferences.

Build 12 Notes:

WARNING: Beginning with CVG 9.1 integrated with Build 12 ORPG, the format of the prod_config file has been changed to include a new entry for the packet 1 geographic coordinates override and a capability to override the default colors of a second data packet. Any development entries in the prod_config file from prior versions of CVG cannot be used / inserted into the CVG 9.1 version of the file. The Product Configuration Dialog should be used to add / modify entries rather that manually editing the prod_config file.

CVG 9.2 (integrated with Build 13) - WHAT'S NEW?

CVG 9.2 Changes:

Product Display Related changes

- Improved display of radial products to greatly reduce the number of black pixels between radials, artifacts of the display resolution and the X-windows drawing primitives.

Other:

- BUG Fixed: The product database size in CVG was smaller than the maximum possible in the RPG. This would cause the display of product other than the product selected for display when using larger product databases.

CVG 9.1c Changes:

Product Display Related changes

- BUG Fixed: A bug introduced in CVG 9.1b that prevents display of the generic radial component used in the DPR product.

CVG 9.1b Changes:

Product Display Related changes

- A Build 12 change in the radial header for the generic radial component changed the azimuth from center azimuth to beginning azimuth.
- Updated the colors (and product specification ICD) for the CC product. This resulted from the AWIPS design being frozen.

CVG 9.1a Changes:

Product Display Related changes

- Configured CVG to display the COMB_RAWDATA versions of the test products (IDs 700-705.

Other:

- Accomplished necessary changes for a site added in Build 12 releases in the change radar data file (KOUN)

CVG 9.1 Changes:

Product Display Related changes

- Modified the configuration of the Dual Pol test products (rawdata).
- Since geographic products using text packet 1 are continuing to be developed, added special treatment for text packet 1 as a new configuration item. This configuration option forces CVG to interpret the offsets in text packet 1 as pixel screen coordinates rather than 1/4 km from the radar (which is standard for geographic products). **WARNING:** This results in a format change to the configuration file prod_config.
- Added capability to modify the color of more than one data packet. The existing configured palette mechanism will primarily be used for the two-dimensional data arrays (all radial and raster data) which is also displayed in the legend bar area. The new capability will be used for other data packets (text, vector, symbols). To prevent inadvertent configuration errors, the existing configured palette mechanism no longer

provides the 'Apply to All Packets" option. An associated packet must be entered for the configured palette to be applied. The two-dimensional data arrays no longer have default palettes associated with them. The legend bars will not be displayed (i.e., no threshold labels or bars) if an associated packet is not configured. **WARNING:** This results in a format change to the configuration file prod_config.

User Interface Enhancements

- NONE.

Bug Fixes

- When CVG is launched, the cvg product list (cvg_db_list.lb) is deleted if it exists. This eliminates occasional problems with the product list.
- Fixed a bug in the display of the wind barb packet (packet 4). It was not displaying with any velocity over 100 kts. Corrected to display up to 195 kts.

Other:

- When in verbose mode output, added the printing out of the scale offset parameters in the product regardless of the legend display mode. Includes note that the output is meaningless unless the parameters are in the product.
- The CVG Color Palette Editor 'cvg_color_edit' has been replaced with a more capable and easier to use utility 'edit_cvgplt'. The new editor is a full-capability interactive palette editor which provides the capability of creating new palette files of a specified size, opening a palette without closing the editor, and saving the edited file to a selected file name. See Volume 4, Document 5, Part B.

CVG 9.0a Changes:

Bug Fixes

- Fixed an issue related to Build 12. With Build 12 the default size of the product database buffer was increased from 8,000 messages to 12,000 messages. Modified CVG to use the new default and to not permit setting the CVG product list buffer size to less than 12,000.

CVG 9.0 Changes:

Product Display Related changes

- Configured to display the Dual Pol test products (rawdata).
- Fixed issues with the display of non-geographic products (RCS, VWP, VAD, etc.).
    - Modified the display of non-geographic products so they can be displayed on the small image area in addition to the large image area.
    - Fixed the incorrect placement of the two-dimensional raster data array in the cross section products.
    - Fixed the problem with vector lines in the VAD product not always being displayed with the correct color.
- The display of text and vector packets was modified so that all text and vectors could be displayed correctly in both geographic and non-geographic products.
- Added special treatment of text packet 1 for a few specific products that use pixel screen coordinates rather than ¼ km coordinates for geographic products.

User Interface Enhancements

- Minor improvements on the display screen and the background map display options dialog.

Bug Fixes

- Fixed several issues in displaying individual layers / packets using the packet selection dialog. The correct packet was not always displayed. With certain products having a two-dimensional data array in layer 1, CVG would crash if overlaying a text packet in layer 2. With other products the 2-D array was erased when overlaying text packets in layer 2.
- Fixed the issue of exporting a displayed generic radial product to PNG or GIF files. However, an issue remains with export to GIF; with some products the image is broken up into sever mixed-up parts. The export to GIF function has been temporarily disabled.

Other:

- Added capability to set the default initial product display screen size and the image drawing area size.
- Made it impossible to override the default color palette assignment for specified data packets even if manually editing the product configuration file.
- Performance enhancement and code simplification. The logic involving the centering and scaling geographic and non-geographic products was placed into standard functions.

CVG 8.8 Changes:

Product Display Related changes

- Fixed a bug with method 5 display of packet 16 and the generic radial component introduced in CVG 8.7. All data levels beyond the last color transition definition were not displayed.
- Improved the data level decoding function to handle the largest and smallest values possible.

User Interface Enhancements

- Minor improvements on the display screen.

Other:

- Fixed three data click bugs. (1) The row reported for packet 17 (LFM grid) was incorrect. (2) The azimuth, radial number, and data level was not reported for the first bin (range = 0). (3) The decoded data values (introduced in CVG 7.7) were not always correct.
- Reconfigured the sample 1 algorithm products (1990 and 1995) to the range resolution of 0.13 nm for super resolution ingest.
- Added a printout of the scale offset parameters to the terminal window when displaying the legend bars using method 5.

CVG 8.7 Changes:

Product Display Related changes

- Modified the generic legend configuration logic to provide the option to use the Scale Offset parameters to calculate designated legend labels. The configuration can specify the source of the Scale Offset parameters as either the configuration file or the threshold fields in the product itself.
- The content and structure of the generic legend configuration file (Method 5) was changed.
- Fixed an error in the generic legend configuration color assignments. All colors

except for leading flag values were off by 1.

User Interface Enhancements

- Modified the graphic display screens to display the decoded data value of the data bin selected with the mouse (if the digital or generic radial product is configured for decoding).
- Product Configuration dialog buttons renamed for clarity.

Bug Fixes

- Corrected a problem in the digital legend preview function in the product information edit dialog window.

Other:

- Modified the mouse click data query to provide a decoded value using the Scale Offset parameters either in the product threshold levels or the product configuration file.
- Performance enhancement and code simplification.
  - Modified the product and legend display functions to reduce the number of times the legend file is parsed (from 3 to 1), to reduce the number of times the legend blocks are drawn (from 2 to 1).
  - Modified the product display function to simplify the redisplay of a product from history (for example when changing the zoom factor, the display attributes, image center, etc.).
- The generic legend configuration (method 5) is the recommended method for all digital and generic radial products.
- The name of the configuration file for product display has been changed from **`res_list`** to **`prod_config`**.

Part A. Introduction

CODEview Graphics (CVG) is an X Windows based product display tool that can be used to display ICD formatted products. ICD product formats are defined in the Interface Control Document (ICD) for the RPG to Class 1 User, document 2620001. In a manner similar to the **CODEview Text (CVT)** utility, **CVG** actually decodes the product. Beyond configuration of a few parameters, no modification to **CVG** is required to display new products, as long as the product contains the supported data packet types and follows the ICD format.

CVG is designed to be a data analysis display tool rather than an operational display tool. There is no attempt to make the **CVG** display look like AWIPS or any other operational meteorological display though color palettes in compliance with the WSR-88D Product Specification are supplied. In the future CVG

- By default, geographical products are displayed at full resolution rather than at a specific scale. There is a zoom capability to provide an ability to compare products having different spatial resolutions.
- Elevation based radial products are conical sections. **CVG** does not accomplish a projection of the data to the surface. The range displayed is slant range.
- **CVG** provides a capability to display background maps. However, when displaying elevation radial products the map overlay is useful only at lower elevations because the conical section data are not projected to the surface. Surface range error increases with the elevation angle.
- The product display colors used by **CVG** comply with WSR-88D Product Specifications with respect to the 4-bit run-length encoded radial and raster products and various text, symbol, and vector data packets. If colors specific to other display systems are desired, color palettes can be defined accordingly. For the 8-bit "digital" radial products and generic radial products, future versions of CVG will provide the capability to use AWIPS default operational colors rather than the development colors.

Basic Display Capability

- **CVG** provides a list of all final products currently in the ORPG product database and allows the user to select a product for display. **CVG** provides the capability to switch to a different product database and to display products stored as individual binary files.
- Once a product is selected, **CVG** either
  - displays the entire contents of the product, or
  - displays a list of all available data layers in the symbology block and the presence of a tabular alphanumeric block (TAB) and a graphic alphanumeric block (GAB) for selection by the user.
- **CVG** detects and reports errors in product structure. The types of errors detected are being expanded.
- **CVG** does not verify the contents of the header portion of the final product. The **CVT** utility should be used to validate data fields in the product description block.
- **CVG** is designed to display all WSR-88D products except: status messages, Radar Coded Messages, Free Text Messages, User Alert Messages, the Weak Echo Region product, and the

SuperOB product. Currently CVG only provides display support for the area and radial components of products containing the Generic Data Packet (packet 28).

NOTE: The concepts of intermediate products, linear buffers, and the ORPG product database are explained in CODE Guide Volume 2 Document 1 - *The ORPG Architecture*.

Additional Functions

Additional features of CVG include:

- Context sensitive on-line help for application windows.
- The capability to display multiple layers / products overlaid on top of one another.
- A ± 32X Zoom capability.
- Manual and automatic animation of a selected product. This can be either a time sequence animation (displaying a product from the previous or subsequent volume scan) or an elevation sequence animation (displaying elevations within a volume scan in either order).
- CVG provides information about the data point of a displayed product with respect to the cursor location.
    - This information includes the coordinates of the point and the data value being displayed.
    - If Method 5 is used to configure display of a product, CVG can also display the decoded value of scaled-integer products (packet 16 and the generic radial component of packet 28).
- CVG can display two products simultaneously. This includes a basic linking between the displayed products during animation and with respect to displaying information about the data point at the cursor location.
- The ability to export graphics to a PNG or GIF image. (Export to GIF temporarily disabled)
- A capability to switch between color palettes and to configure new products for display.
- The ability to display products from varied sources: alternate Product Database Linear Buffer and binary disk files.
- CVG automatically handles ORPG compressed final products. The product structure is examined to determine whether the product must be uncompressed prior to display.

Limitations

Product Display Related Limitations

- With one exception, CVG can display products created by previous versions of the radar, whether reading from a saved product database or individual product binary files. The exception: CVG cannot display the DMD product (which is a generic product) that was created with an ORPG prior to Build 8.
- While CVG can decode scaled integer array products (unsigned integer arrays in packet 16 and the generic radial) using the Scale-Offset method, **CVG does not use the traditional parameters in threshold levels 1-3 to decode the data levels** (see Section II Part D and Appendix A).
- Product Specific:
    - CVG must be configured correctly to handle new products that do not follow typical methods of encoding information into unsigned integer arrays. One example, product ID

134, Hi Res VIL does not have a constant increment between encoded data values. Because of this **CVG** cannot decode the values using standard parameters. The product must be configured using a method which permits specifying the digital legend labels explicitly.

  - **CVG** does not accomplish a geographic projection of the array data in the Hourly Digital Precipitation Array (DPA) product. The 1/40 LFM Digital Precipitation Data Array (packet code 17) and the 1/4 LFM Precipitation Rate Data Array (packet code 18) are displayed in a tabular manner enabling the evaluation of the row and column data.
  - Product ID 49 (product code 62) Storm Structure is unique in that it is similar to Stand Alone Tabular Alphanumeric Product. However, it also includes additional data not intended for display. **CVG** does not display this additional data.
- While helpful diagnostics have been added to assist detection of structure problems with the major block / product lengths and with the GAB and TAB portion of the product, similar capability is not yet provided for displaying individual data packets in the symbology block.
- CVG does not verify the contents of the header portion of the final product. The **CVT** utility should be used to validate data fields in the product description block.
- Generic Components:
  - Currently **CVG** only provides display support for the area and radial components of products containing the Generic Data Packet (packet 28). The **CVT** utility also supports the text and table components.
  - **CVG** only supports the unsigned integer type data for display of the generic radial products. In the future, signed integer and real data types will be supported. The **CVT** utility supports all data types.
- **CVG** provides a capability to display background maps. However, when displaying elevation radial products the map overlay is useful only at lower elevations because the conical section data are not projected to the surface. Surface range error increases with the elevation angle.

Other Limitations

- If **CVG** is being used with more than one ORPG Build, it should be compiled on the latest Build. Known differences between ORPG Builds are handled with the **Product from ORPG** option menu by selecting the ORPG Build that produced the products / database linear buffer.
- **CVG** should be used with a monitor/console in a Tru Color mode. Specific limits are:
  - PRODUCT DISPLAY REQUIREMENTS: **CVG** can display products in 16-bit, 24-bit, and 32-bit mode.
  - EXPORT TO IMAGE FILE REQUIREMENTS: In order to export the displayed product to a GIF / PNG image file, the X-windows display must be in either 24-bit mode or 32-bit mode. (Export to GIF temporarily disabled)
- **CVG** has a graphical canvas of 1840 x 1840 pixels. This is the inherent limit of how much of the product data can be displayed at a given zoom factor. An alternate canvas of 768 x 768 pixels is provided.

Known Problems

Problems NOT caused by **CVG**:

- If you are using the KMLB data supplied with CODE you may notice a graphical artifact which is manifested in various products as a ring of data dropout at about 80 miles. This artifact is in the Archive II data. You may also see other artifacts with higher elevations of base reflectivity products which are not caused by **CVG** (a bug in some legacy base products).
- **CVG** is sensitive to products having inaccurate information in certain header fields. These include offsets to portions of the ICD products, length of block, number of radials / rows, number of range bins, number of bytes in a radial / row, etc. This can cause nonsense to be displayed (**CVG** may crash reading nonsense data).
- **CVG** may crash if data packets are not properly constructed. For example, each radial contained in a digital data array (packet code 16) must begin on a halfword boundary. If there are an odd number of data bins, the structure must be padded with one additional byte (containing data value 0) as stated in the ICD.

There remain several problems with a few features.

- Product Display
  - **A possible difference in the color selection for display between method 2 and method 5.** The error appears to be focused on the lowest colors next to the leading flag values.
  - CVG does not handle the various resolutions of the Combined Shear (CS) product. **This product was eliminated in Build 11**.
  - As used by several precipitation products in the symbology block, text packet 1 is used in a non-standard fashion. **CVG wraps long lines of text in packet 1 at 80 characters in order to display the text on the graphic screen. CVG also provides a configuration option to permit use of pixel screen coordinates in a geographic product.**
- **Volume animation does not always work with product data bases that have been "wrapped" (data base filed then older products replaced by newer products**. This has been noted when the data base has wrapped multiple times and more recent data ingested (message 31 containing DP data).
- The base product information normally displayed in the upper right corner of the display window sometimes does not appear when displaying a 4-bit product after an 8-bit product. *Needs to be verified.*
- **The range rings do not display if the combination of off-center display and zoom factor places the radar location beyond the edge of the drawing area**.
- The export to **GIF** function does not work correctly for all products (the image is broken into several mixed-up parts). This function has been temporarily disabled.
- Even though product configuration checks, tests for final product format, and product structure parsing checks have been added, **CVG may still occasionally freeze, crash, or simply display a blank image if decoding data in a format that is not expected. Please report all problems to the address given under the Help->About CVG** menu.
- When animating two products simultaneously (Screen 1 and Screen 2 linked), the products will eventually get out of sync (for example, different volumes during a time-series animation) during automatic or continuous animation. The products remain matched if manually stepping to the next products.

CVG Preferences

1. With the **local installation** that is accomplished when the ORPG is compiled:
   - The environmental variable **`CVG_DEF_PREF_DIR`** must be set to **`$HOME/tools`**

2. If the optional **global installation** was accomplished with the **`cvg_global_install`** and **`cvg_install_config`** scripts, the **`CVG`** default configuration files are placed in **`/usr/local/share`**.

   If a **global installation** was accomplished and this location was altered by modifying the **`cvg_install_config`** script, the definition of **`CVG_DEF_PREF_DIR`** must reflect this new location.

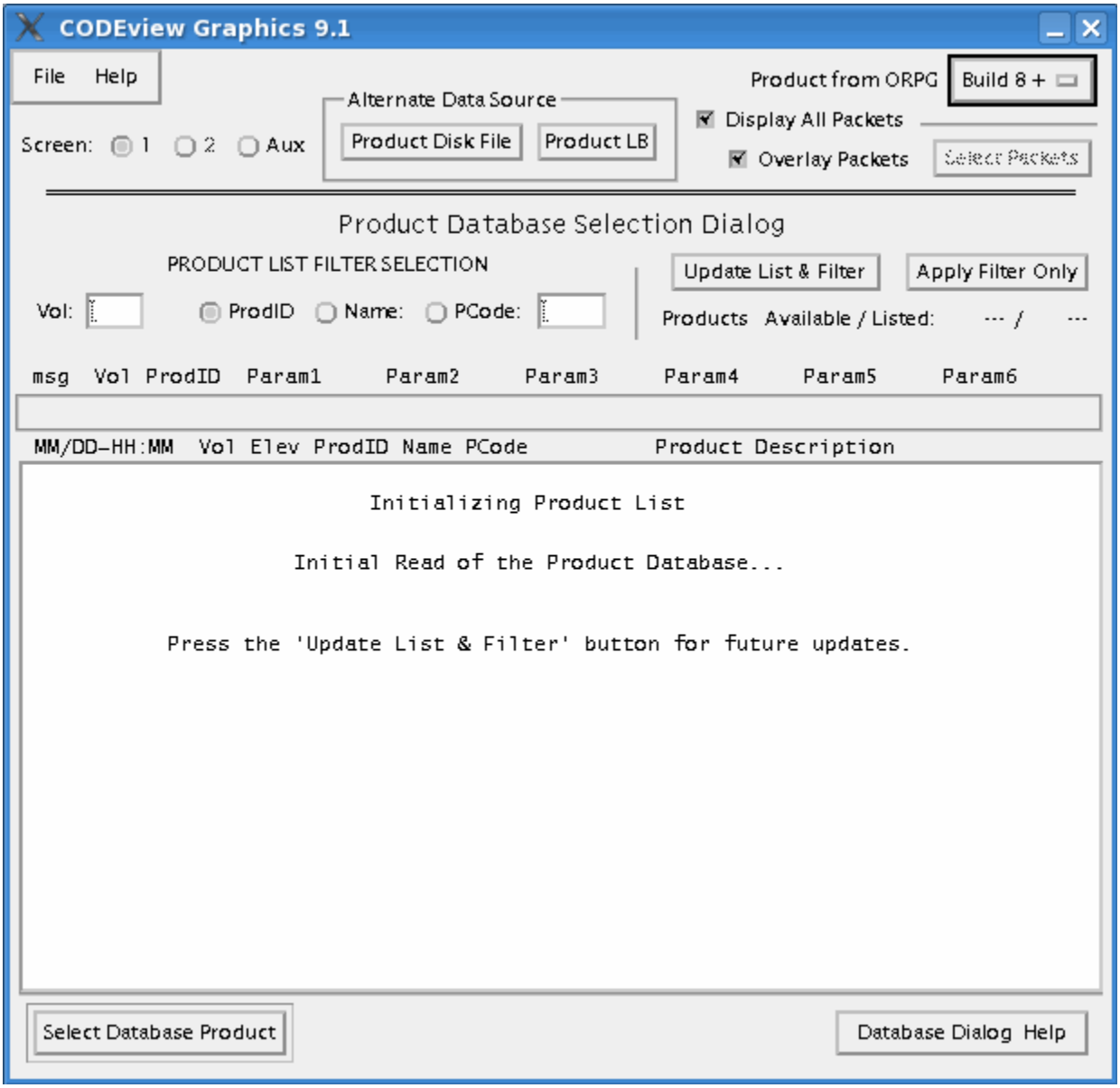
The local configuration directory, **`$HOME/.cvgN.N`**, is named according to the version number. For example: **`.cvg7.0`** for **`CVG`** 7.0. When launched, **`CVG`** searches for the local configuration directory **`$HOME/.cvgN.N`**. If the local configuration directory does not exist, it is created and the default configuration files are copied from the global directory.

`CVG` uses the local configuration files which can be customized. These local files can be replaced by the defaults by (1) renaming or deleting the local configuration directory (**`~/.cvgN.N`**) and (2) launching **`CVG`**. This is useful if the configuration files have become corrupted.

Part B. Displaying a Product from the Database

1. Launching CODEview Graphics

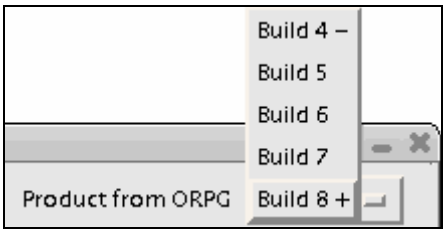
CODEview Graphics is launched by executing **cvg** (lower case) at the command line. The main window is displayed.



When CVG is launched, the above message is displayed for a few seconds while CVG initializes the product list. If no list is produced after about 5-10 seconds, press the 'Update List & Filter' button. There are times when CVG cannot produce a database product list. This could simply be due to having

accomplished an ORPG start without ingesting any radar data or due to incorrect configuration of the CVG preferences. Messages are provided in the list window offering suggested remedies.

Before selecting a product for display, the Build Number of the ORPG that produced the products should be selected using the **Product from ORPG** option menu This applies to all sources of data (reading from database or individual binary files).



One reason the Build number must be set correctly is to ensure the correct sort order of the database product list.

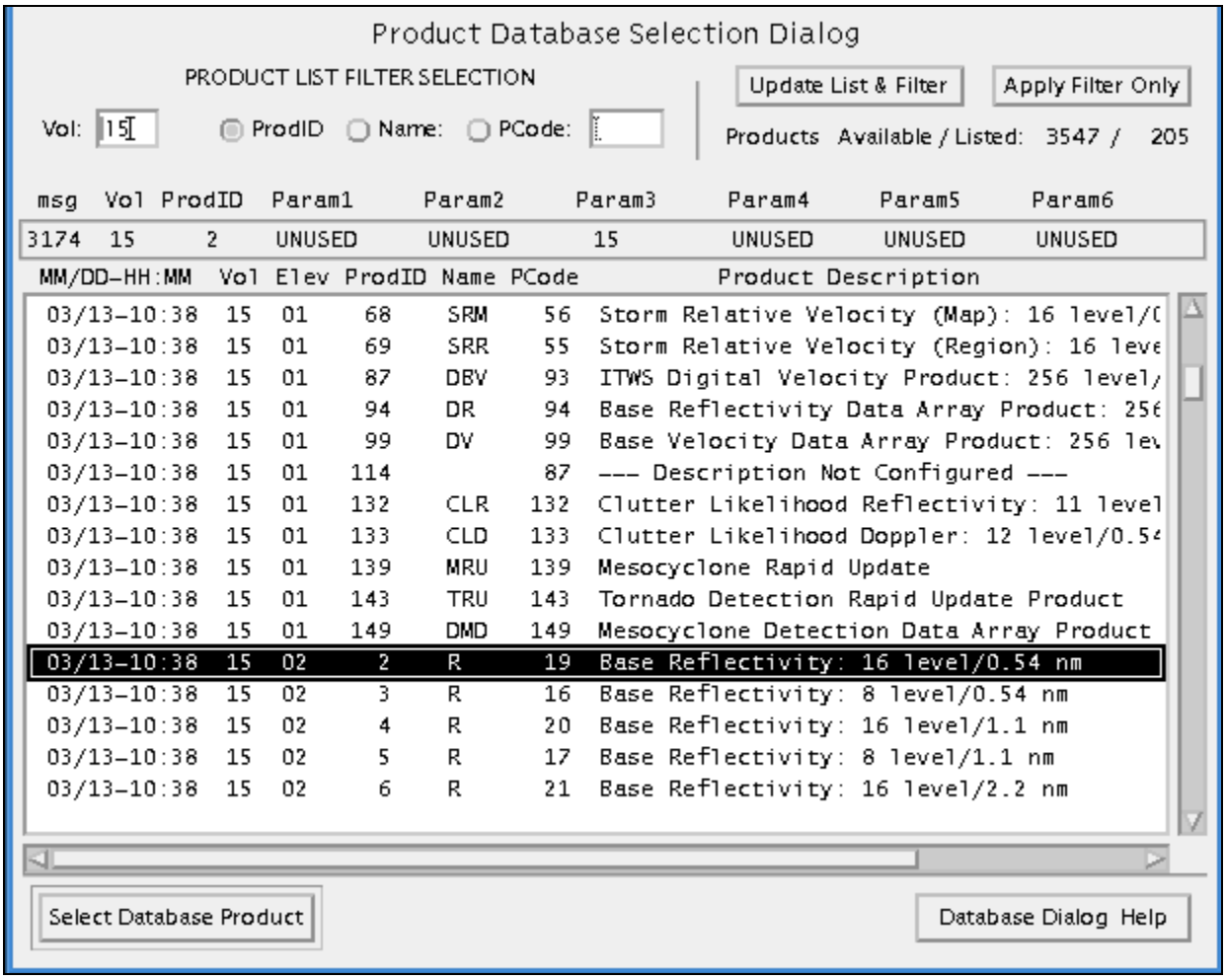
2. Selecting a Product Message

A product message is selected from the Product Database Select Dialog window as follows. Options for selection of products from a binary disk file are explained in Part D of this document.

A list of products in the configured database linear buffer is displayed automatically when **CVG** is launched. This list is sorted on volume time, elevation index, and product ID.

The product list may be filtered on volume number and/or one of the following: product ID, short product name (1-3 character mnemonic), product code. This is accomplished by entering values in the corresponding data entry field and then pressing / repressing either the **Update List & Filter** button or the **Apply Filter Only** button.

The **Update List & Filter** button first obtains a new product list from the product database (if available) then applies the filter to the new list. The **Apply Filter Only** button filters the existing list.



A product is selected by either double-clicking on the desired product or by highlighting (selecting) the desired product and clicking the **Select Database Product** button.

Product Request Parameter Display Pane

This pane, just above the database list, displays the 6 request parameters associated with the currently highlighted product. This information is useful in selecting certain products that were generated as a result of multiple one-time-requests. Also displayed are: the internal database message sequence number, the volume number, and the product ID. The database message sequence number is used to select a product with the CODEview Text (**`CVT`**) utility.

Database Product List Notes:

- The first column is the volume date-time in the **`MM/DD-HH:MM`** format. **`Vol`** is a sequence number for the volume (1, 2, 3 .....). The **`Elev`** is the ordinal of the elevation in the current Volume Scanning Strategy. **`ProdID`** is the product ID or linear buffer number of the product. The **`Name`** is the 1-3 character short name and **`PCode`** is the product code displayed at the ORPG and AWIPS display device. Currently intermediate products in the database (warehoused) are not listed by CVG. **`Product Description`** is a brief description of the product.

- The product description will be "**`Description Not Configured`**"

- When using the **ORPG Product Info** as the source of product descriptions and the linear buffer containing the binary product configuration information for the ORPG has no entry for that product ID or the buffer is not present.
- When using the **CVG Product Descriptions** as the source of product descriptions and the alternate list of product descriptions has no entry for that product ID.

This could occur with archived product database linear buffers and with binary product files. This can also occur when displaying user added products created on another ORPG or when displaying Legacy products that were removed in Build 4. Even though the description is "**`Description Not Configured`**", the product will be displayed if CVG is properly configured.

- For "legacy" products (product codes less than 130), the linear buffer number is not the same as the legacy product code (message code). For example, product 19, base reflectivity, has a linear buffer number of 2. The product attribute table in the **`product_attr_table`** configuration file is one source for determining the linear buffer number associated with a product.

- If using CVG with an ORPG database prior to ORPG Build 6, all elevation indexes may indicate 0 if the correct ORPG Build number has not been set using the **Product from ORPG** option menu.

LIMITATION: **`CVG`** displays an Error message and aborts product display if attempting to display a product that is not configured or if certain configuration parameters remain at initial settings.

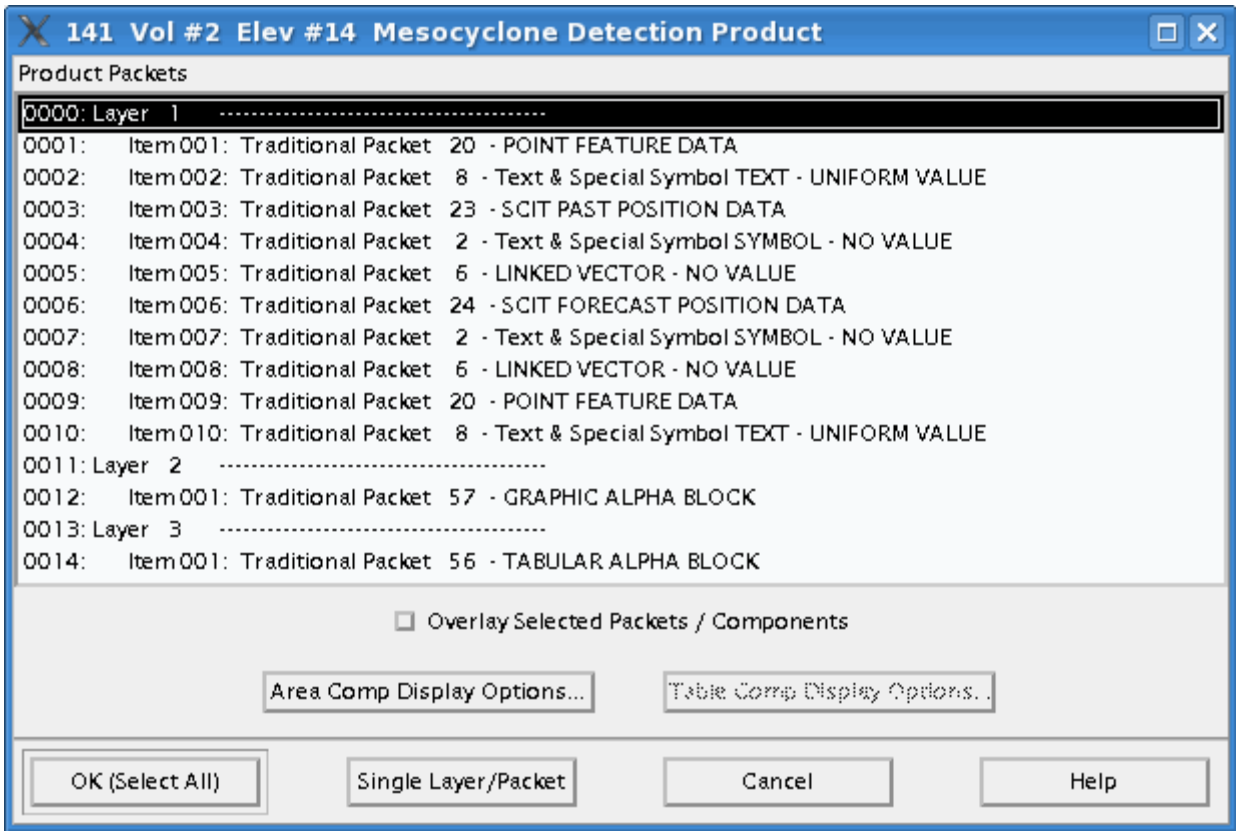
`CVG` may crash if attempting to display incorrectly configured final products (ICD products) or if attempting to configure and display non-ICD products stored in the product database (*warehoused* intermediate products). Currently intermediate products in the database are not listed in the database product list.

3. Selecting Portions of a Product to display

When a product is selected and loaded, the Packet / Component Selection Dialog window automatically opens if the Display All Packets check box is not checked. Note that the Overlay Packets check box is not active in this case.

Traditional Products

A 'Traditional' WSR-88D Graphic Product may contain one or multiple data layers. The Packet Selection Dialog window lists all of the layers (and packets within each layer) in the symbology block. The GAB and TAB, if present, are listed as a layer.



An entire product (except for the TAB) can be displayed by double clicking anywhere in the list, by hitting the 'Return' key, or by clicking the **OK (Select All)** button. A TAB must be selected individually for display.

An entire layer or an individual packet may be selected for display by highlighting the item and clicking the **Single Layer/Packet** button.

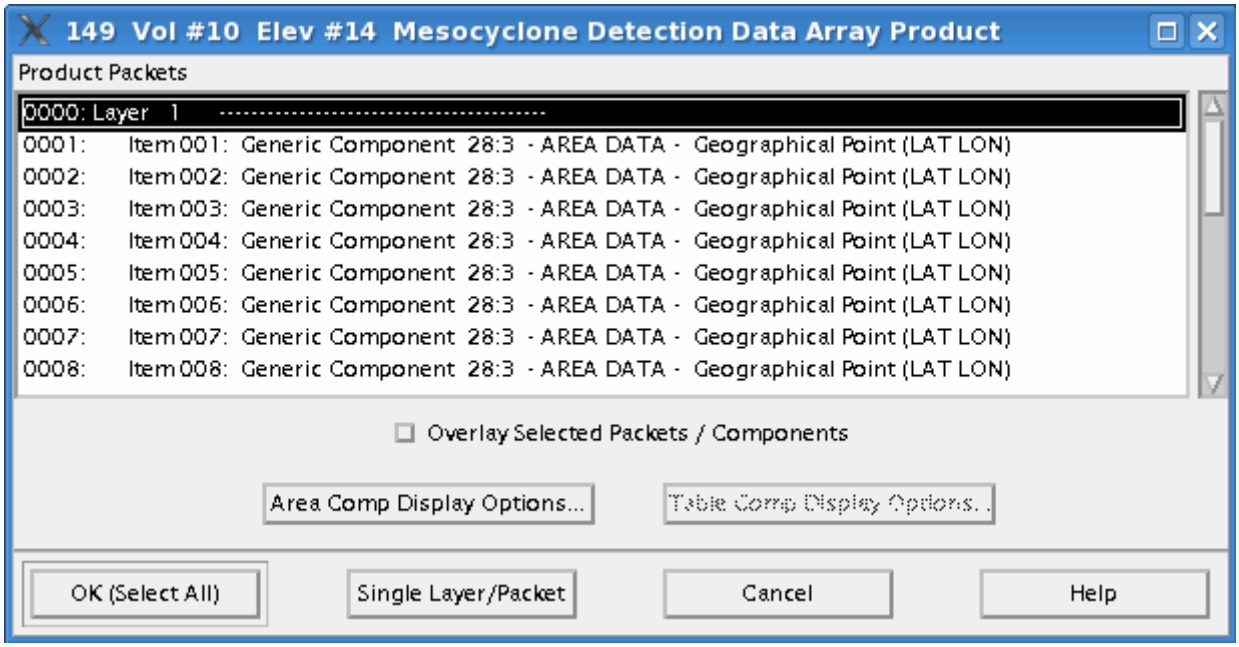
If the **Overlay Selected Packets / Components** Check Box is checked prior to selecting the packet(s), the selection will be displayed over the existing product being displayed. If not checked, the existing displayed product (if any) will be replaced by the selected packets.

When either the **OK (Select All)** button or the **Single Layer/Packet** button is pressed the dialog disappears and the selected data is displayed on the product display screen.

> NOTE: One product requires special consideration. Due to the nature of the DPA product only a single layer should be displayed at any time. CVG provides a prompt to the user when this product is displayed.

Generic Products

A 'Generic' WSR-88D Graphic Product contains one data packet 28 in a single data layer. The Packet /Component Selection Dialog window lists the layer and all of the generic components within the generic product data packet.

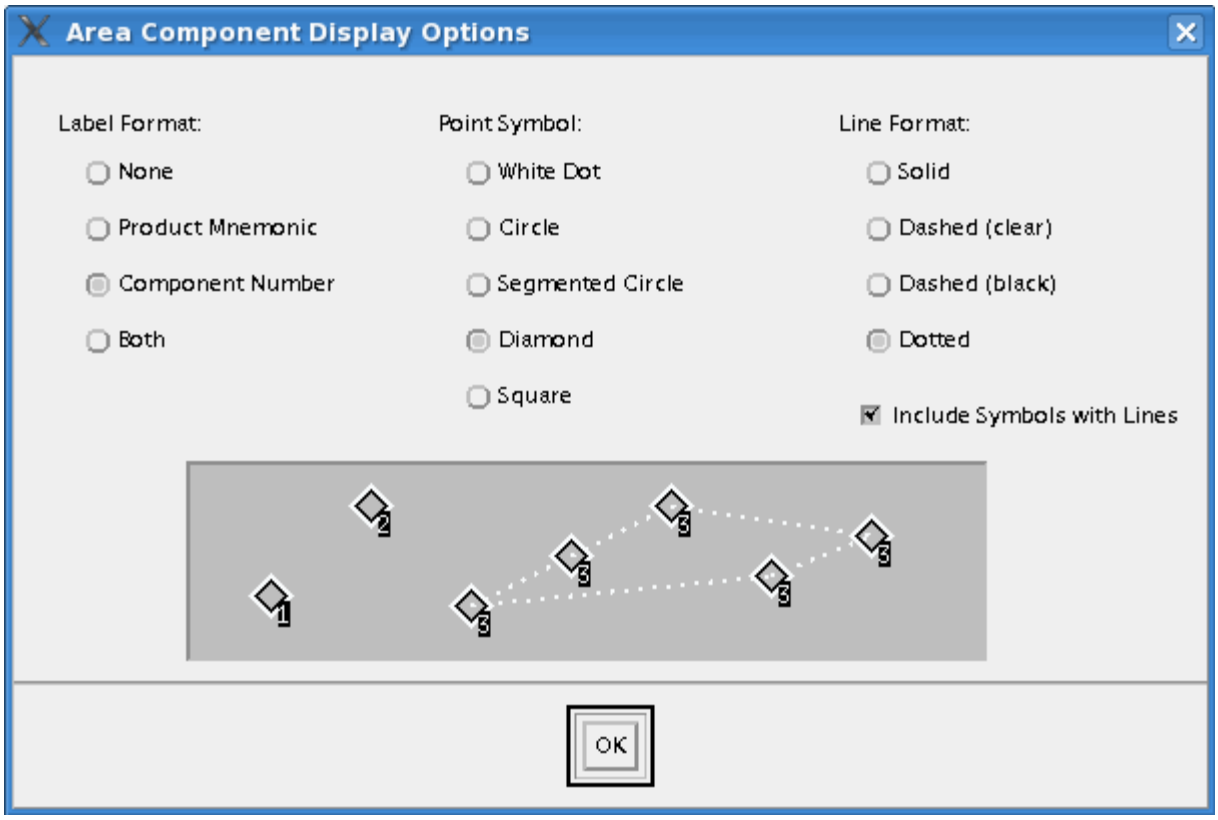


The selection of components for display is accomplished in the same manner as selection of data packets for display in a traditional product.

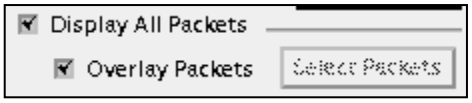
Customizing Display Options for the Generic Area Component

The display of generic products using the area component is not fully developed. Initially some control over the display of area components is provided manually by using the Area Comp Display Options... button. A capability is provided to select several options for the symbol used for the point representing the location of the feature, the label next to the symbol, and the format of the line connecting the points.

If the area and point components are displayed individually, different display attributes can be selected for each component.



Bypassing the packet selection dialog

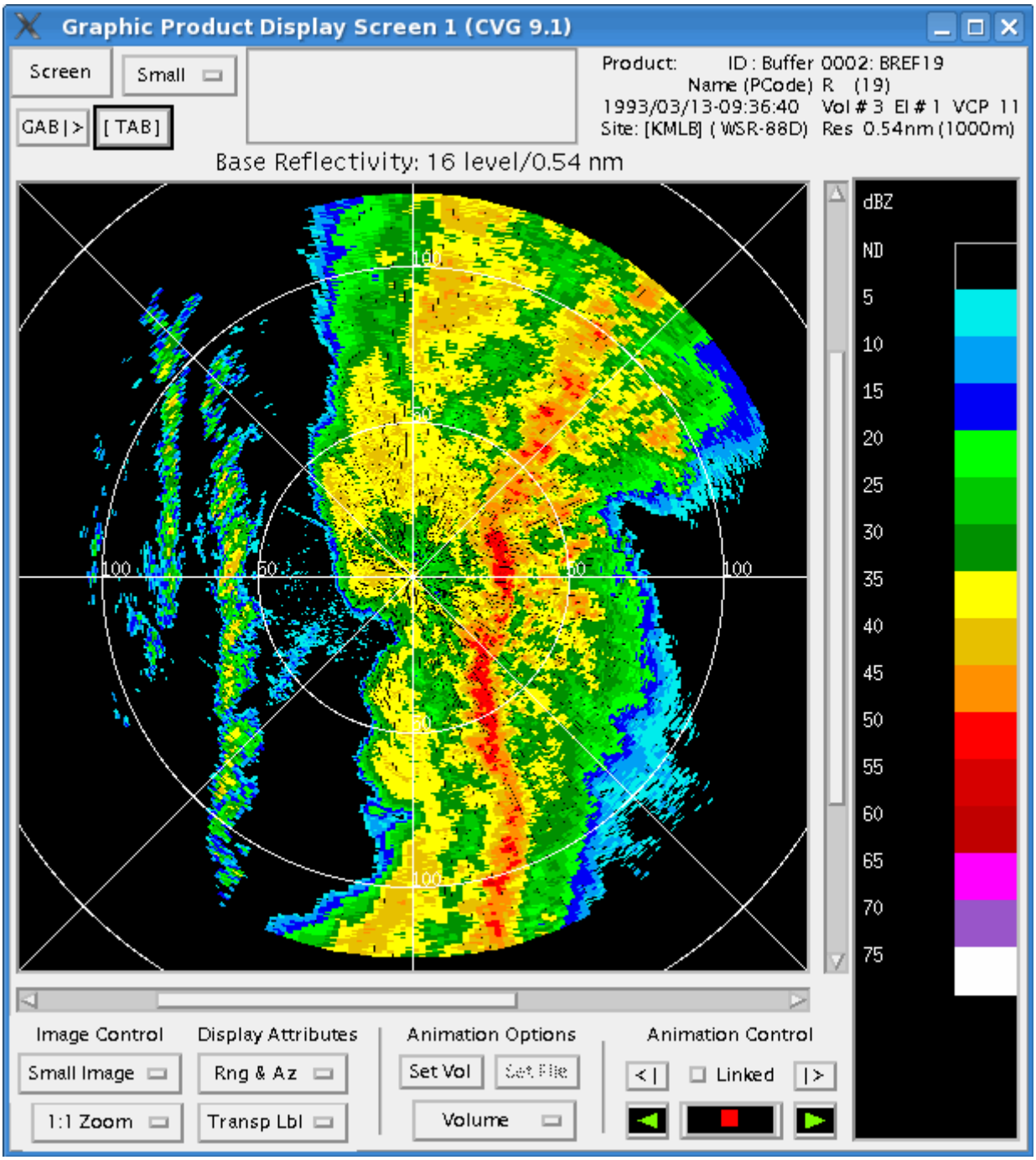


If generally all portions of the selected product are to be displayed, the packet selection dialog can be skipped by checking the Display All Packets check box on the main window. This will display all graphic portions of the product including the GAB but will not display the TAB.

With **Display All Packets** selected the **Overlay Packets** check box is active and can be used to control product overlay.

4. Displaying the Selected Data

The selected packets are displayed in the CVG Graphic Display Window.



The graphical data are displayed in full resolution and may be too large to be seen at one time. In this case the product display pane is a view port showing a portion of the product. The entire product is captured in a virtual canvas behind the display pane. The scroll bars can be used to view selected portions of the image.

Product Information Panel

The Base Product Image pane in the upper right portion of the display window provides information about the displayed product. This information always pertains to the initially selected product rather than subsequent overlays. The information includes:

- The product ID number and buffer name (**`2:BREF19`** in the example).
- The short product name (1-3 character mnemonic) and product code (**`R (19)`** in this example).
- The GMT date and time that the data was produced by the radar.
- The volume sequence number and elevation index of the product.
- The type of scanning strategy being used (**`VCP 11`** in the example).
- The horizontal resolution of the product (in Nautical miles).
- The site id and type radar (**`KMLB`** and **`WSR-88D`** in the example). Note that this information is not in the radar data per se. **For the site ID (and therefore radar type) to be correct, the ORPG must be using the correct site adaptation data when the products were generated.**
- The product description is shown just above the graphic display pane. This description will be "**`Description Not Available`**"
    - When using the **ORPG Product Info** as the source of product descriptions and the linear buffer containing the binary product configuration information for the ORPG has no entry for that product ID or the buffer is not present.
    - When using the **CVG Product Descriptions** as the source of product descriptions and the alternate list of product descriptions has no entry for that product ID.

NOTE: The currently selected display zoom factor is shown in the option menu in the lower left corner of the display window.

Color Legend Panel

The color legend is displayed on the right side of the display window. The threshold labels represent the minimum value (or flag value) that is represented by that color.

> For products originally intended for display - which use special run length encoded (RLE) data packets with 4, 8, or 16 data levels - the special flag values and data level thresholds are encoded explicitly in the product. For digital products (that were not originally intended for display) and the new generic radial products, the threshold levels must be configured with **`CVG`** preferences. When configuring products for display by **`CVG`**, we recommend following the convention for RLE products that provides labels representing the threshold (or minimum) value to be assigned a particular color. See Section II of this document, *Configuring Products for Display by CVG*, for additional information.

5. CVG Product Error Tests

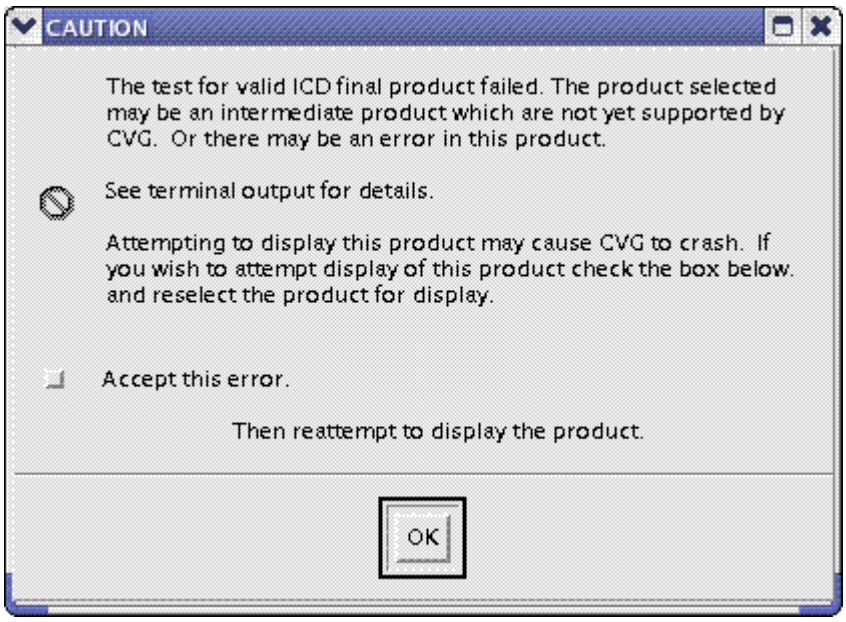
CVG accomplishes several validity tests of WSR-88D final product prior to display. These include a basic test for an ICD product that tests 3 product header values for proper range, a consistency test for product block lengths and product size. In order to accomplish additional evaluation of the product these tests can be ignored and the product displayed (with a risk of causing the CVG application to fail). These tests are also accomplished by CVT.

Test for Final Product (ICD Product Format)

The test for a valid ICD product is accomplished while the product is being read (from either the product database or an individual product disk file.

This test simply looks at three header values to determine if a final product has been loaded. These fields are: the -1 divider for the Product Description Block, the RPG elevation index, and the volume number (1-80).

Additional information is provided in the terminal window.

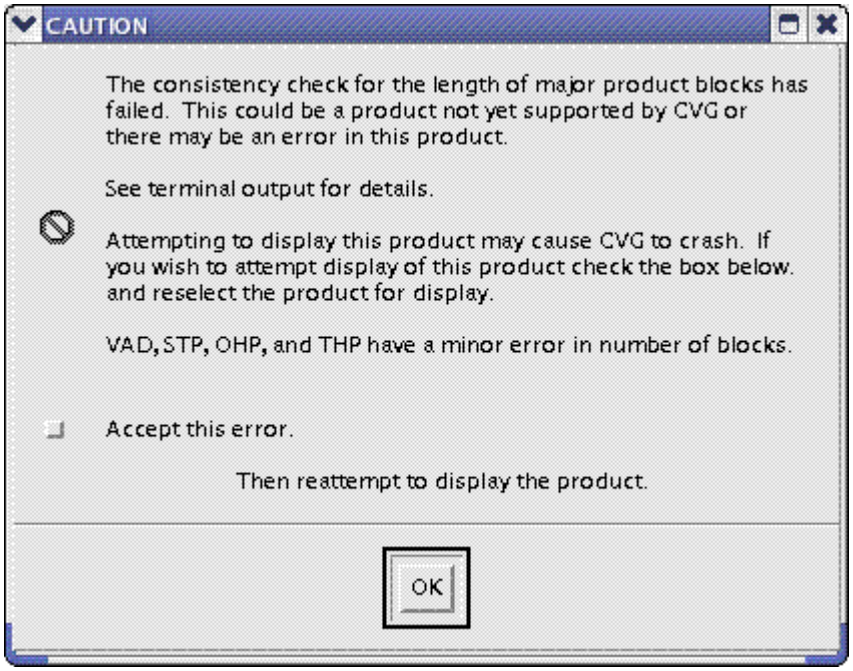


Test for Top Level Block Structure

The test for high level structure of the product (major block lengths and overall product size) is accomplished after loading the product and before parsing the individual data packets.

This test includes parsing the individual pages of the GAB and the Stand-Alone-Tabular Alphanumeric Product. The individual pages of the TAB are not checked until the TAB is displayed.

Additional information is provided in the terminal window.



Test of Individual Data Packet Structure

Currently the only test of individual data packets is the un packing of the run-length encoded radial / raster packets (AF1F, BA07) during display. This is accomplished to avoid CVG crashes. To avoid crashing, CVG terminates parsing packets when an invalid packet ID is detected. Additional tests are planned for future versions of CVG.

Part C. CVG Display Functions

----- Product Display Screen Functions -----

Display Window Size

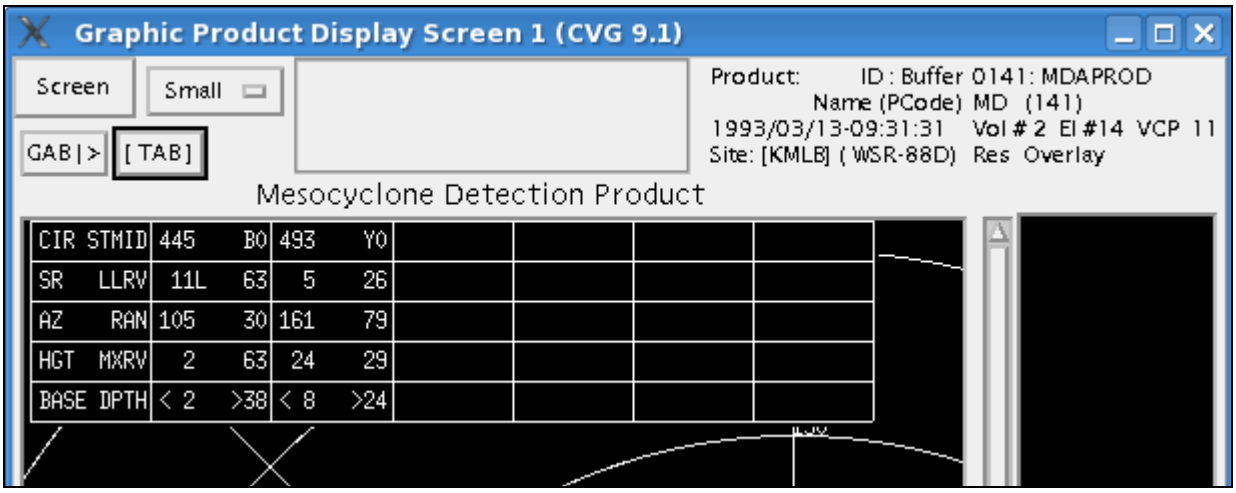
The Small (default) window size provides the smallest window in which the user interface elements are all available. This requires a minimum display resolution of 1024x768. With a 1280x1024 display, two product display windows fit side by side with no overlap.

The Large window size can be used with display resolutions greater than 1024x768 to provide a useful maximum display window size that retains the aspect ratio of the product screen. With a display resolution of 1024x768 the Large window size is the same as the small window size.

GAB Display

There are two ways to select the GAB for display. (1) The GAB can be selected as a layer from the Packet Selection Dialog window (either individually or via 'Select All'). (2) Hitting the **GAB | >** button on the graphic display window will display the GAB of the most recently displayed graphical product.

If the layer representing the GAB has been selected, it is also displayed in the Graphic Product Display Screen in the upper left hand corner of the product image (use the scroll bars to bring into view).

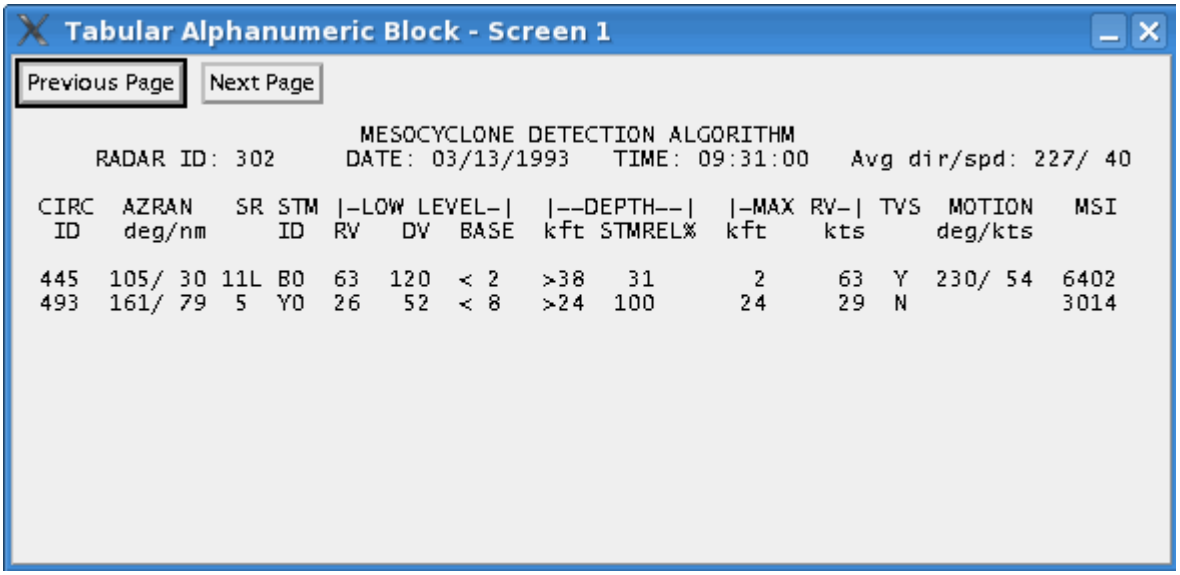


After the Gab is displayed, clicking the **GAB | >** button pages through the GAB.

TAB Display

There are two ways to select the TAB for display. (1) The TAB can be selected as a layer from the Packet Selection Dialog window. (2) After the graphic portion of a product has been displayed, hitting the **[TAB]** button on the graphic display window will display the TAB of the most recently displayed graphical product.

If the layer representing the TAB has been selected, it is displayed in a separate window.



The TAB can be paged through using the **Previous Page** and **Next Page** buttons.

Image Zoom

After display, an alternate zoom factor can be selected using the Zoom option menu in the **Image Control** pane. Changes to this field are immediately applied to the corresponding product display screen if a product is currently displayed.

The default factor of 1:1 is used when a product display screen is opened regardless of the previous setting.

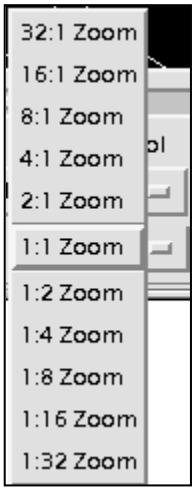
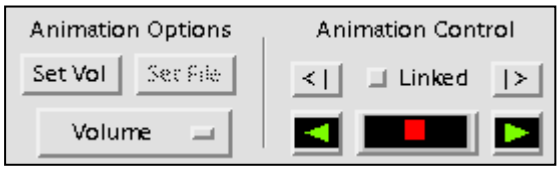


Image Animation

The Animation Panel provides the capability to quickly compare related products.



The animation type select option menu at the bottom of the **Animation Options** sub panel determines the type of animation:

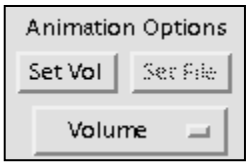
A **Volume** (or Time) Series displays the same product from the previous or subsequent volume.
An **Elevation** Series displays different elevations of the same product from within a volume scan in either order.
A **Most Recent** mode automatically displays the most current version of the selected product. This functionality is only defined when displaying products from an ORPG product database while that ORPG is running and ingesting base data.
A **File Series** animation is planned for a future version of **CVG**. This will animate products contained in individual product binary files.

Animation Control Sub Panel

Manual Animation Controls Loads either the next or previous set of products (base image and overlaid products).

Automatic Animation Controls Continually loads either the previous or the next set of products (base image and overlaid products) until the red stop button is pressed. There is an inherent delay in loading and rendering products. If the Most Recent mode has been selected, the most current version of the product in the database is continuously displayed and updated as required.

Animation Options Sub Panel



The **Set Vol** button allows the user to control which volumes within the product database are used in a time series animation.

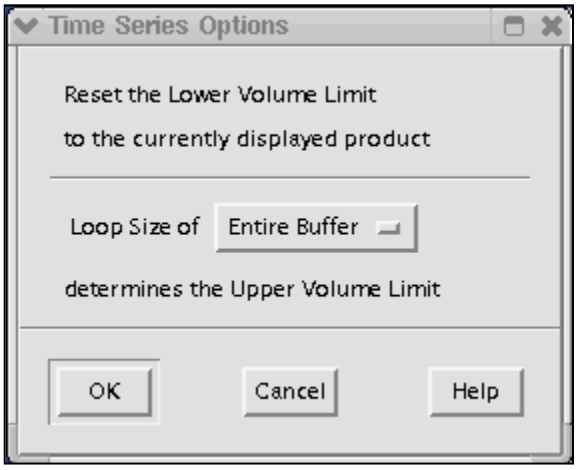
Set File Button: *Not implemented.*

Volume (Time Series) Animation Options

The location and size of the animation loop must be set before initiating a time series animation. This is accomplished by

1. Displaying the product representing the lower end of the animation loop (the earliest volume time) in the product window.

2. Setting the limits of the animation loop by pressing the **Set Vol** button. The following dialog window is displayed.

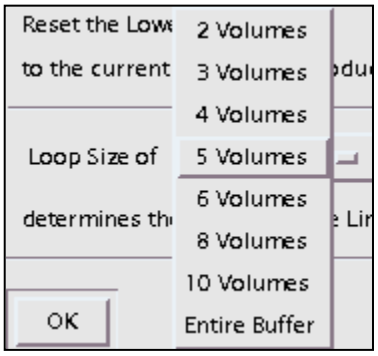


Pressing **OK** closes the window and sets the first volume to the currently selected product and the last volume according the selected size.

Pressing **Cancel** closes the window without setting the options.

The default loop size is the Entire Buffer.

Loop sizes from 2 - 10 volumes can be selected. If the **Entire Buffer** is selected, the animation progresses completely through the product database.

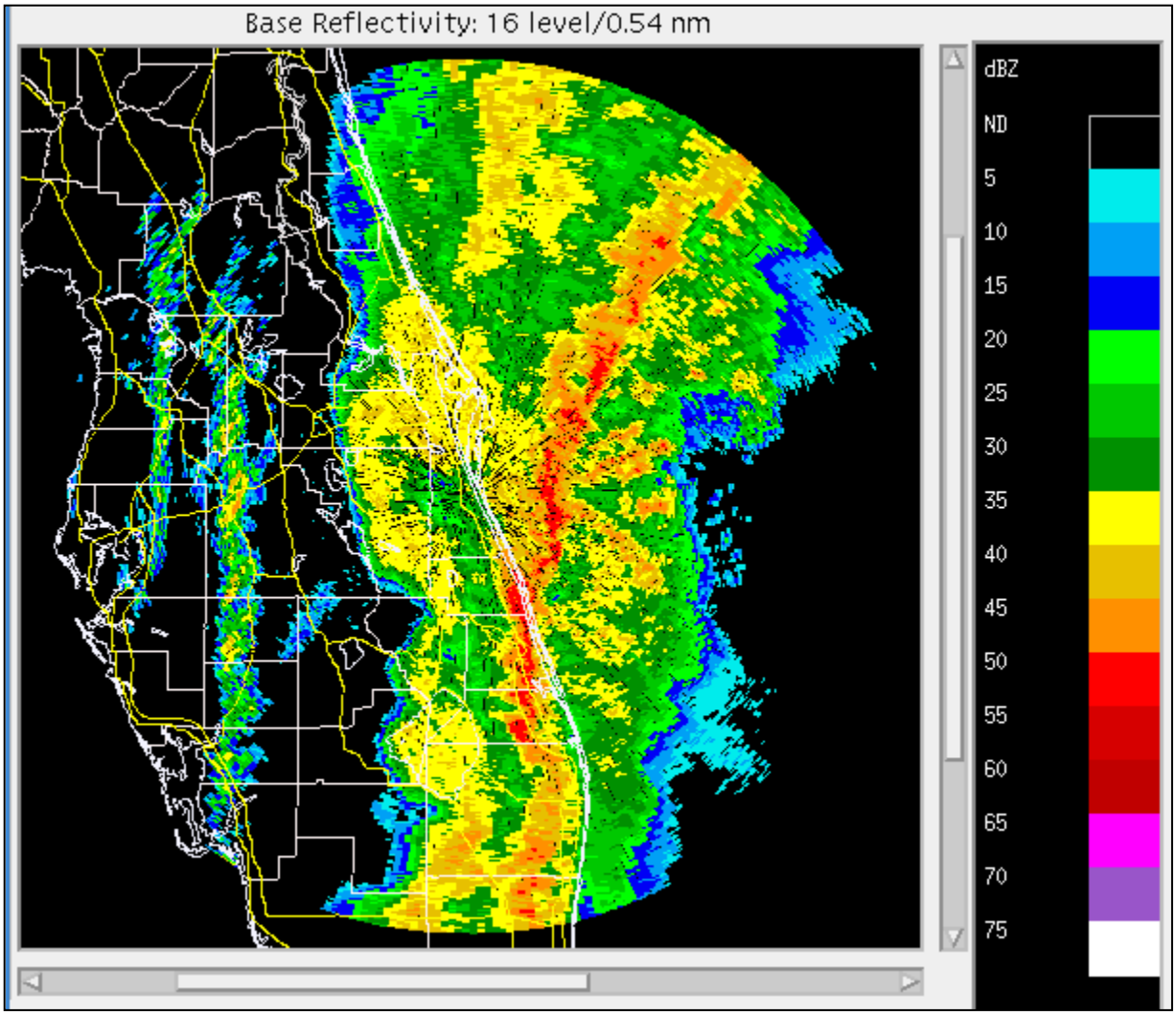


After setting the loop limits, if another product is displayed by any other means (manually through the product / packet select dialogs or via the Most Recent animation mode) the loop limits must be reset via this dialog.

Range Ring / Background Map Overlay

The top option menu in the **Display Attributes** pane provides options to display range ring & azimuth line references and county background maps. These attributes can be changed at any time, even while a product is being displayed. The following picture is an example of a background map display.

Range ring spacing is variable and depends upon both the product resolution and the current display zoom factor. Azimuth lines are drawn every 45 degrees. Note that range rings and background maps will not be correctly overlaid unless the product resolution has been correctly configured.



Highways and rail roads can be displayed at different levels of detail. See Part E of this document.

SPECIAL NOTE:

- For background maps to be properly displayed, the location of the radar antenna must be correct in the final product header. For products created in a development environment, this requires that the Site Adaptation Data must have been modified to reflect the source of the input radar data before the products were generated. See Document 5 Section II Part A of this volume for changing the site adaptation data.

- For both range rings and background maps to be properly displayed, the resolution must be configured correctly for the product being displayed.

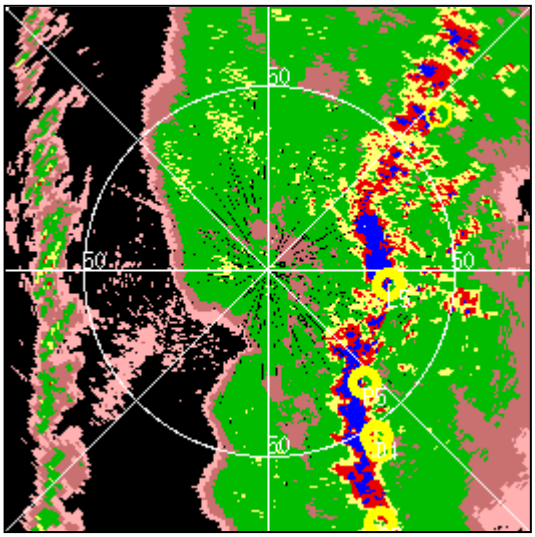
- When displaying elevation radial products the map overlay is useful only at lower elevations. Range error increases with the cosine of the elevation angle. This is because the displayed range is slant range rather than range across the earth's surface.

A few sample maps are created the first time CVG is launched. A new map creation utility **`map_cvg`** is now included with CVG. In addition, a script is provided that can create a few selected or all maps with a single command. See Volume 4 Document 5 for additional information.

Label Format Selection

The Label Format option menu in the **Display Attributes** pane provides options alter the display of overlay labels and range ring labels to improve legibility.

Transparent Background



Black Background

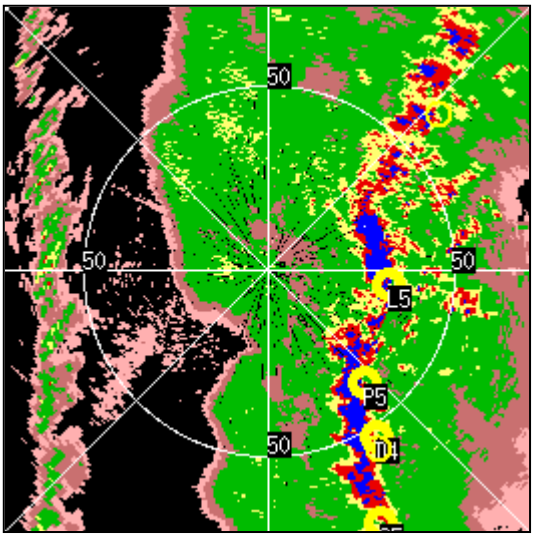


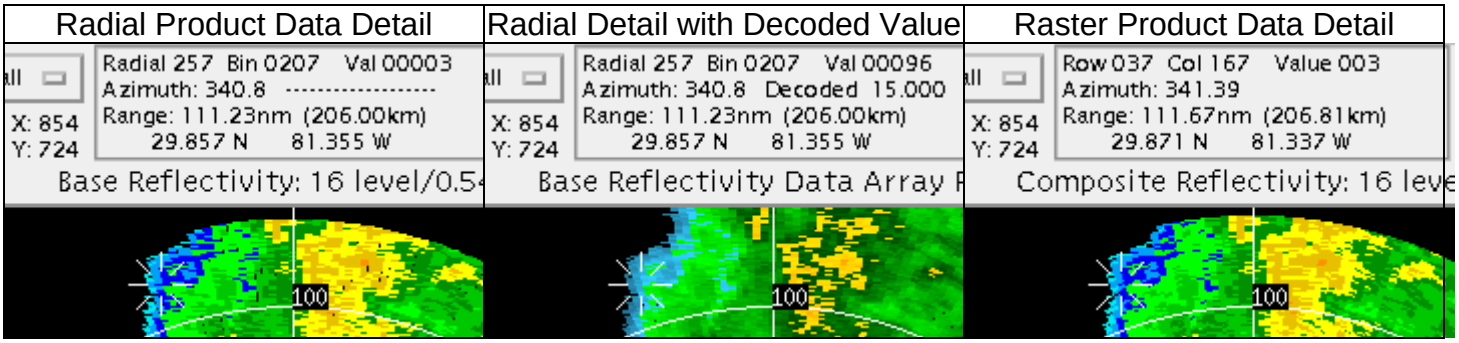
Image Size Selection.

The Image Size option menu in the **Image Control** pane allows selection of a **Small Image** (768x768 pixel screen size) instead of the default **Large Image** (1840x1840 pixel screen size). The selection is immediately applied to all product display screens.

Mouse-Click Data Query

If the image being displayed is a radial or raster data product, clicking the left mouse button will display

the location of the data under the cursor and the data value at that location. The data are displayed in the data pane at the top of the display window.



There are several points to note:

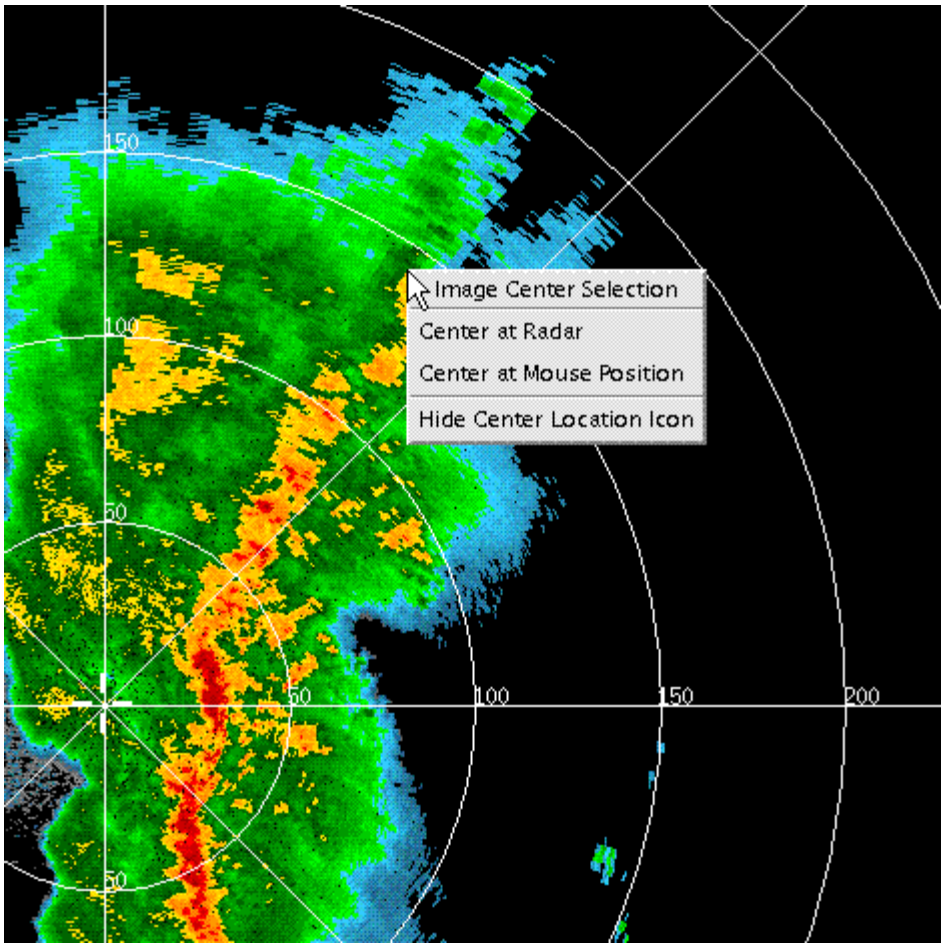
- The Latitude / Longitude and range / azimuth are displayed for all geographic products.
- The product resolution must have been configured correctly in order for the displayed Range: information to be correct.
- The product must have been created with the correct radar site adaptation data for the Latitude and Longitude to be correct.
- If a digital product (packet 16) and a generic radial product are configured for display using Method 5, which can use Scale Offset parameters, the decoded real value is displayed. See center example above.
- For polar coordinate (radial based) products:
    - The azimuth angle displayed for the radial product is the beginning azimuth not the center azimuth of the radial. This is taken directly from the header information in the data packet.
    - The range displayed is the near edge of the radial bin. Bin #1 has a range of 0.
- If an overlay product is displayed directly (i.e., no geographic radial / raster data underneath), only the azimuth / range and Latitude / Longitude are reported.
- For most non-geographic products no data are returned. One exception, the row, column, and data level are reported for the precipitation accumulation array in the DPA product.

Changing the Center of a Geographic Image

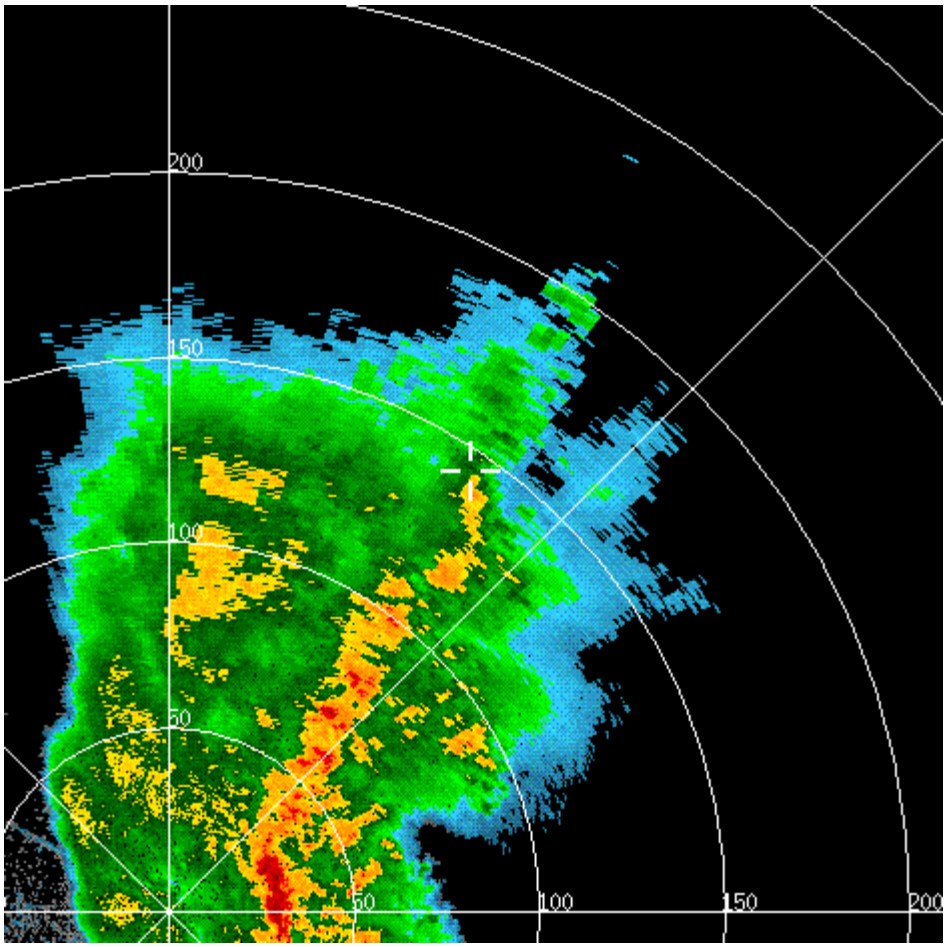
Normally when a geographic image is displayed the radar location is placed in the center of the image. Any geographic point can be selected to be displayed at the center of the image. This can be useful when inspecting a portion of the product near the edge (high range). This avoids having the area of interest off the edge of the image when increasing the zoom factor.

The Image Center Selection Menu pops up when clicking the right mouse button on any portion of a displayed geographic product.

- Either the current mouse pointer location or the radar location can be selected as the image center by choosing the appropriate menu item.
- The center location is indicated by a white plus symbol. The last menu option toggles the display of this symbol on and off.



The following image shows the image recentered on the selected geographic point. The selected location remains in effect during product animation, while changing the zoom factor, and when changing the display window or image size. The image is recentered at the radar when a new product is manually selected for display.

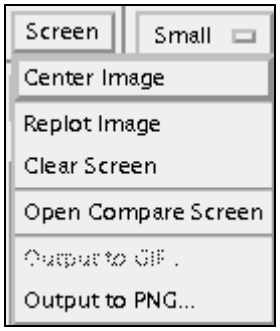


The Screen Menu Options

The **Screen** pull down menu on the Graphic Product Display Screen provides several options related to the displayed image.

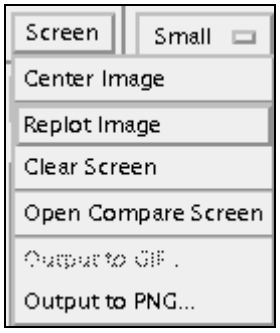
Centering the Image

The **Center Image** selection in the **Screen** pull down menu will redraw the product image including all overlays. This can be used to clear the contents of the data pane while retaining image display.



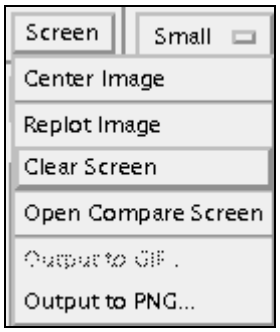
Replotting the Image

The **Replot Image** selection in the **Screen** pull down menu will redraw the product image including all overlays. This can be used to clear the contents of the data pane while retaining image display.



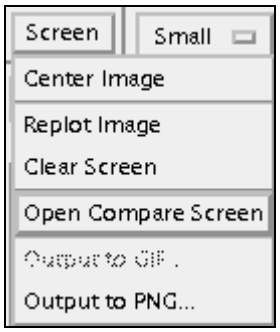
Clearing the Screen

The **Clear Screen** selection in the **Screen** pull down menu will clear the screen of the displayed image and erase / free all related internal data.

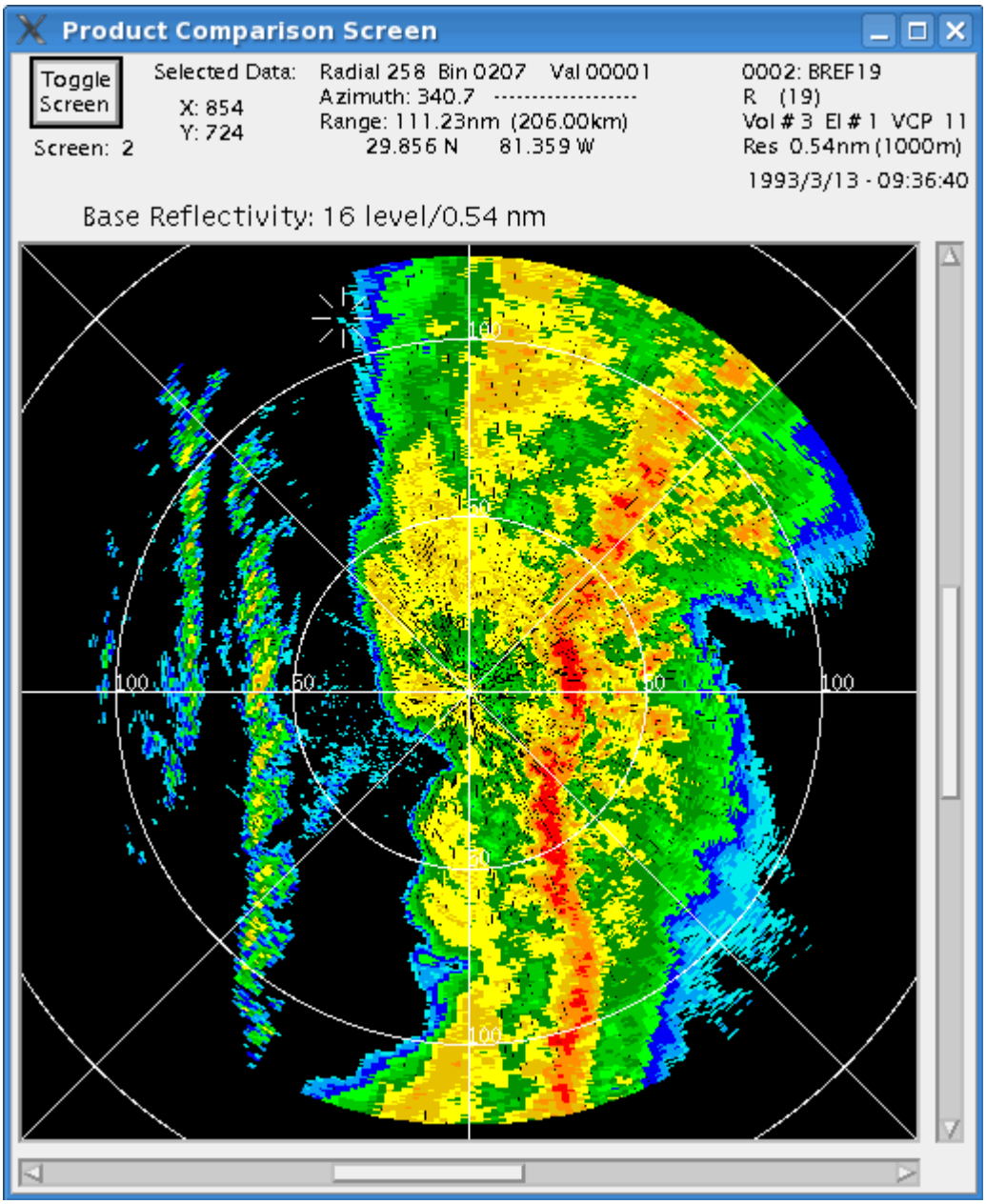


Using the Product Comparison Screen

A Product Comparison Screen can be opened with the fourth selection of the **Screen** pull down menu. **NOTE**: The Product Comparison Screen will not appear unless there is a product displayed in both graphic product display screens (1 and 2).

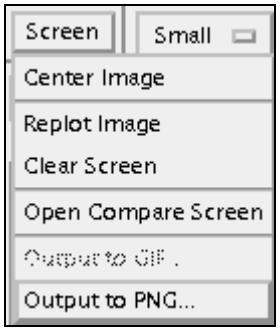


Once the product compare screen is opened, the product viewed is switched between the Screen 1 and Screen 2 via the **Toggle Screen** button. This permits rapid visual comparison of the two products.



Exporting GIF / PNG Images

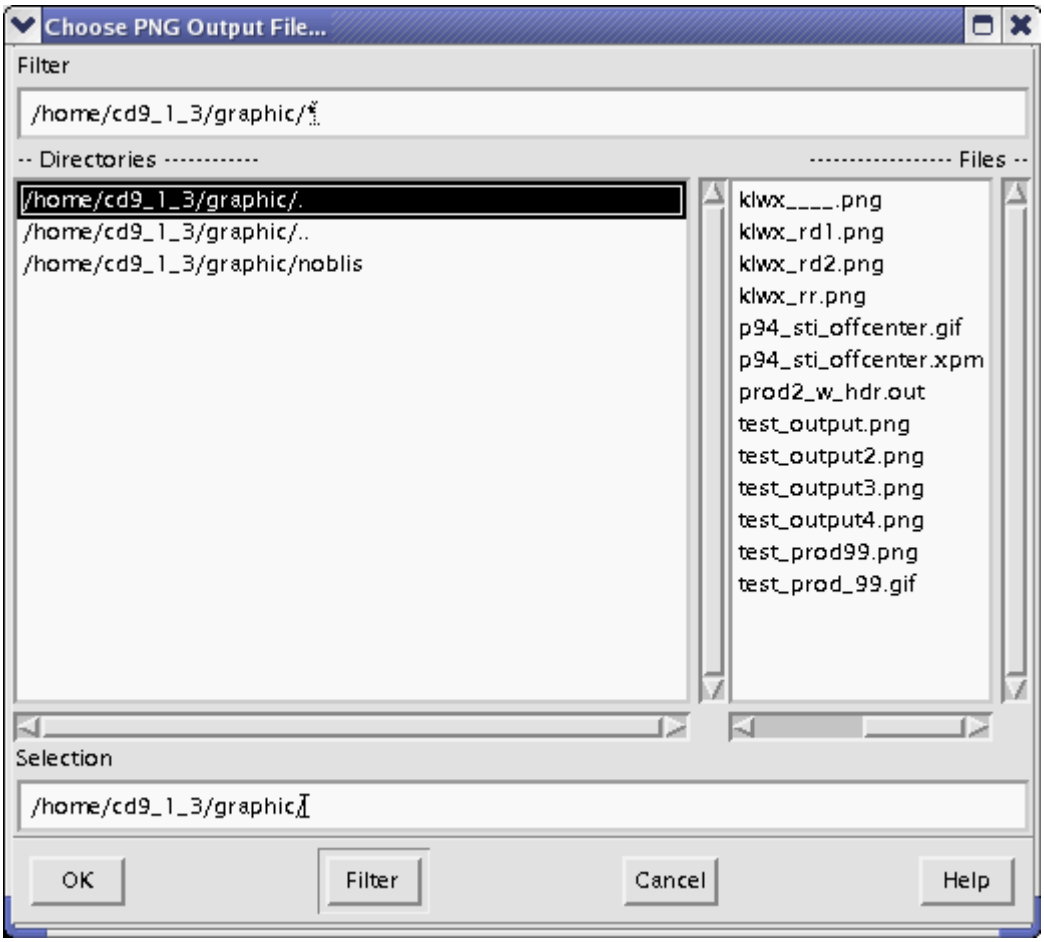
With graphical data displayed, the entire image from the graphic product screen can be exported to a binary image file with the **Screen** pull down menu.



The **Output to GIF...** option writes the contents of the image pane to a GIF formatted disk file. **This function is currently disabled.**

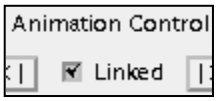
The **Output to PNG...** option writes the contents of the image pane to a PNG formatted disk file.

A standard file dialog is used to write the file to disk.



The exported image is based on the previously selected screen size (1840x1840 or 768x768). The smaller size generally provides a more pleasing image with the legend closer to the image.

Linking the Product Display Screens



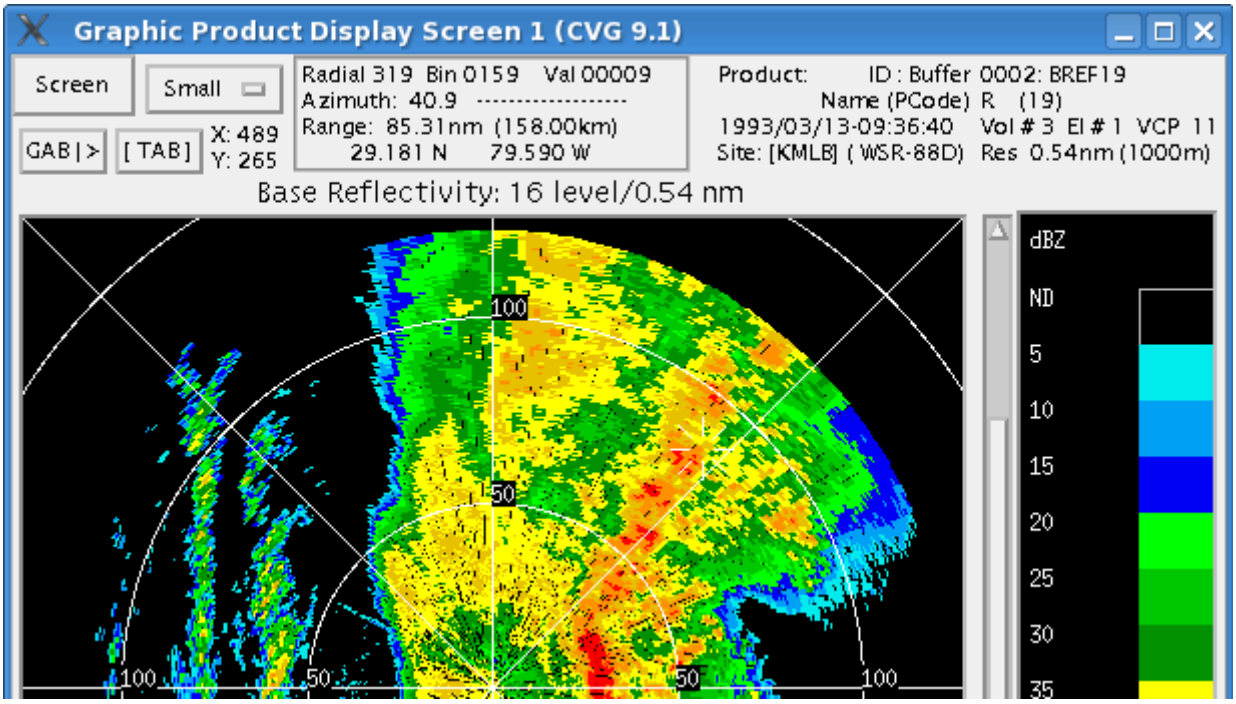
With two products displayed (see next section), checking the **Linked** check box links both screens for animation and scrolling. The third (auxiliary) screen cannot be linked. When linked:

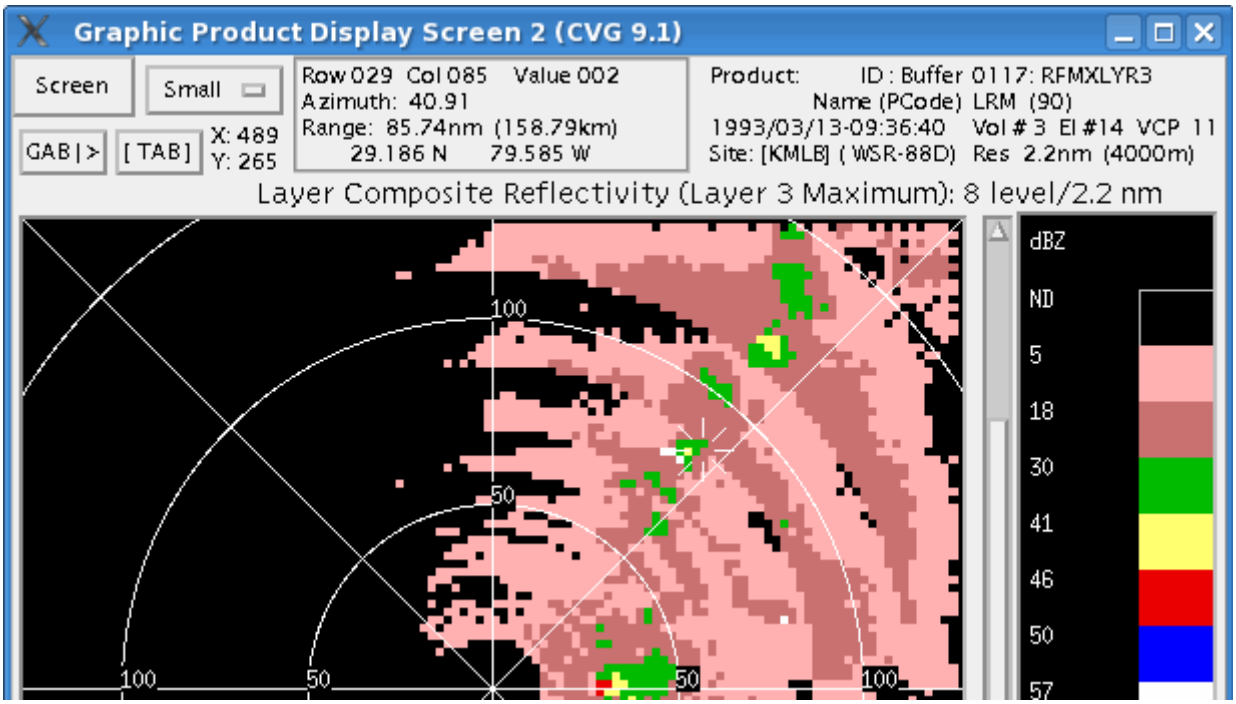
- Product animation can be controlled from either screen and animation occurs in both windows.
- Scrolling the product graphic display pane in screen 1 will also scroll the graphic display pane in screen 2.
- Selecting a geographic center for one image also changes the center of the linked image (spatial resolution must be the same).

----- CVG Main Window Functions -----

Displaying more than one product simultaneously

Once a product is displayed in Screen 1, a second product can be displayed in Screen 2 by clicking the **Screen 2** radio button and selecting a product for display.



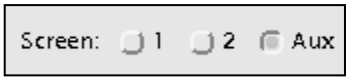


Note that the two images in the example have different resolutions. By adjusting the Zoom scale, they have been set to the same scale visually.

With two products displayed, an on-click data query accomplished by clicking on a product in either screen will result in a query being made on the closest matching location in the other screen (see example above).

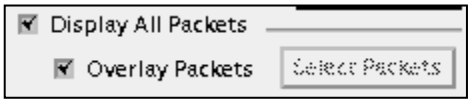
> For the two-display data query to compare data from the same location, the display scale must be the same. In other words, if displayed, the range rings of the two products would be the same size. Products with identical resolutions must be displayed with the same Zoom factor while products with different resolutions must be display with corresponding Zoom factors providing the same relative display scale (see example above).

A product can be displayed in an Auxiliary Screen by clicking the **Screen Aux** radio button and selecting a product for display. The auxiliary screen is not related to the other screens (not linked for data query or animation and not used by the compare screen).



Bypassing the packet selection dialog

If generally all portions of the selected product are to be displayed, the packet selection dialog can be skipped by checking the **Display All Packets** check box on the main window. This will display all graphic portions of the product including the GAB but will not display the TAB.

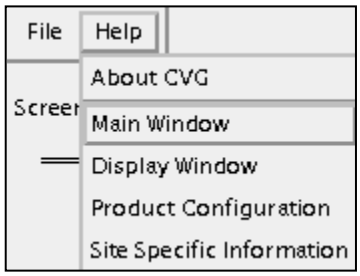


With Display All Packets selected the Overlay Packets check box is active and can be used to control product overlay.

On-line Help

The main application **Help** pull down menu provides on-line help for the **Main Window**, the **Display Window**, the **Product Configuration** preferences screen, and the **Site Specific Information** preferences screen.

Additional on-line help is provided via a **Help** button on individual dialog screens.



Part D. Displaying Products not in the ORPG Database

Displaying a product in an alternate product database

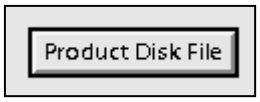
CVG automatically uses the ORPG product database configured for the ORPG account in which the user is logged on. This feature can be overridden by selecting a different database file in via the **Preferences...** option from the **System** pull down menu. See Part E. CVG Preferences - Basic CVG functional preferences. Note: leaving this item blank in the preferences edit dialog will cause **CVG** to use the configured database when next launched.

One use for this feature is that an ORPG can be run with specific input data and developmental products collected. This database linear buffer file can then be copied and saved for later analysis.

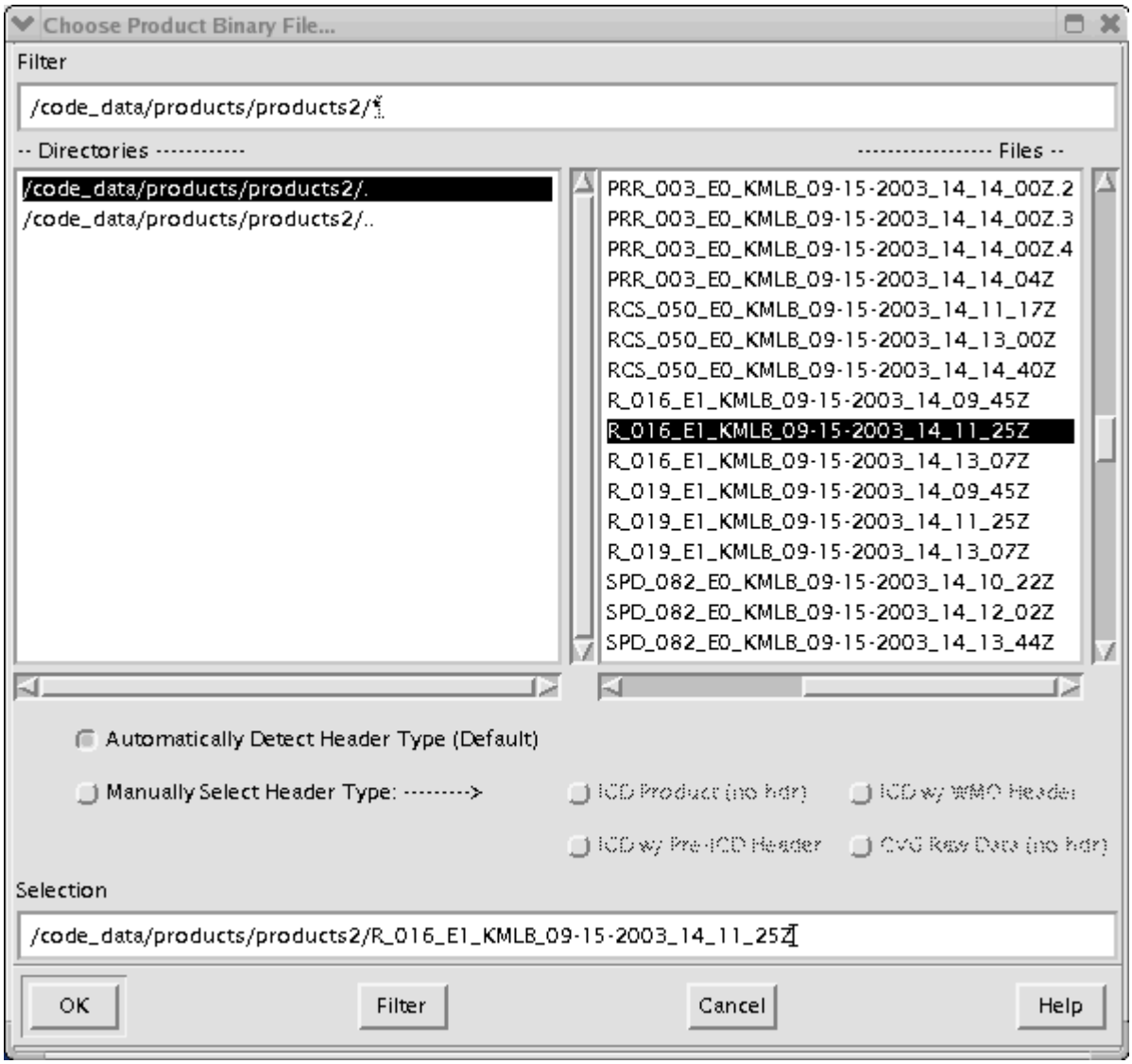
NOTE: As stated previously, the ORPG Build number should be set to match the source of the product database file. The **Product from ORPG** option menu is used to set the build number. This is critical if reading a product database created with an ORPG prior to Build 6.

Displaying a product in an individual binary disk file

The **Product Disk File** button is used to select a binary disk file for product display.



A standard file dialog is displayed. Using this dialog, select the binary disk file containing the product data. When this feature is first used, **CVG** will open the dialog to the current working directory.



The binary disk file must contain one product in one of three formats. CVG automatically detects whether an additional header is present in the file. If for some reason this automatic detection fails, the user can manually select the header type included with the Manually Select Header Type radio button. This activates three additional radio buttons for specifying the header type:

- ICD Product (no hdr) - A standard final product (ICD format) with no additional header included. This is the format in which the ORPG distributes the product.

- ICD w/Pre_ICD Header - A final product (ICD format) following an internal 96 byte (pre-ICD) header. This format can be produced with the CVT utility.

- ICD w/ WMO Header - A final product (ICD format) following a WMO header. This format is obtained from the WSR-88D central product server (e.g., the NCDC web site).

- A fourth format called CVG raw data is no longer supported.

Displaying a product in an ORPG intermediate linear buffer file

The capability to select an intermediate product for display was only a demonstration feature for CVG.

This capability is currently disabled and may be revived if intermediate products are produced in a standard format.

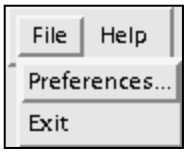
Part E. CVG Preferences - Basic CVG functional preferences

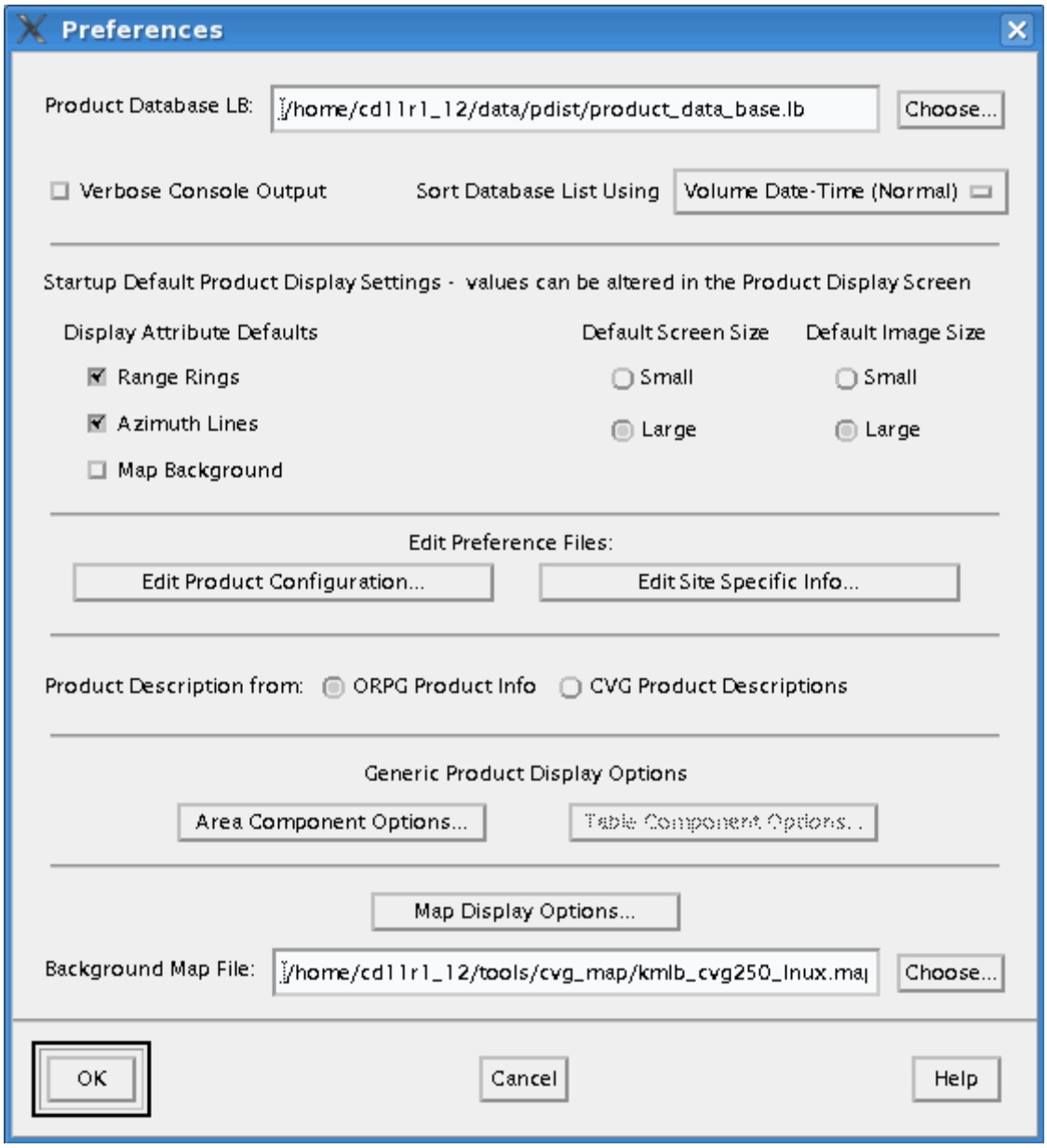
When CVG is launched, if a local directory **`$HOME/.cvgN.N`** (where **`N.N`** is the CVG version number) does not exist, it is created and the default configuration files are copied from their installed location into this local configuration directory for use by CVG. These local files can be replaced by the defaults by (1) renaming or deleting the local configuration directory (**`~/.cvgN.N`**) and (2) launching **`CVG`**. This is useful if the configuration files have become corrupted.

The preferences menu provides a basic capability of avoiding a manual edit of these files. The only exception is the **`prod_db_size`** configuration file which can be used to set the size of the database product list. See *Product List Size* at the end of this document.

Main Preferences Menu

The main preferences dialog is selected by choosing the **Preferences...** option from the **File** pull-down menu on the **`CVG`** main window.



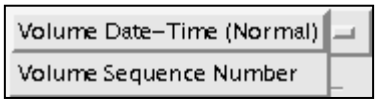


After changing any preferences, the **OK** button must be clicked to apply the changes.

The **Product Database LB** edit box is used to select an alternate product database linear buffer file.

The **Choose...** button brings up a standard file dialog window. **NOTE: You must have both read and write permissions for the product database linear buffer file.**

The **Verbose Console Output** check box is used to provide a more verbose information output in the terminal window from which **cvg** was launched. Though much of this information is not of interest to an algorithm developer, it can be useful when configuring a new product.



The Sort Database List Using... option menu provides a choice in the primary index used for sorting the list. The standard order is using Volume Date-Time. An alternate order using the Volume Sequence Number is sometimes useful if ingesting data sets from different radars or sets containing non-contiguous data.

The Startup Default Product Display Settings pane selects the initial settings for the following display settings when CVG is launched. These values can be altered from the Product Display Screen.

The **Display Attribute Defaults** check boxes control the default behavior for Range Rings, Azimuth Lines and Map Background overlay.

The **Default Screen Size** toggle

The **Default Image Size** Toggle

The **Product Description From...** selector buttons determine the source of product descriptions and mnemonics used in the product list and legend areas. ORPG Product Info uses the contents on the ORPG account's product information buffer which is derived from the **`product_attr_table`** configuration file (including any snippets). CVG Product Descriptions uses the contents of the CVG configuration file '**`prod_names`**' in the **`~/.cvgN.N`** directory.

The Edit Preference Files pane provides dialogs to edit two CVG configuration files. The **Edit Product Configuration...** and **Edit Site Specific Info...** buttons open the following additional preferences dialogs.

Product Configuration Dialog

The purpose for the Product Configuration dialog is to provide the required information for **`CVG`** to correctly handle certain aspects of product display. When displaying a new development product, this dialog is used to add the preferences for a new product ID.

Clicking the **Edit Product Configuration...** button on the main preferences dialog brings up the Product Configuration Edit dialog window.

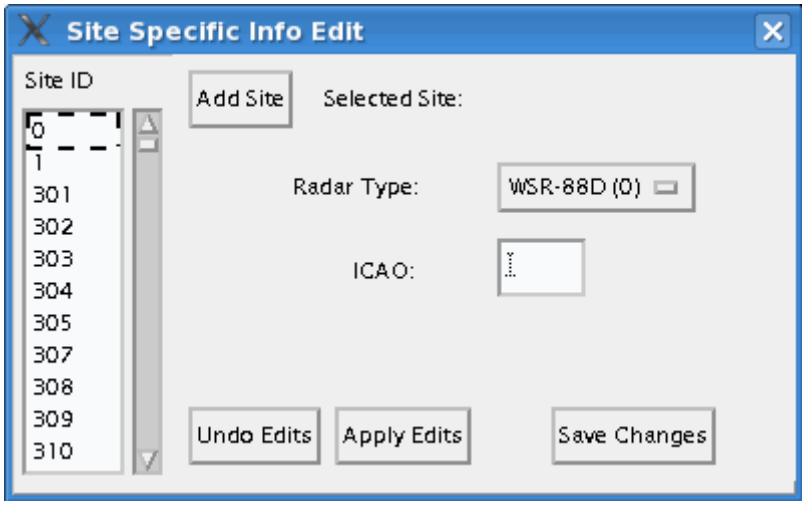
Procedures for the configuration of products for display by CVG are provided in Section II of this document.

Site Preferences Dialog

Currently, the only purpose for Site Preferences is to permit the radar type and site ID to be displayed with the product. If the site adaptation data has been updated to reflect the source of the radar data being ingested, the RPG ID is used to display the correct ICAO identifier.

Since all operational WSR-88D sites are already listed in the site data preferences file, there is little need to edit this data. *However, if using data not obtained from an operational WSR-88D site, see the Note below.*

Clicking the **Edit Site Specific Info...** button on the main preferences dialog brings up the Site Specific Information Edit dialog window.



Site Adaptation Data Note

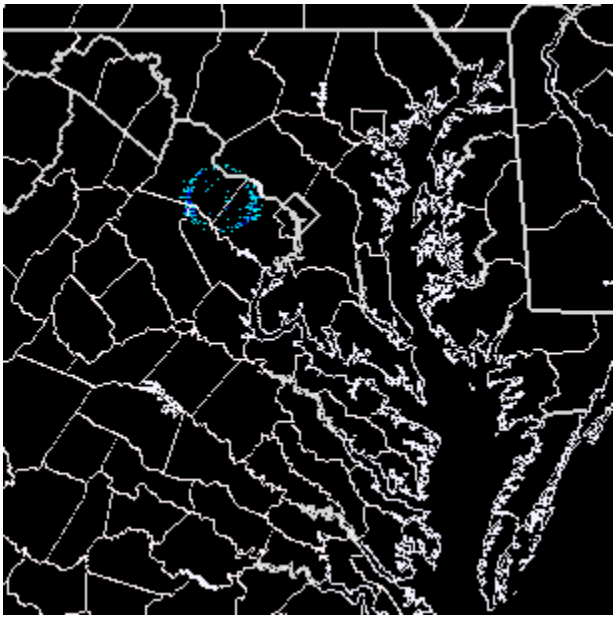
All operational WSR-88D site adaptation data is included with the ORPG. However, if using radar data or products obtained from non-operational sites, non-WSR-88D sites, or experimental sites, the basic adaptation data should be added to the **`change_radar`** data file.

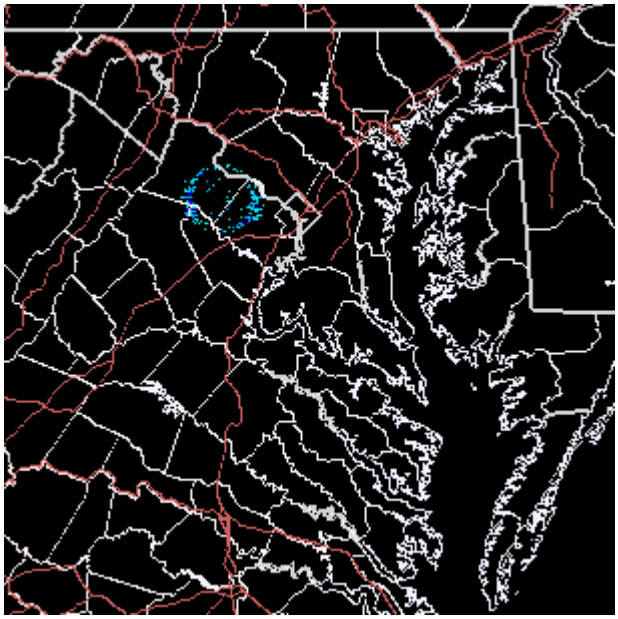
The site data is required in order for the radar ID and type to be displayed with the product and in order for the background map created for that site to be displayed. See Document 5 Section II Part A of this volume.

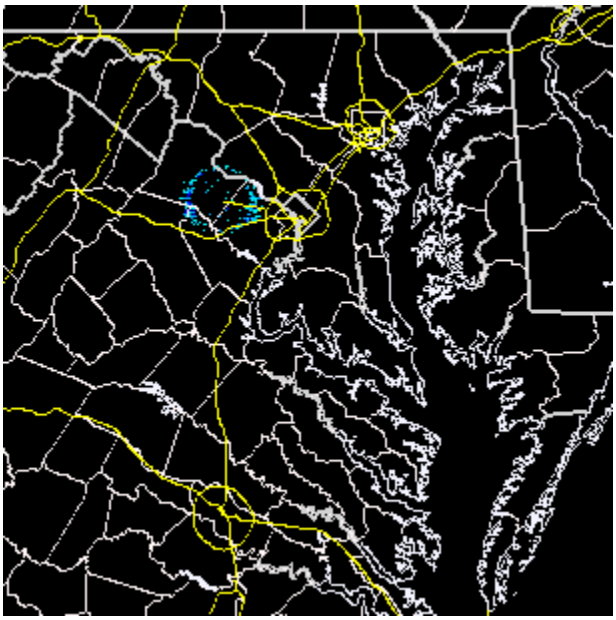
Background Map Options

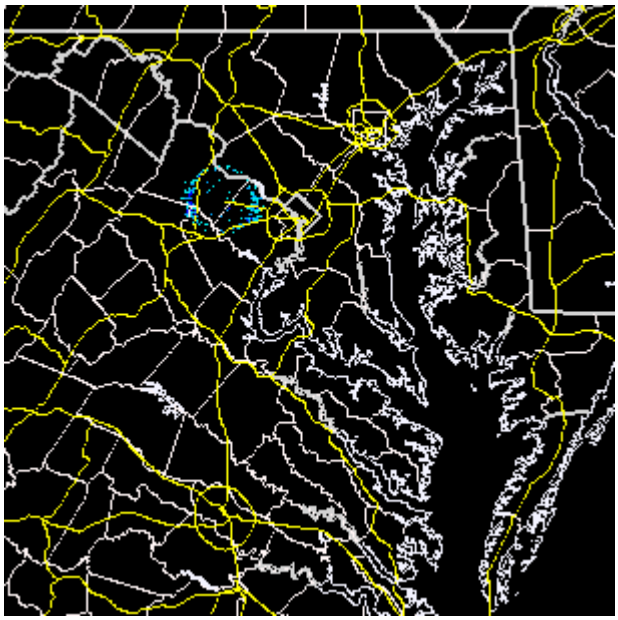
Clicking on the **Map Display Options...** button opens a dialog that permits selection of the number of highways and rail roads to include in the background map.

The following is an example of the level of detail.


No highways, no rail roads


Major Rail Roads


Major Highways


More Highways

The **Background Map File** edit box determines the background map file to be used. Currently, only maps for a few sites are included with the ORPG source code. They are installed in the **`~/tools/data`** directory. CODE includes a few additional maps and a Background Map Utility for creating maps.

The **Choose...** button brings up a standard file dialog window.

Product List Size

`CVG` preferences are stored in text configuration files in the **`~/.cvgN.N`** directory. The preferences menu provides a basic capability of avoiding a manual edit of these files. The only exception is the **`prod_db_size`** configuration file which can be used to set the size of the database product list. The default **`CVG`** list size is 12,000 entries and can be adjusted from 12,000 - 32,000. **This does not change the size of the ORPG product database.** The size of the database is determined by its configuration in the **`data_attr_table`** file. The CVG list must be at least as large as the database size.

Volume 4. Document 2. -
CODEview Graphics (CVG)

Section II Configuring Products for Display by CVG

CODEview Graphics 9.2

This document covers the following topics.

- **Part A. Product Specific Preferences** Using the Product Configuration Edit dialog to configure products for display.
- **Part B. CVG Color Palettes** A description of some of the palettes provided and procedures for creating new palette files.
- **Part C. Scaled Integer Products - Color Palettes & Threshold Labels** An introduction to the legend configuration methods for generic radial components and digital products (data packet 16) and how color palette and digital legend files are used with these products.
- **Part D. Scaled Integer Products - Configuration for Display** A description of the structure of the generic legend configuration file and the digital legend configuration files and guidance for corresponding color palette files.

For a list of recent changes implemented in CVG see Section I of this document.

When CVG is launched, if a local directory **`$HOME/.cvgN.N`** (where **`N.N`** is the CVG version number) does not exist, it is created and the default configuration files are copied from their installed location into this local configuration directory for use by CVG. These local files can be replaced by the defaults by (1) renaming or deleting the local configuration directory (**`~/.cvgN.N`**) and (2) launching **`CVG`**. This is useful if the configuration files have become corrupted.

Part A. Product Specific Preferences

This section of the document describes using the Product Configuration Edit dialog window to configure products for display. All other preferences are documented in Section I Part E. of this document.

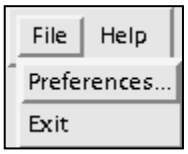
Build 12 Notes:

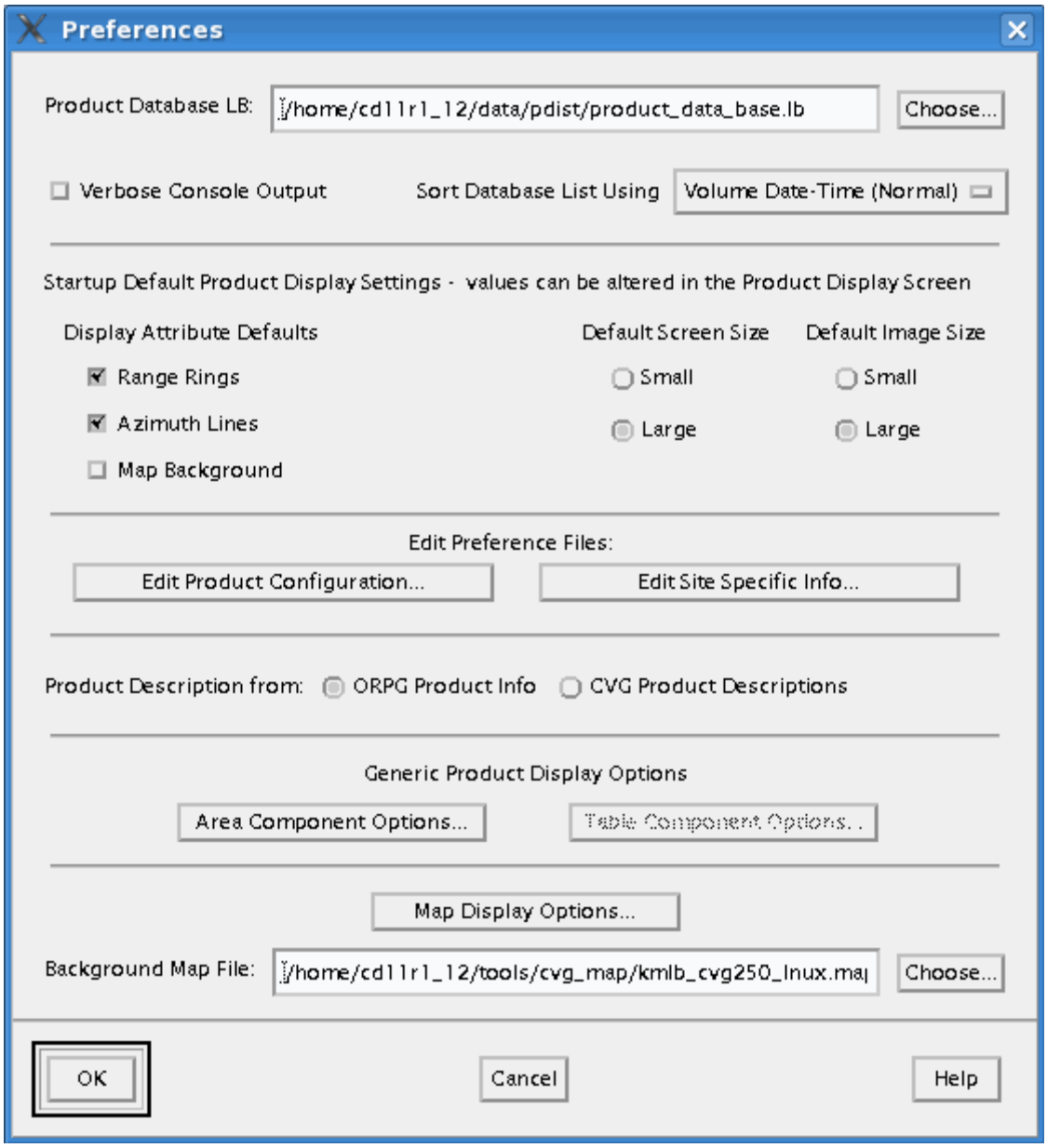
WARNING: With the update to CVG 9.1 provided with Build 12 CODE, the format of the prod_config file has been changed to include a new entry for the packet 1 geographic coordinates override and a capability to override the default colors of a second data packet. Any development entries in the prod_config file from prior versions of CVG cannot be used / inserted into the CVG 9.1 version of the file. The Product Configuration Dialog should be used to add / modify entries rather than manually editing the prod_config file.

Product Integration Note: If a development product is being sent to the NWS Radar Operations Center (ROC) for integration into the operational system, the CVG configuration information contained in the configuration **`prod_config`** file, and a copy of newly developed legend configuration files and color palette files should be sent to the CVG Development Lead: Brian Klein **`(brian.klein@noaa.gov)`**.

Main Preferences Menu

The main preferences dialog is selected by choosing the **Preferences...** option from the **File** pull-down menu on the **CVG** main window.



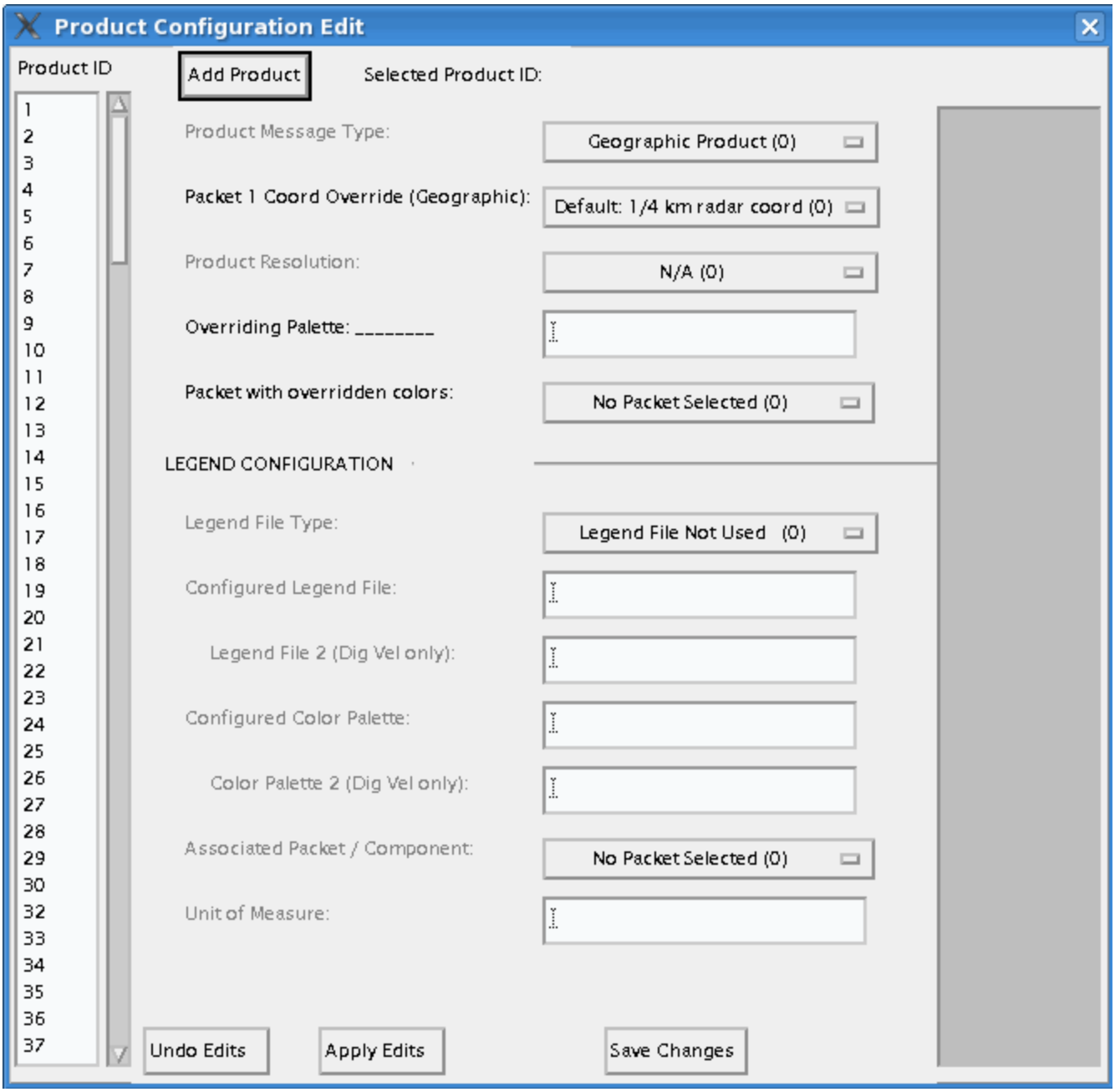


After changing any preferences, the **OK** button must be clicked to apply the changes.

Product Configuration Menu

The purpose for the Product Configuration dialog is to provide the required information for **CVG** to correctly handle certain aspects of product display. When displaying a new development product, this dialog is used to add the preferences for a new product ID. It is also used to modify the display configuration of existing products.

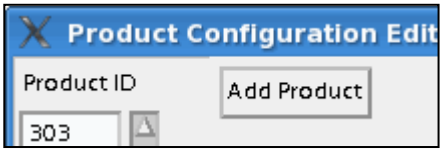
This dialog safely edits the main product configuration file **prod_config**. While **prod_config** can be edited manually, it is recommended this dialog be used to reduce errors.



Clicking the **Edit Product Configuration...** button on the main preferences dialog brings up the Product Configuration Edit dialog window.

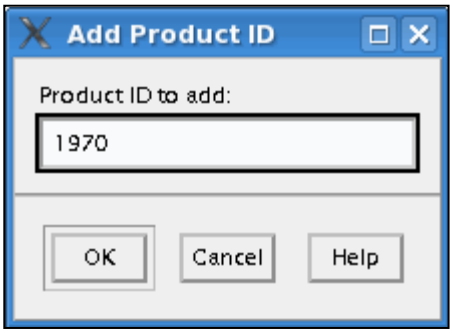
To add a new product to the product list:

1. Press the **Add Product** button.

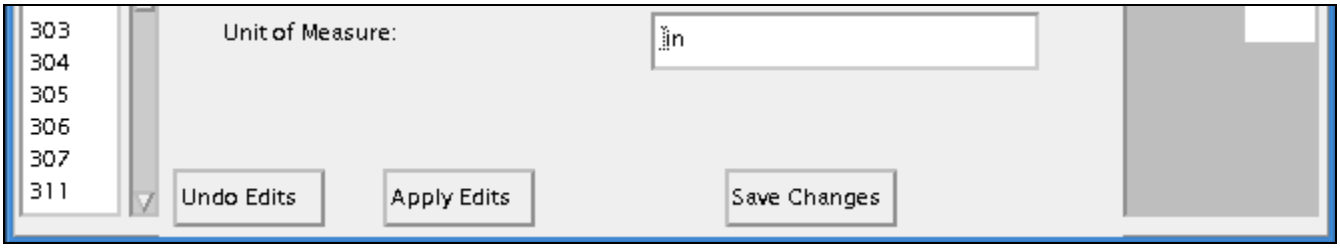


2. Enter a new product ID in the resulting dialog and press **OK**. This product ID (linear buffer number) must not be used by any other product.

   Note: if the ID is already configured in CVG an error pop-up is displayed after pressing **OK**.



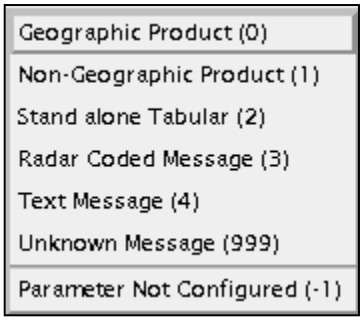
The proper sequence to enter / change preferences is:



1. Select the product ID from the **Product ID** list.
2. Edit the individual preferences as described below. At any time prior to pressing the **Apply Edits** button, pressing **Undo Edits** will restore the original configuration.
3. Press **Apply Edits** to save to memory. In addition, the preview pane for the Digital Legend or the Configured Palette is updated if appropriate. At this point CVG will use the new settings during the current session only.
4. Press **Save Changes** to save to disk. This modifies the configuration file.

Product Configuration Parameters

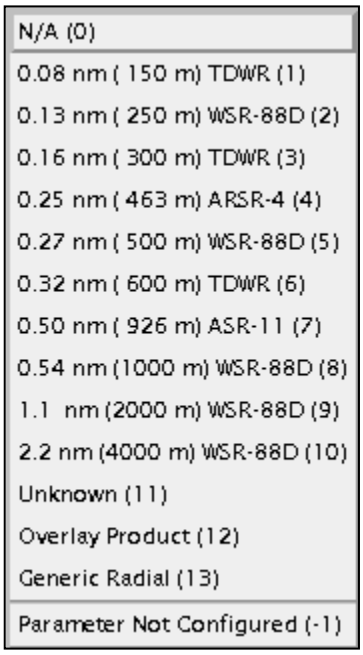
The following configuration parameter select menus and edit boxes are contained in the Product Configuration Edit dialog window. Configuration examples are provided just after the explanation of each parameter. The number in parentheses in the following option selection menus is stored in the CVG product configuration file **'prod_config'**.



Product Message Type: This parameter reflects the format and type of data in the WSR-88D final product. The value for this parameter is critical for correct product display. The first two types listed are graphic products which comprise the vast majority of WSR-88D products.
A message type of **`Geographic Product (0)`** indicates a graphic product with features that correspond to a two-dimensional *geographic* position with respect to the radar antenna. The majority of WSR-88D products are of this type. The second most common message type is **`Non-Geographic Product (1)`**. One example of non-geographic products are cross-section products which are considered *non-geographic* because only one dimension is across the surface.

The default value of '`Parameter Not Configured (-1)`' must be changed, otherwise CVG will display an Error message and abort product display. See on-line help for other message types are.

Packet 1 Coord Override (geographic) This parameter overrides the default coordinates for the placement of data packet 1 in a geographic product. The default coordinates are 1/4 km from the radar location. Several products use coordinates in pixels from the upper left corner of the screen. This option only applies to geographic products (0).



Product Resolution: This parameter represents the horizontal resolution of the product and is used for geographical products and geographical overlay products. The configured resolution affects the drawing of overlays (including background maps and range rings) and the reporting of product data bin information when clicking on a geographic product.

Geographic products (with the exception of overlay products) must be configured with the correct resolution (**`1-10`**).

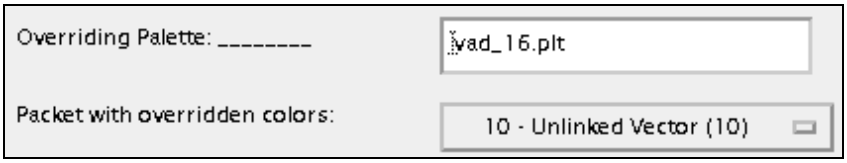
Many products (for example the Mesocyclone product) are intended to be displayed on top of a 2 dimensional geographic image (radial or raster). These products should be configured as **`Overlay Product (12)`**. When displayed over 2 dimensional geographic products, overlay products use the resolution of the underlying product. When displayed stand-alone, an inherent resolution of 1 KM is used.

For most products other than geographic, the resolution is set to **`N/A (0)`** which represents no resolution.

Generic geographic products (the generic radial component and the generic area component) have the horizontal resolution included in the product so a specific resolution is not specified. For the generic radial component the resolution **`Generic Radial (13)`** is must be selected. The generic area component is configured with the resolution **`Overlay Product (12)`**.

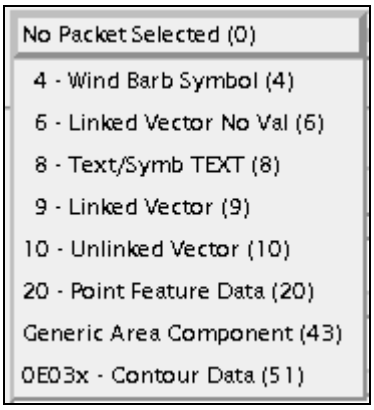
The default value of '`Parameter Not Configured (-1)`' must be changed, otherwise CVG will display an Error message and abort product display. Other resolutions are documented in the on-line help.

Overriding Palette This specifies the filename of the CVG color palette file used to override the selected packet's default colors.



These files are located in the local **`CVG`** configuration directory **`~/.cvgN.N/colors`**. **If not configuring a palette file, the value "`.plt`" must be entered as the filename.** See Parts C. and D. of this document for information on creating and selecting a color palette.

Packet with overridden colors This parameter determines to which data packet within the product the overriding color palette is applied during display and overrides default color palette.



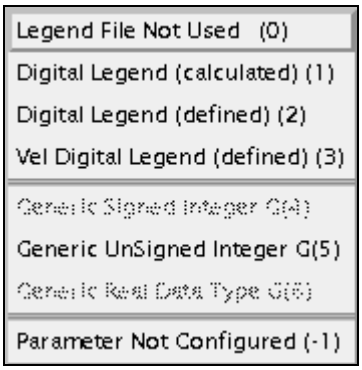
Some of the packets listed (4, 6, 8, 9, 10, 20) are normally displayed using the default palette. Caution should be used when overriding these default colors because they are based on the officially assigned colors in the Product Specification ICD. Sometimes a new product will require the defaults to be modified based upon the needs of the NWS. If creating a new product for integration into the operational system, the details must be coordinated during a design review.

Packet 20 is unique in that new symbols can be defined for a new product. This is normally done via coordination of the using organization (e.g. NWS) and will require modification of CVG (contact the CVG Development Lead: Brian Klein: **`brian.klein@noaa.gov`**). Existing symbols should be used with caution but if used different colors could be assigned for the new product.

The Generic Area Component (43) is a relatively new packet. Configuration of the lines and symbols has not been formalized and is still accomplished manually during the display of the product (see *Generic Product Display Option Menu* later in this section). Use of this packet in a product intended to be integrated into the operational radar will require coordination and approval during the design review process. A method needs to be developed in the CVG logic for display of the area component for selecting colors from a packet (contact the CVG Development Lead: Brian Klein: **`brian.klein@noaa.gov`**).

The packet for Contour Data is a special case and not covered by this document. It is currently used by one product.

Legend File Type: This parameter specifies the method of deriving data threshold level information that is displayed in the CVG product legend next to the color bars. It also affects color assignments from the configured color palette file. **A legend configuration is only applicable to products containing 2 dimensional data arrays.**



A legend file type of **`Legend File Not Used (0)`** is used for any product that is not either a digital product (using data packet 16 or 17) or a generic radial component. This includes products that were designed for display in the original radar system using packet AF1F for radial data and packet BA07/BA0F for raster data. These products have the individual data level threshold values recorded explicitly in the product and do not use a legend configuration file.

The recently developed type: `Generic Unsigned Integer G(5)`, referred to as "method 5", is the most universal method and is recommended for use with all digital products using data packet 16 and products using the generic radial component. This legend configuration method provides the options for explicitly defined labels or labels calculated using the Scale Offset parameters. In the future, support for signed integer arrays (method 4) and arrays of real data types (method 6) will be added. See Part C of this document for advantages for using this method.

There are three additional legend configuration methods that can be used with digital products using data packet 16 and 17. **`Digital Legend (calculated) (1)`** (method 1) is intended for a quick and easy configuration of new development products. The legend labels are calculated from parameters provided in the file and can only be used if the data encoding follows several specific rules. **`Digital Legend (defined) (2)`** (method 2) is intended for final configuration of a product. The legend labels are explicitly defined in the legend configuration file. **`Vel Digital Legend (3)`** (method 3) is similar to method 2 but is used for digital velocity products which are configured with two explicitly defined legend definitions and two color palettes. See Part C of this document for an expanded description of these configuration methods.

The default value of '`Parameter Not Configured (-1)`' must be changed, otherwise `CVG` will display an Error message and abort product display.

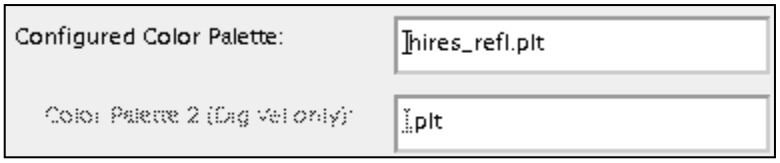
Configured Legend File: This specifies the filename of the legend configuration file for digital and generic radial geographic products. These files are located in the local **`CVG`** configuration directory **`~/.cvgN.N/legends`**.

If a legend file is not used, the value of "`.lgd`" must be entered as the filename.

If the Legend File Type is set to `1`, `2`, `3`, or `5` and no legend filename is entered, `CVG` will display an Error message and abort product display. See Part D. of this document, *Scaled Integer Product Display - Configuration for Display*, for additional information.

> For the two-dimensional data array packets, the legend thresholds configured here are called "Development Thresholds". In the future CVG will provide a capability to use the "AWIPS Default Thresholds". Some of the existing "Development Thresholds" are the same as those contained in the Product Specification ICD, including all of the RLE raster and radial packets (AF1F, BA0F, BA07).

Legend File 2: Only used for a digital (8-bit) velocity product. **If the Legend File Type is set to `3` and a second legend filename is not entered, `CVG` will display an Error message and abort product display.**



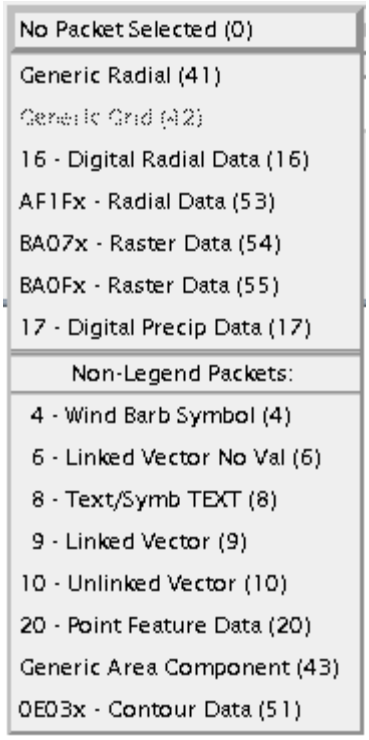
Configured Color Palette: This specifies the filename of the CVG color palette file used when displaying the product. These files are located in the local `CVG` configuration directory `~/.cvgN.N/colors`. Some overlay data packet types (i.e., text & special symbols and wind barbs) have a default palette that normally should not be overridden. **(CVG 9.1)** However, the display of geographical two-dimensional data packets (radial/raster run length encoded data, digital data arrays, or generic radial components) require a Configured Color Palette.

> For the two-dimensional data array packets, the colors configured here are called "Development Colors". In the future CVG will provide a capability to use the "AWIPS Default Colors". Some of the existing "Development Colors" are the same as those contained in the Product Specification ICD, including all of the RLE raster and radial packets (AF1F, BA0F, BA07).

If not configuring a palette file , the value "`.plt`" must be entered as the filename. See Parts C. and D. of this document for information on creating and selecting a color palette.

Color Palette 2: Only used for a digital (8-bit) velocity product. **If the Legend File Type is set to `3` and a second palette filename is not entered, `CVG` will display an Error message and abort product display.**

Associated Packet Type: If non-zero, this parameter determines to which data packet within the product the configured color palette is applied during display and overrides any existing standard color associations.



The default value is No Packet Selected `(0)`. **(CVG 9.1)** If zero, the configured palette is not applied to any packet in the product (before CVG 9.1 zero resulted in the palette being applied to all packets in the product).

The first seven packets listed are the two-dimensional data array packets which will have a legend color bar and threshold labels displayed.

```
Generic Radial (41)
Generic Grid (42) - not yet supported
16 - Digital Radial Data (16)
AF1Fx - Radial Data (53)
BA07x - Raster Data (54)
BA0Fx - Raster Data (55)
17 - Digital Precip Data
```

(CVG 9.1) These packets must have their colors set by a configured palette since they do not have any default color palettes associated (before CVG 9.1 the two-dimensional arrays did have default color palettes).

> For the two-dimensional data array packets, the colors configured here are called "Development Colors". In the future CVG will provide a capability to use the "AWIPS Default Colors". Some of the existing "Development Colors" are the same as those contained in the Product Specification ICD, including all of the RLE raster and radial packets (AF1F, BA0F, BA07).

The remaining part of the list, the Non-Legend Packets, is identical to the **Packet with overridden colors** list. Changing the default packet for the Non-Legend Packets listed is normally accomplished via the **Overriding Palette** and the **Packet with overridden colors** parameters described above. They are also listed here to provide the opportunity to modify two packets' colors even if the product does not contain a two- dimensional array packet.

Some of the packets listed (4, 6, 8, 9, 10, 20) are normally displayed using the default palette. Caution should be used when overriding these default colors because they are based on the officially assigned colors in the Product Specification ICD. Sometimes a new product will require the defaults to be modified based upon the needs of the NWS. If creating a new product for integration into the operational system, the details must be coordinated during a design review.

Packet 20 is unique in that new symbols can be defined for a new product. This is normally done via coordination of the using organization (e.g. NWS) and will require modification of CVG (contact the CVG Development Lead: Brian Klein: `brian.klein@noaa.gov`). Existing symbols should be used with caution but if used different colors could be assigned for the new product.

The Generic Area Component (43) is a relatively new packet. Configuration of the lines and symbols has not been formalized and is still accomplished manually during the display of the product (see *Generic Product Display Option Menu* later in this section). Use of this packet in a product intended to be integrated into the operational radar will require coordination and approval during the design review

process. A method needs to be developed in the CVG logic for display of the area component for selecting colors from a packet (contact the CVG Development Lead: Brian Klein: **brian.klein@noaa.gov**).

The packet for Contour Data is a special case and not covered by this document. It is currently used by one product.

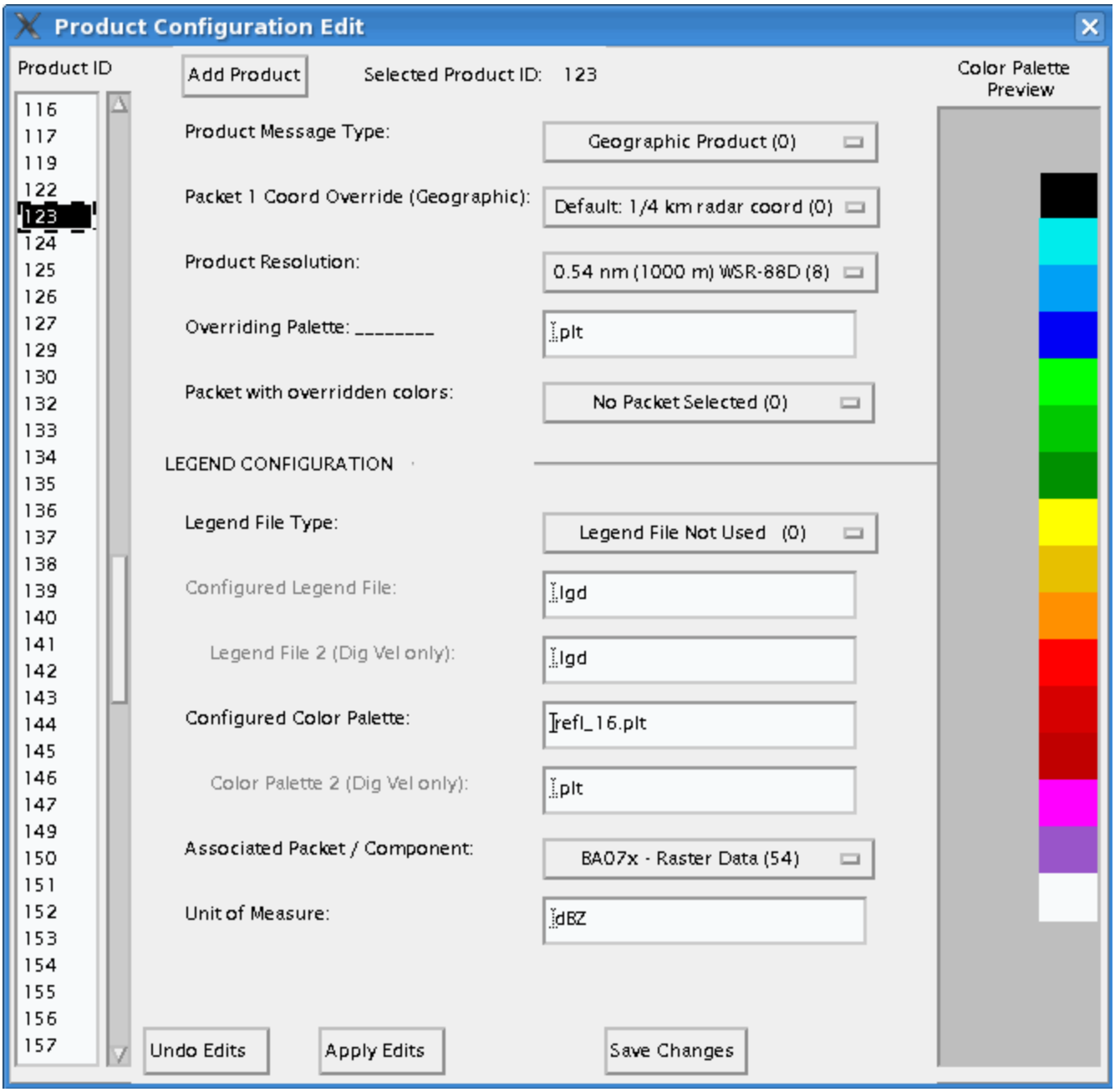
Data packets not listed in this option menu cannot be configured. If supported by CVG the inherent color association (default color palette) is always used. This includes text packets 1 and 2, mesocyclone data packet 3, vector arrow data packet 5, unlinked vector no value packet 7, correlated shear data packet 11, TVS data packet 12, storm id packet 15, hail data packet 19, etc.

Unit of Measure: The string, for example "**dBZ**" will be the units displayed for the threshold values listed in the product display legend. Any short string of characters (including spaces) can be entered. Examples of existing products: "**dBZ**", "**dBA**", "**in**", "**inches**", "**inches/hr**", "**kt**", "**kt rms**", "**deg**", "**deg/km**" "**kg/m^2**", "**m/s**", "**Prob Clutter**", "**E-4/**s". The default value is the string "**units**". The unit of measure is typically blank for products not containing radial or raster two dimensional data packets. For products that are not displayed by CVG (e.g., intermediate products) a short description has been included.

SPECIAL NOTE: Radial and raster run length encoded products (data packets AF1F, BA0F, BA07) must use a palette with the same number of colors as data levels in the product. The use of CVG color palette files with Scaled-Integer products is covered in Parts C and D of this document.

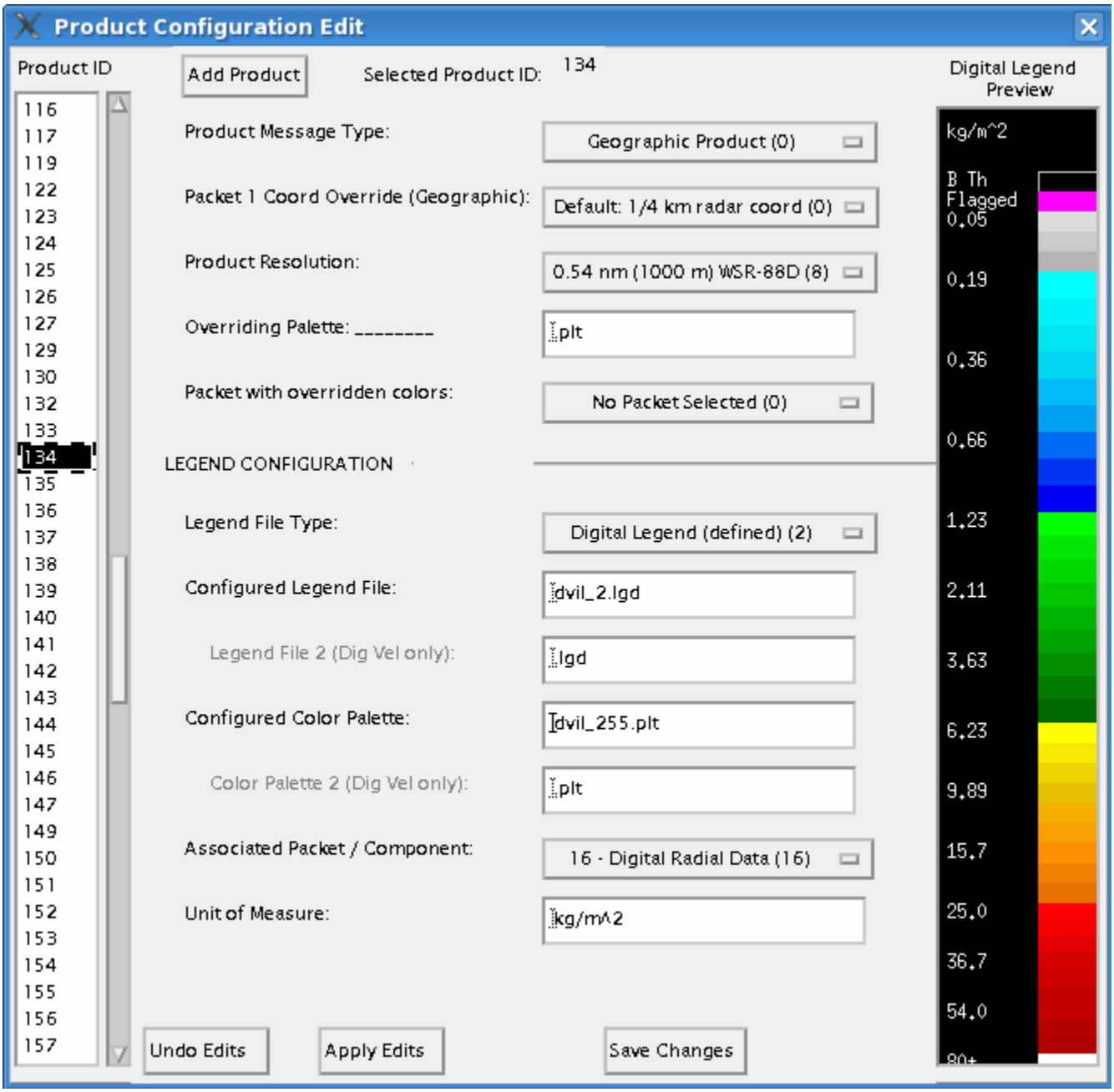
Product Configuration Examples (CVG 8.7 Edit Window)

Product Preferences Example 1 - Original RLE Product (non-digital)



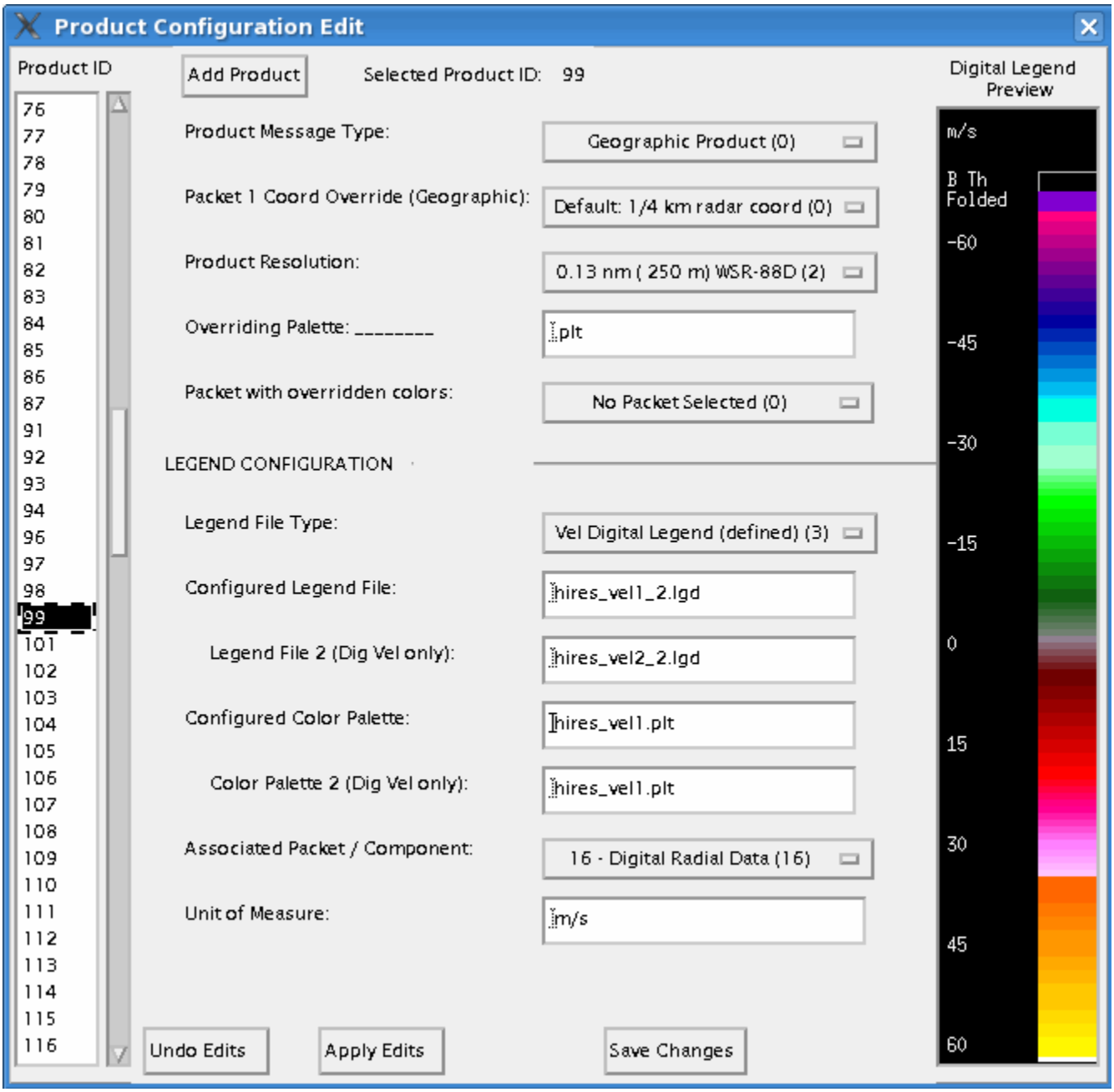
The example above shows the preferences set for Product ID **`123`**, which is a Composite Reflectivity product, 16-level. This is a geographical product (Message Type **0**), having a resolution of 1 KM (Product Resolution **8**). The product uses a raster data packet and the threshold labels are contained within the product so no legend file is used (Legend File Type **0**). No legend file is configured **`.lgd`**. The Associated Packet Type is **`BA07x - Raster Data (54)`**. The Color Palette Preview pane displays the 16 colors in the Configured Color Palette: **`refl_16.plt`**. In this example, the colors are identical to those contained in the Product Specification ICD.

Product Preferences Example 2 - Digital Geographic Product - Method 2



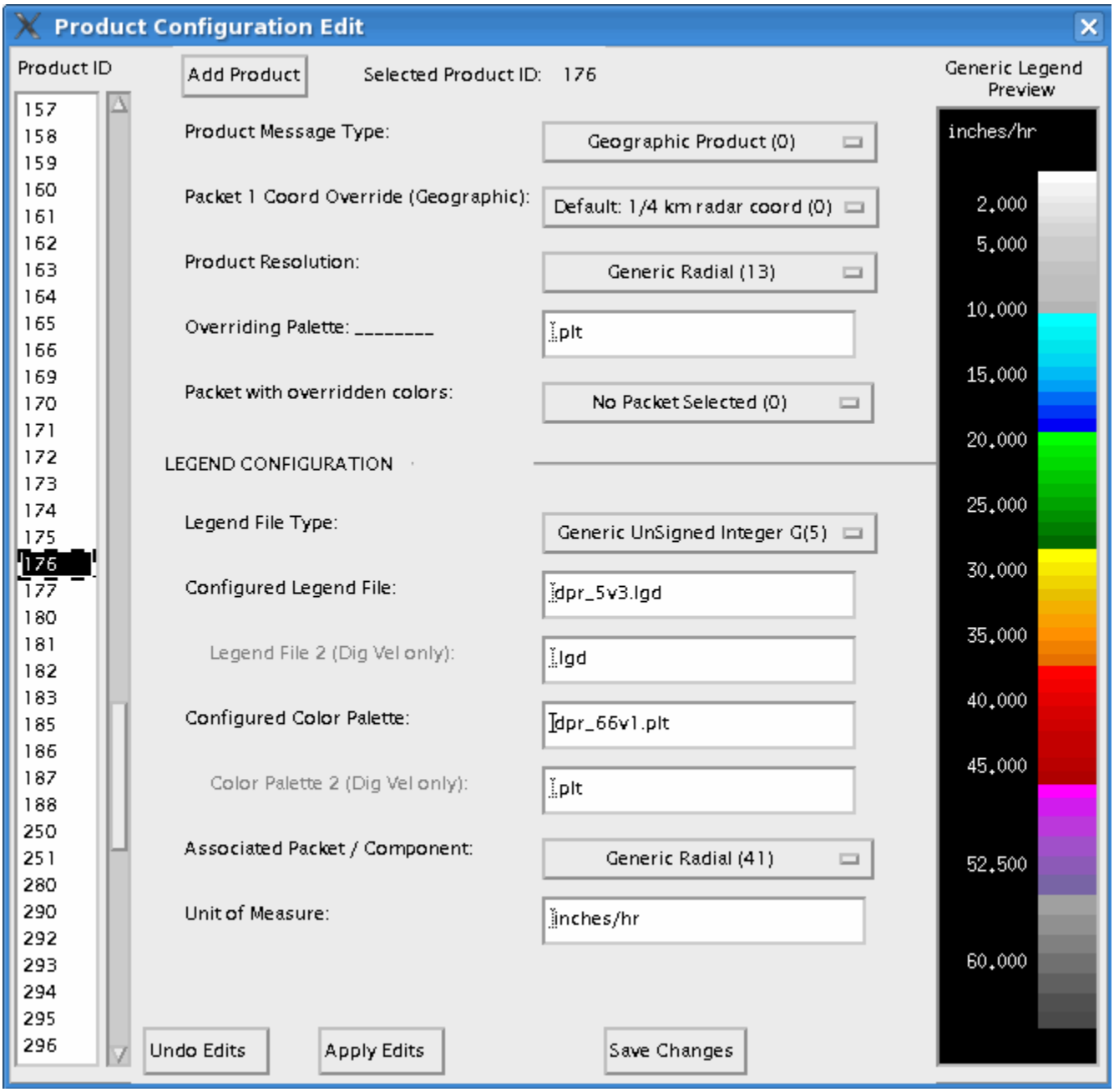
This example (Product ID **134**) is the configuration for a geographical product (Message Type **0**), having a resolution of 1 KM (Product Resolution **8**). The product uses a digital data packet and the threshold labels are stated explicitly (Legend File Type **2**) in the configuration file **`dvil_2.lgd`**. The Associated Packet Type is **`16 - Digital Radial Data (16)`**. The Digital Legend Preview pane displays the product legend resulting from the Configured Legend File: **`dvil_2.lgd`** and the Configured Color Palette: **`dvil_255.plt`**.

Product Preferences Example 3 - Digital Geographic Product - Method 3



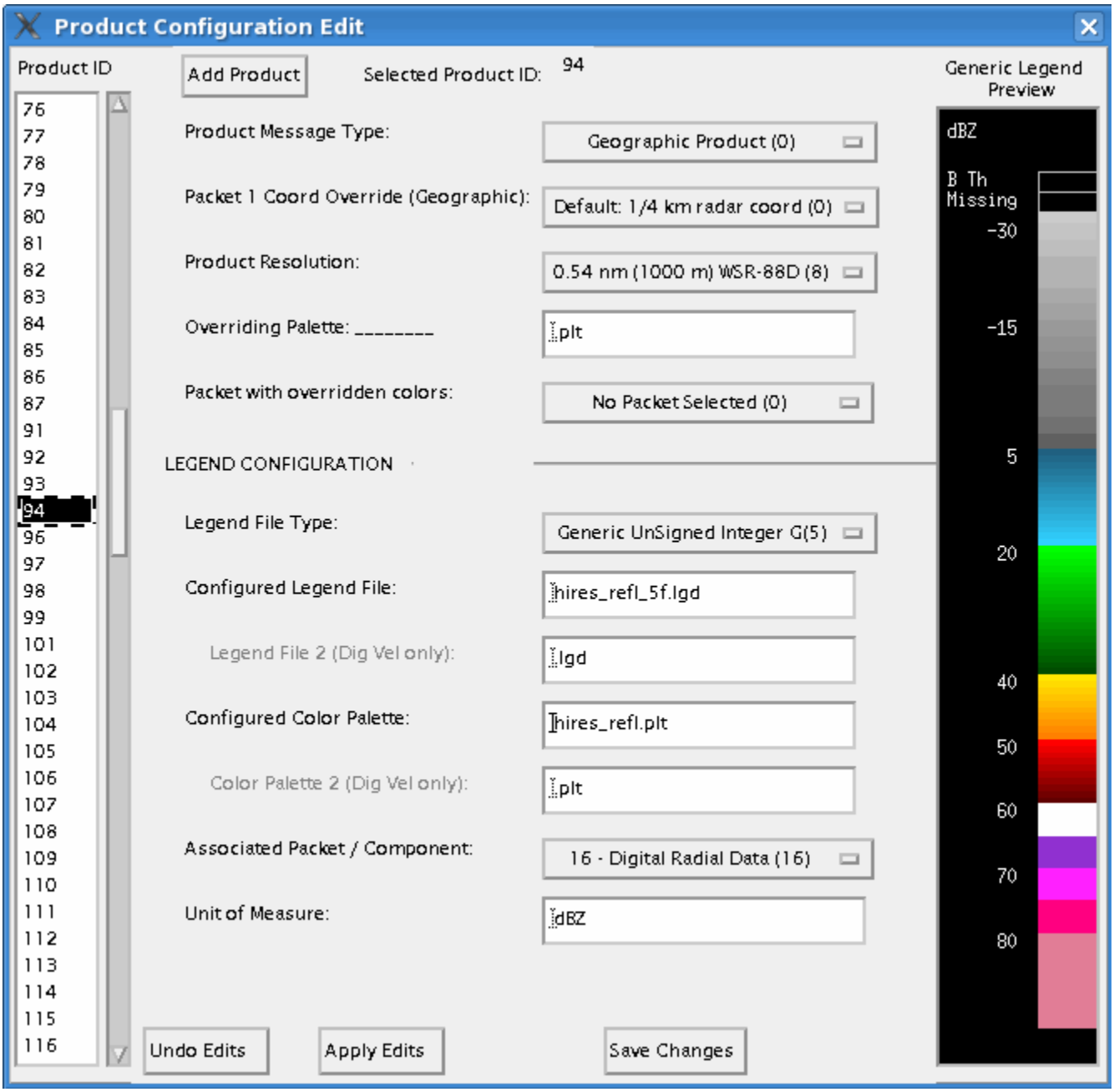
This example (Product ID **99**) is a special configuration for a digital (8-bit) velocity product having a resolution of 0.25 KM (Product Resolution **2**). The product uses a digital data packet and the threshold labels are explicitly defined for a digital velocity product (Legend File Type **3**). Two legend configuration files and two color palette files are configured. The second files are used when the product was produced with the radar in Doppler mode 2. The Associated Packet Type is **16 - Digital Radial Data (16).** Currently the Digital Legend Preview pane can only display the legend produced by the first legend file and first color palette file.

Product Preferences Example 4 - Generic Geographic Product - Method 5



This example (Product ID **`176`**) is the current configuration of the future Dual Polarization product DPR. It is a geographic product using the generic radial component (Product Resolution 13). All generic radial products must use one of the generic legend file types. Currently only unsigned integer arrays are supported for generic radial products (Legend File Type of **`Generic UnSigned Integer G(5)`**).

Product Preferences Example 5 - Generic Geographic Product - Method 5

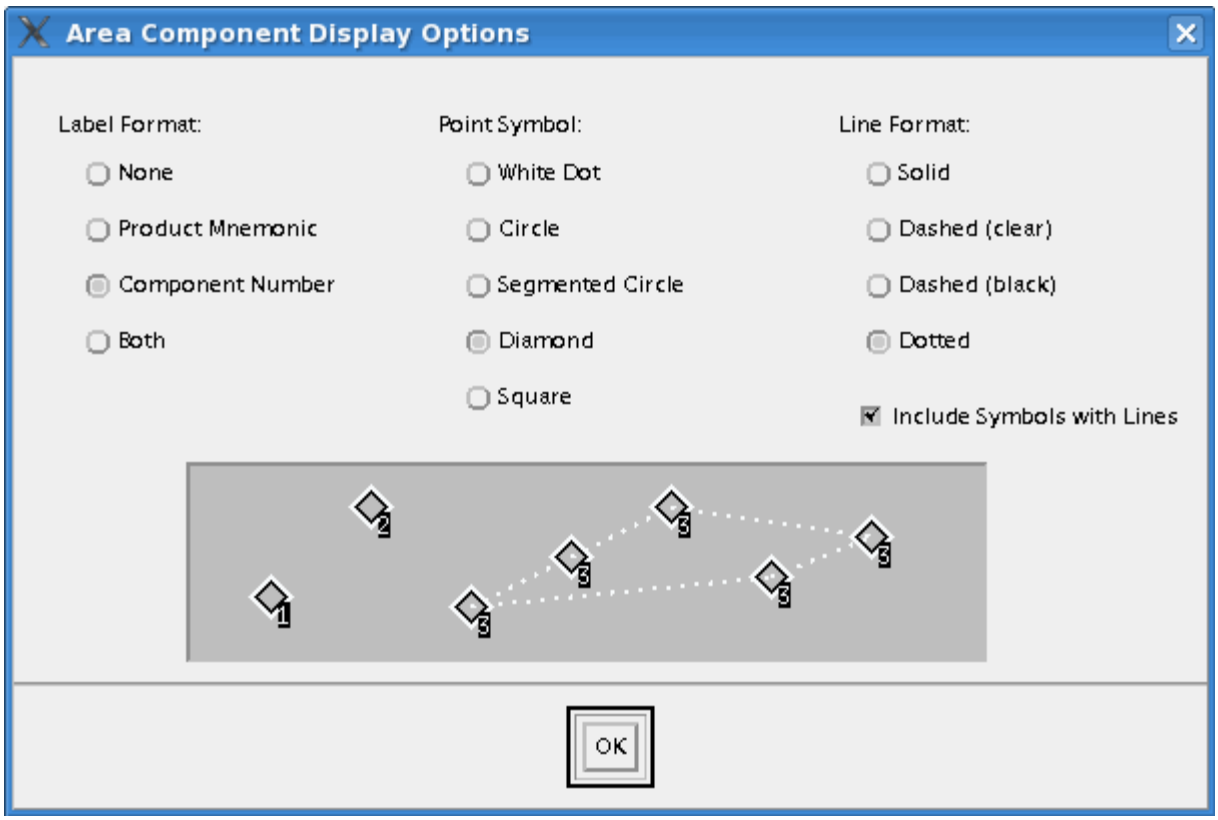


This example (Product ID **`94`**) demonstrates use of the generic legend (Legend File Type of **`Generic UnSigned Integer G(5)`**) for unsigned integers for a digital product with data packet 16.

Generic Product Display Option Menu

Products using the new generic data packet (packet 28) can be relatively self-descriptive if designed appropriately. One aspect of generic products is the isolation of display characteristics from the product content. There currently is no guidance on using generic product component parameters to convey any display hints or to distinguish components from each other for the purpose of using different display attributes. One example, the values of three area component parameters are used by AWIPS to determine which symbol to use in the display of the DMD product in a manner consistent with the display of the traditional MRU product.

CVG provides some support for the display of generic area components. While a mechanism for configuring CVG to display these components in a product specific manner remains to be developed, a capability to manually select some display attributes has been provided for area components.



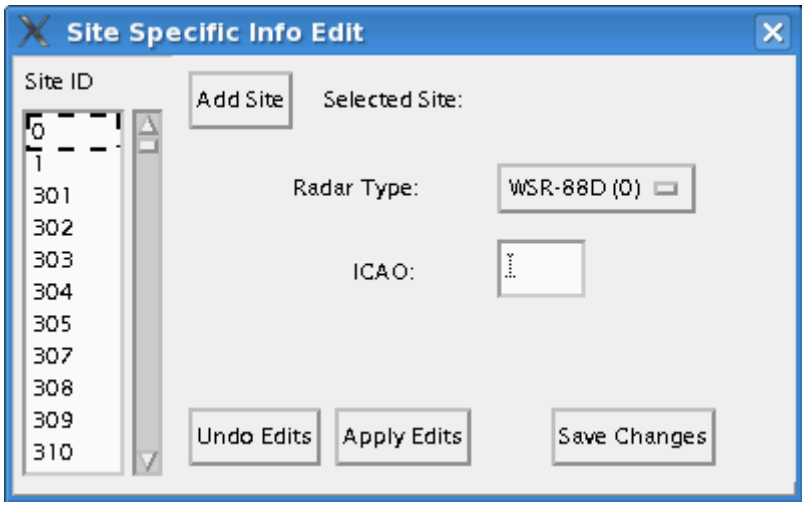
Basic options are provided for the symbol displayed at the plotted location of each point, the label displayed for each point, and the line used to connect points in components containing multiple points. The results of the selection are demonstrated in the preview pane.

Site Preferences Menu

Currently, the only purpose for Site Preferences is to permit the radar type and site ID to be displayed with the product. If the site adaptation data has been updated to reflect the source of the radar data being ingested, the RPG ID is used to display the correct ICAO identifier.

Since all operational WSR-88D sites are already listed in the site data preferences file, there is little need to edit this data.

Clicking the **Edit Site Specific Info...** button on the main preferences dialog brings up the Site Specific Information Edit dialog window.



Part B. CVG Color Palettes

Color Palette - WSR-88D Product Associations

CVG includes sufficient color palettes in order to correctly display all WSR-88D products. Many of the color palette - product associations made in **CVG** are derived from the Product Specification Document. This insures that product colors are rendered in a standard fashion. **CVG** also includes palettes for products that were never intended to be displayed in the original radar system. These "non displayable" products are the digital data array products (having up to 256 data levels) and products using the generic radial component (which can have even more data levels). Appendix B contains a brief reference to many of the palettes currently supplied with **CVG**.

> All existing palettes have their first color defined as "**black**". This color is the background color for the display. For best results, any palette created for displaying WSR-88D products must have "**black**" as the first color. This also means that the meaning of the first data level must be consistent with not being displayed (blending in with the background). The palette file must not have more colors that the number of data levels in the product. These requirements not mandatory if using the generic legend configuration (method 5) for scaled integer product configuration.

CVG color palette files

Several sample "blank" palettes are provided in the color palette directory **~/.cvgN.N/colors** which can be edited manually or which can be modified by the Color Palette Editor.

Each sample "blank" palette contains three rows of zeros with the first row representing red, the second row green, and the third row blue. The number of columns is determined by the number of colors to be specified. There are several sizes of blank palette files containing 16 - 256 colors. The sample 16 color palette file, **blank_16.plt**, looks like:

```
16
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Other sample 16-color palettes are included that are filled with all one color, for example **all_white_16.plt**, **all cyan_16.plt**, etc. These can be modified or used in situations where only one color is desired.

Most useful color palette files contain different colors. The standard 16 color reflectivity palette, **refl_16.plt**, looks like:

```
16
  0   0   0   0   0   0   0 255 231 255 255 214 192 255 153 255
  0 236 160   0 255 200 144 255 192 144   0   0   0   0  85 255
  0 236 245 245   0   0   0   0   0   0   0   0   0 255 201 255
```

Note 1: The number of numerical color entries must be correctly listed on the first line of the file for CVG to correctly use the palette file.

Note 2: Do not modify any files in the **~/.cvgN.N/colors** directory (other than a new development file) as this would change the configuration of an existing product. The modifications should be made to a copy of a file having a different name.

Some of the more useful color palettes for digital products can be seen in Appendix C as displayed by the **CVG** Color Palette Editor.

Editor for CVG Color Palettes

The **Editor for CVG Color Palettes** is a basic X Windows program that aids in defining color palettes. This program (**edit_cvgplt**) replaces a previous program (**cvg_color_edit**), is easier to use, and provides full interactive capability to create new palette files and edit palette files. The instructions for using this editor are provided in Section IV of this document, *Additional CODE Utilities*.

SPECIAL NOTE: Radial and raster run length encoded products (data packets AF1F, BA0F, BA07) must use a palette with the same number of colors as data levels in the product. This palette size (if not using method 5) is also recommended for digital products, data packets 16 and 17. If a palette with fewer colors is used, **CVG** attempts to spread the colors equally over the valid numerical data values. The following part of this document contains additional guidance for palette size for digital products.

If using method 5 the palette size is only restricted to a maximum of 256 colors and must contain the required colors. Method 5 does not require all colors be used.

Part C. Scaled Integer Products - Color Palettes & Threshold Labels

These paragraphs provide an introduction to the configuration of color and legend label display for scaled integer products. A detailed description of configuration of these products is provided in Part D.

Scaled Integer Products

Scaled integer products use unsigned integer type arrays to hold encoded real data values. There are two basic types of scaled integer products.

1. Digital Products - These products use data packet 16 which contains a two-dimensional polar coordinate array of 8-bit integers, permitting up to 256 data values to be stored. These products are called Digital Products because they originally were not intended to be displayed, only used for data processing.

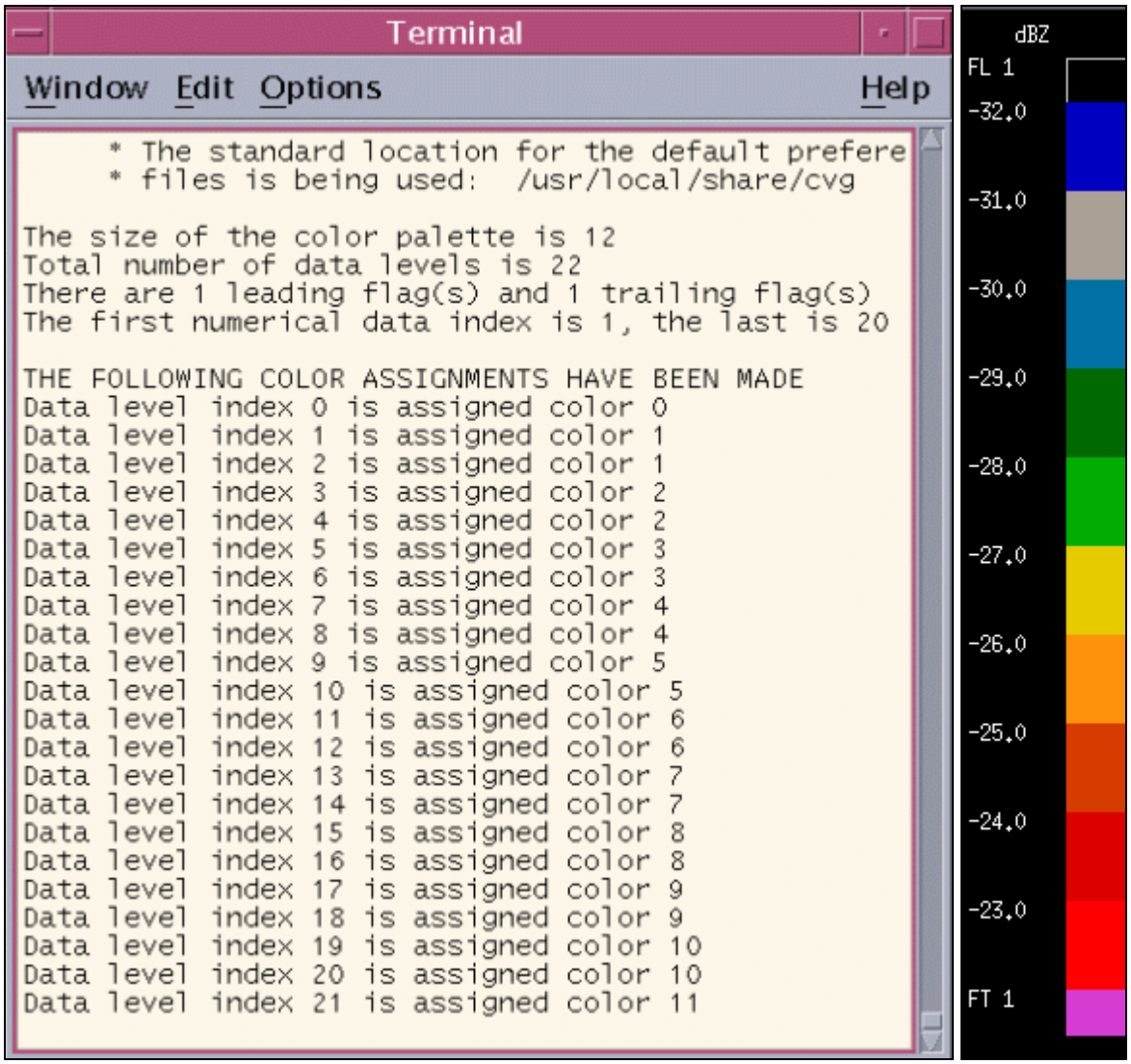
2. Generic Data Array Products - These products use packet 28, the generic data products. The Generic Radial Component can be used for polar coordinate data arrays and the arrays can be one of the following data types: 8-bit / 16-bit / 32 bit signed integers, 8-bit / 16-bit / 32 bit unsigned integers, and the real types float / double.

Threshold Labels in the Product Display Legend

The following figure is a non-operational example of the color bar portion of the CVG product display legend. The data levels increase from the top to the bottom. Often the first 1 or 2 colors are used for a single data value having a special meaning rather than a numerical value. These non-numeric values are call 'flags'. The following convention is followed for flag values.

1. Each flag value is represented by a distinctive color.
2. Flag values are either 'leading' because they occur before the first numerical data level or they are 'trailing' and occur immediately following the highest numerical data level.

The label **FL 1** in this example is the label for a flag value represented with the color black. This flag is a ***leading flag*** because a data level before the first numerical data level is used. Another label **FT 1** is the label for a flag value represented with a purple color. This flag is a ***trailing flag*** because a data level after the last numerical data level is used.



The remaining 10 colors represent the entire range of numerical data levels. The following convention is always followed with the legend for numerical data levels.

1. With numerical data levels, the label is a threshold level. That is, the label states the minimum value that is assigned the color to the right when displayed.

2. Because the label is the lowest value for that color, it is always aligned with the top of the color bar.

For example, the label **-28.0** next to the light green bar represents the lowest data value (-28.0 dBZ) to be assigned the light green color. The diagnostic information displayed in the text window indicates that data levels 9 and 10 are assigned color number 5 in the palette file. Color 0 is black and color 5 is light green.

Digital products (containing data packet 16 and 17 in the product symbology block) were not originally intended for display and therefore do not have legend labels encoded in the *Product Description Block* threshold fields as do the run length encoded products (data packets AF1F, BA0F, and BA07).

Configuration Methods for Digital Products (data packet 16)

The digital products use 8-bit integer arrays in data packet 16 and have up to 256 data levels. The implementation of digital product configuration methods 1 - 3 prevents their use with 16-bit and 32-bit scaled integers available in the generic radial product.

Recommended Legend Configuration Method

Method 5 (generic unsigned integer) is the recommended method for configuring digital products (and the only method for the generic radial component.

Method 5 - Generic Unsigned Integer Products

Support for unsigned integer arrays was added in CVG 8.4 and modified in CVG 8.7. This method is used with the generic 8-bit, 16-bit, and 32-bit unsigned integer arrays in the generic radial component. This method can also be used with digital products using data packet 16.

The generic product configuration method accomplishes everything that Methods 1 and 2 provide and more. This method is more straight-forward than Method 2 and requires about the same level of effort as Method 2. The legend configuration file is larger because in addition to determining flag and threshold labels, a line is included for every color transition (or threshold) assigned. No more than approximately 60 - 70 color transitions are typically needed (128 maximum). However, a smaller color palette file can be used with this method.

This method has several advantages over previous methods.

- The mouse click data query will return the decoded value using the Scale Offset parameters (for those products having a linear encoding that can be described with Scale Offset).
- The legend will respond to changes in the encoding scheme (dynamically changing parameters) if the product includes the necessary Scale Offset parameters in the threshold level fields. See the Class 1 ICD for Build 12 or later.
- This is the most straight forward configuration method eliminating the need for large color pallet files for precise alignment of color bars and legend threshold labels.
- If the encoding cannot be described with the standard Scale Offset formula, the labels can still be explicitly specified in a manner similar to Method 2.

Note: the Scale Offset decoding parameters only work if the real data are encoded in a linear fashion as described in Appendix A.

Other Legend Configuration Methods

Method 1 - Automatically Calculated Legend Labels

The purpose of this method is to provide a means for the quick configuration of a new product for preliminary analysis.

- Flag values are assigned a unique color from the CVG color palette file

- Each flag value has a legend label
- Legend labels for numerical values are spaced evenly over the range of data values and the labels correctly represent threshold values with the data level the lowest assigned the color to its right.
- If using a CVG color palette file with fewer color values that the number of data values (recommended), the colors are spread across the data range.

This method should not be used for the final configuration of a product. The disadvantages of this method are:

- The labels provided for the flag values are not related to their meaning.
- The selection of the numerical threshold levels cannot be adjusted and has no relationship with the meaning of the data.
- This method only works if the real data are encoded in a linear fashion as described in Appendix A.

Method 2 - Explicitly Defined Legend Labels

The purpose of this method is to provide a means of custom configuration of a product.

- The text of all labels (both flag values and numerical values) is explicitly configured.
- The numerical data levels which have threshold labels can be selected with consideration for the nature and meaning of the meteorological data to the user.
- The number, selection and placement of the colors used can be precisely configured
- This method can be used regardless of the manner in which the real data has been encoded into the 8-bit integer values.

The only disadvantage of this method is that it takes more effort than method 1. To easily match designated threshold labels with the beginning with the colors, a large palette file, one containing the number of colors equal to the total number of data levels in the product must be used.

Method 3 - Explicitly Defined Labels for Velocity Products

This method is a simple extension of Method 2 that is used only for digital velocity products that have two encoding values, one for each of the RDA velocity modes. A different legend configuration file and color palette file is configured for each mode.

Configuration Methods for Generic Radial Components

Generic radial components are recently being used for the first time. These components have three kinds of data arrays: signed integer arrays, unsigned integer arrays, and real data arrays. CVG will have three almost identical methods for the configuration of threshold labels and assignment of display colors. **The original methods 1, 2, and 3 cannot be used with the generic radial component.**

The generic product configuration methods accomplish everything that Methods 1 and 2 provide and more.

- The text of all labels (both flag values and numerical values) can be explicitly configured.
- The labels of numerical data levels can also be calculated using the Scale Offset parameters. The Scale Offset parameters can be provided in two ways.
    - In the legend configuration file.
    - In the threshold level fields of the product itself. Proper use of these fields is described in the RPG Class 1 User ICD for Build 12 or later. This permits the legend display to respond to dynamically changing parameters.
- If configured for decoding, the mouse click data query will return the decoded value using the Scale Offset parameters (for those products having a linear encoding that can be described with Scale Offset).
- The numerical data levels which have threshold labels can be selected with consideration for the nature and meaning of the meteorological data to the user.
- The number, selection and placement of the colors used can be precisely configured.
- This method can be used regardless of the manner in which the real data has been encoded into the 8-bit integer values by explicitly specifying the labels for the numerical value thresholds (color assignment lines).

The Scale Offset parameters can be used to support the display of the decoded data value with the mouse click data query function. If the Scale Offset parameters are properly contained in the product, CVG reflect the dynamically changing decoding parameters that some products use.

About the same level of effort is required as for Method 2 used with digital products. The legend configuration file is larger because in addition to determining flag and threshold labels, a line is included for every color transition (or threshold) assigned. No more than approximately 60 - 70 color transitions are typically needed (128 maximum). However, a smaller color palette file can be used with this method.

Method 4 - Generic Signed Integer Products

Support for signed integer arrays has not yet been implemented.

Method 5 - Generic Unsigned Integer Products

Support for unsigned integer arrays was added in CVG 8.4 and modified in CVG 8.7. This method can be used with the generic 8-bit, 16-bit, and 32-bit unsigned integer arrays used in the generic radial component. This method can also be used with digital products using data packet 16.

Method 6 - Generic Real Data Products

Support for real data arrays has not yet been implemented.

Examples of digital legend display

1. Examples of generic product legend labels:

Method 5 - Product 176 DPR

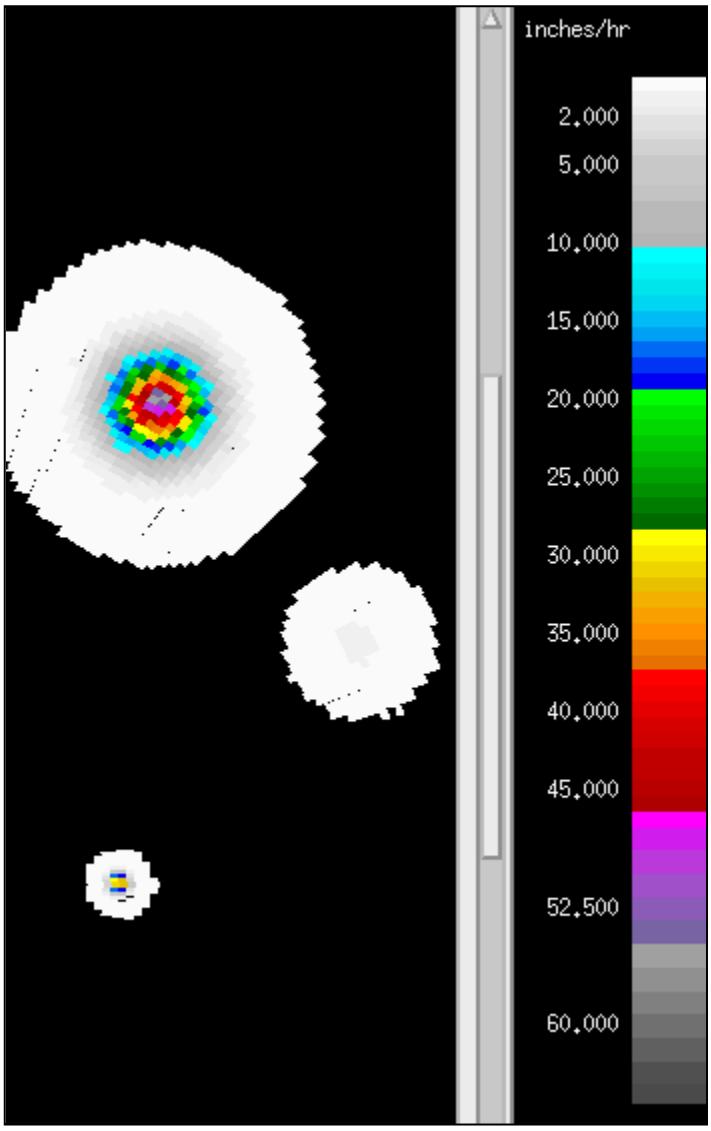
The legend configuration file used in this example is **`dpr_5v3.lgd`**.

The color palette file used with the legend file is **`dpr_66v1.plt`**.

In this example the numerical threshold labels are calculated from the data levels using the Scale Offset parameters in the legend file. There is also an option to decode data levels into numerical threshold labels using the Scale Offset parameters within the product.

Note that the numerical labels are right justified and aligned. Three decimal places are displayed as configured in the legend file.

With the generic methods, the size of the color palette file is not critical as with the previous configuration methods. The palette must contain the number of colors required for the display of the product but not all colors need to be used. This is because the legend file explicitly assigns a data level representing the minimum data level to be assigned to each color. This can be called a threshold or transition data level.



This is a preliminary display of a development product. The look of the final version of this product is subject to change.

Method 5 - Product 94 DR

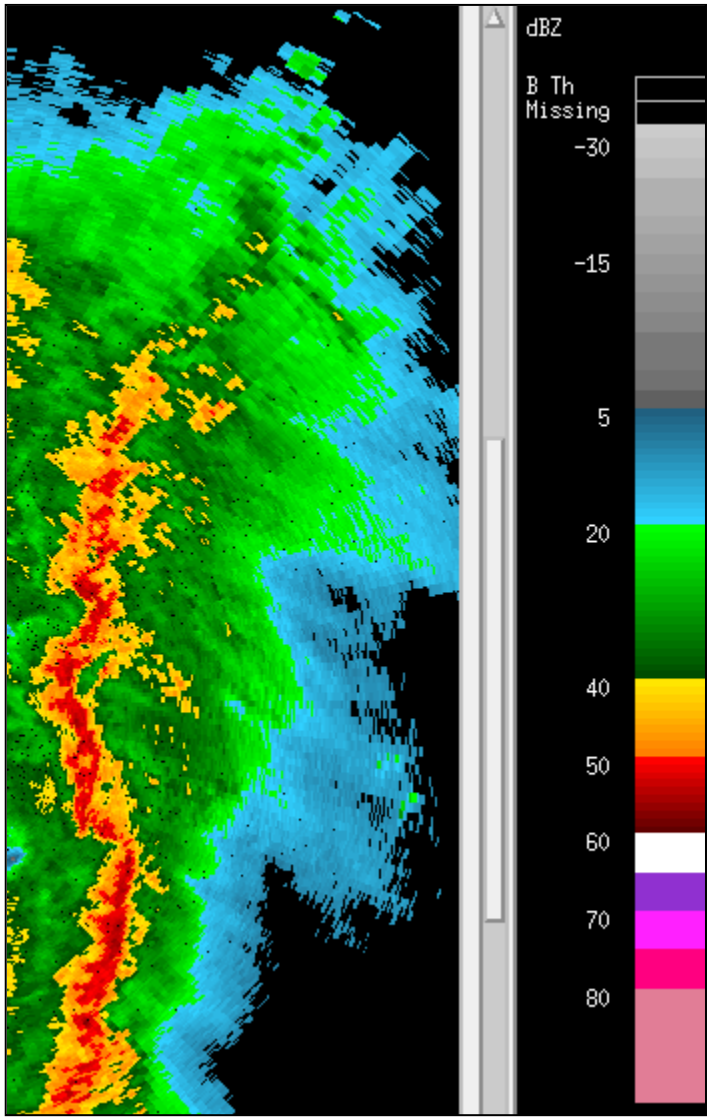
The legend configuration file used in this example is **`hires_refl_5f.lgd`**.

The color palette file used with the legend file is **`hires_refl.plt`**.

In this example the labels for the flag values are explicitly defined in the legend configuration file. The numerical threshold labels are calculated from the data levels using the Scale Offset parameters in the legend file.

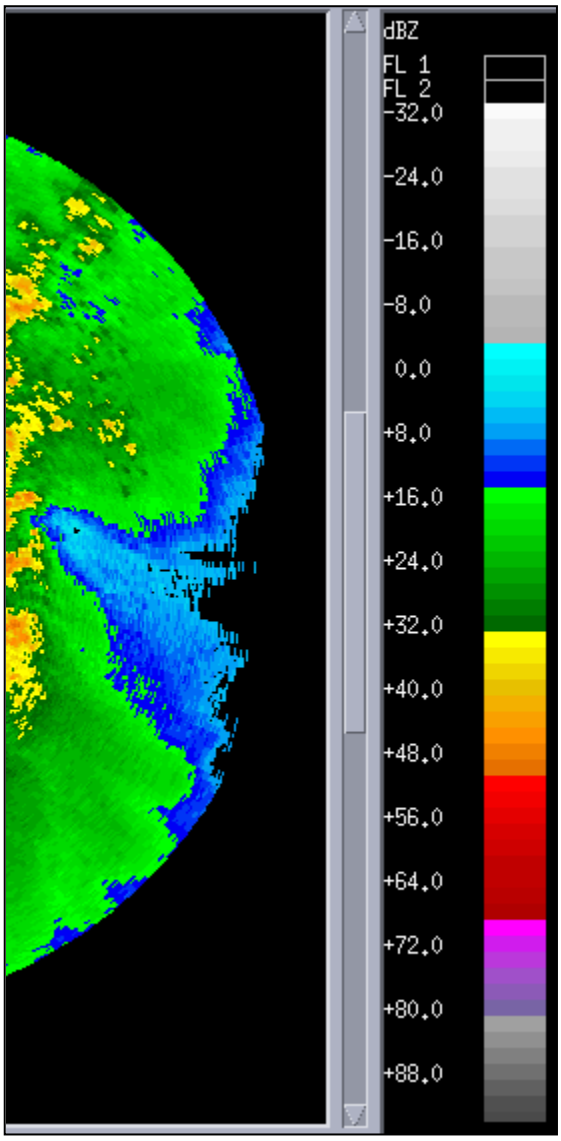
Note that the numerical labels are right justified and aligned. No digits to the right of the decimal are displayed as configured in the legend file.

With the generic methods, the size of the color palette file is not critical as with the previous configuration methods. The palette must contain the number of colors required for the display of the product but not all colors need to be used. This is because the legend file explicitly assigns a data level representing the minimum data level to be assigned to each color. This can be called a threshold or transition data level.



2. Examples of calculated and explicitly defined legend labels:

Method 1 - Product 1991 DIGREFL



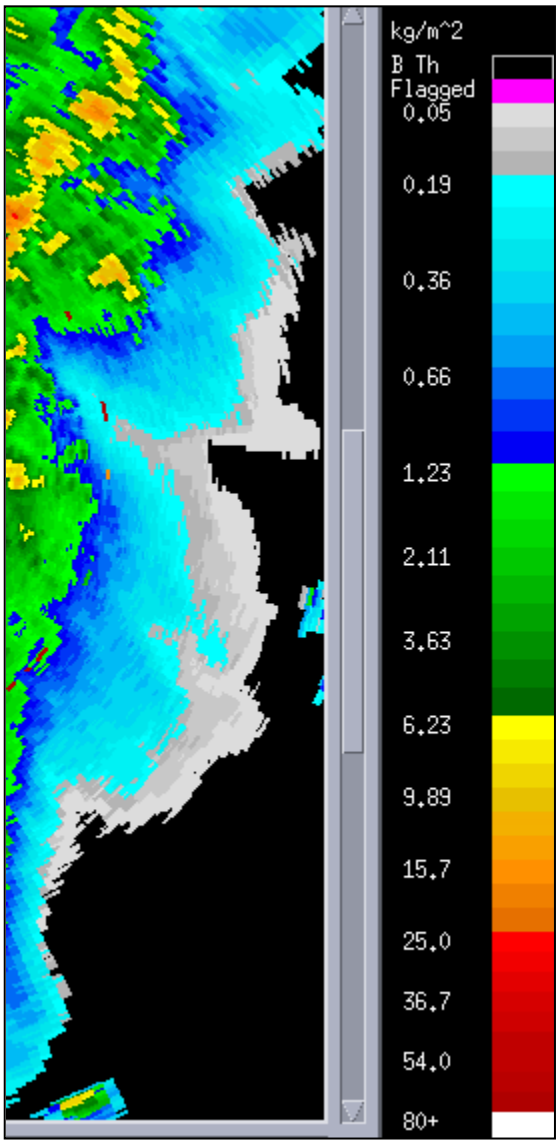
This example is a product with 256 data values including two leading flags.

A CVG color palette file with 66 color values was used (**`drefl_66.plt`**). With Method 1, quick and good results are obtained with a palette file having a color value for every 4 data values (64 in this case) plus a color for each flag value.

The two flag values (labeled **FL1** and **FL 2**) are assigned the first two color values in the palette file (black). The remaining 62 colors in the palette file are spaced over the remaining 254 numerical values.

The automatically calculated numerical threshold labels are calculated from the parameters in the legend file (**`refl_1.lgd`**) and are aligned with the color bars. Note that even though the numerical labels are aligned with a bar, the major changes of color may not be aligned with a threshold label and may not be suitable for the final product configuration.

Method 2 - Product 134 HRVIL



This example is a product with 255 data values including two leading flags.

A CVG color palette file with 255 color values was used (**`dvil_255.plt`**). With Method 2, using a palette file having the same number of color values as the total number of data values is recommended. This provides the best control of matching major color changes with numerical labels that have been chosen for their meteorological significance.

The two flag values have been given meaningful labels (**B Th** and **Flagged**).

The data levels for the numerical threshold labels are chosen first. The colors are assigned by editing the CVG color palette file (**`dvil_2.lgd`**). The first two colors (black and light purple) are matched with the two leading flags. Actually, with the CVG palette size equal to the number of data levels, each data level is assigned the corresponding color value in the palette file.

The remaining 253 color values in the palette file are edited to create an effective palette having 38 colors spread over the range of numerical values. This is accomplished by having identical colors defined for adjacent color values in the file. The number of color bars created with identical adjacent values is not critical but it does determine the visual quality of the image.

Part D. Scaled Integer Products - Configuration for Display

Introduction

Before attempting to create legend configuration files and CVG color palette files, you must be have a basic understanding of CVG color palettes and threshold labels covered in the previous paragraphs (See Section C. of this document). **The tool `edit_cvgplt` is provided to create and edit color palette files. Currently there is no tool to assist in creating the legend configuration file.**

Originally the information contained in *Product Description Block* threshold fields in digital products (and in generic radial products) was not sufficient to display these products. **`CVG` will not use this information but uses configuration parameters in legend configuration files (methods 1, 2, and 3) to decode data values for calculation of threshold labels.** There is one exception: For Method 3, threshold field 1 in digital velocity products is used to determine the velocity mode of the radar when the product was produced.

Using configuration Method 5, if the product includes the parameters supporting the Scale Offset decoding formula, `CVG` can read the decoding parameters from the threshold fields to decode the data levels for display. See the Build 12 (or later) RPG Class 1 User ICD.

Recommended Legend Configuration Method

Method 5 (generic unsigned integer) is required for products using the generic radial and is the recommended method for configuring digital products.

Configuration of Generic Radial Products and Digital Products using Method 5

Support for unsigned integer arrays was added in CVG 8.4 and modified in CVG 8.7. This method is used with the generic 8-bit, 16-bit, and 32-bit unsigned integer arrays in the generic radial component. This method can also be used with digital products using data packet 16.

The generic product configuration method accomplishes everything that Methods 1 and 2 provide and more. This method is more straight-forward than Method 2 and requires about the same level of effort as Method 2. The legend configuration file is larger because in addition to determining flag and threshold labels, a line is included for every color transition (or threshold) assigned. No more than approximately 60 - 70 color transitions are typically needed (128 maximum). However, a smaller color palette file can be used with this method.

Other advantages over Method 2 include the ability to use the new Scale Offset parameters in the product itself or as specified in the legend file and if the product is compatible with Scale Offset, the mouse click data query can return the decoded value.

CVG includes two method 5 sample legend files that can be used as a template: **`method_5_sample_no_flags.lgd`** and **`method_5_sample_w_flags.lgd`**. These files have significantly different formats in CVG 8.6 and 8.7. It is highly recommended that the update to CVG 8.7 be installed.
.

The format of the generic legend file will support all generic legend configuration types.

In-Line Comments

Complete lines can be commented out anywhere in the legend file by placing the character '**#**' in the first column of a line. Partial line comments are permitted only in the 10 lines defining the configuration parameters. The parameter must be the first token in the line, the remainder of the line can be a comment.

10 Configuration Parameter Lines

The first 10 non-comment lines in the legend file define the basic parameters for configuration. They must be entered in the correct order. The parameter must be the first token in the line.

1. Generic Legend File Data Type. The value **`4`** represents signed integer data type (not yet supported), the value **`5`** represents an unsigned integer type, and the value **`6`** represents a real data type (not yet supported). This value must correspond to the Legend File Type configured in the Product Configuration dialog and is one of several tests CVG uses to detect errors / inconsistencies in generic legend configuration.
2. Number of Color Assignments. The number of data lines in the configuration file. Each line assigns a color to a data level and optionally assigns a threshold level label to the data level. For color assignment, this is the lowest level (threshold level) to be assigned the color specified. No more than 128 colors can be assigned.
3. Minimum Data Level. The minimum data level (not decoded value) used by the product. This includes both flag and numerical values. The format of the value entered corresponds to the data type specified in parameter 1. By current convention unsigned integer products always have a minimum of **`0`**.
4. Maximum Data Level. The maximum data level (not decoded value) used by the product. This includes both flag and numerical values. The format of the value entered corresponds to the data type specified in parameter 1. If the source (parameter 8) is **`$product`**, this value serves as a default value for use in the product configuration legend preview pane.
5. Number of Leading Flags. Starting with the minimum data level, the number of data levels that have special meaning (flag value) rather than a numerical value. If the source (parameter 8) is **`$product`**, this value serves as a default value for the product configuration legend preview pane.
6. Number of Trailing Flags. Ending with the maximum data level, the number of data levels that have special meaning (flag value) rather than a numerical value. If the source (parameter 8) is **`$product`**, this value serves as a default value for the product configuration legend preview pane.
7. Numerical Label Format. If the legend label is not explicitly defined via a text string, then the value of this parameter represents the number of digits displayed to the right of the decimal point. The value of this parameter should be related to the precision of the actual data. The minimum value for this parameter is **`0`**, the maximum is **`6`**. The value of '**`0`**' forces a real value to look like an integer value The maximum of 6 decimal places is based upon the maximum length of the label of 8 characters including the dot (**`n.nnnnnn`**).

8. Source of Decode Parameters. This determines the source of the parameters to be used for the Scale Offset decoding. Scale Offset decoding is used by the CVG mouse click data query feature and can be used to calculate the legend threshold labels for numerical values. The parameters affected are: maximum data level (param 4), number of leading flags (param 5), number of trailing flags (param 6) along with the Scale and Offset parameters (param 9 and 10).
   ---- The value **`$no_decode`** specifies no decoding is to be accomplished. **`$no_decode`** must be entered if the data values cannot be described via the Scale Offset method; otherwise the mouse click data query feature will display an incorrect decoded value.
   ---- The value **`$product`** will result in CVG using the Scale Offset parameters contained in the product itself. These parameters must be in the product threshold level fields as specified in the Build 12 (or later) RPG Class 1 User ICD.
   ---- The value **`$file`** will result in CVG using the values entered in this configuration file. The following parameters should be entered correctly even if **`$product`** is used.
9. Scale Parameter. The value of the Scale parameter used in decoding. This is a real number. Note that `0.0` is an invalid value for Scale. **See the paragraph titled Scale Offset Limits for CVG Legend below for additional guidance for this parameter.** If the source is **`$product`**, the value serves as a default value for the product configuration legend preview pane.
10. Offset Parameter. The value of the Offset parameter used in decoding. This is a real number. **See the paragraph titled Scale Offset Limits for CVG Legend below for additional guidance for this parameter.** If the source is **`$product`**, the value serves as a default value for the product configuration legend preview pane.

Decoding (if accomplished) uses the values for Scale, Offset, maximum data level, number of leading flags, and number of trailing flags. If these parameters are taken from the product, the entries in this file serve as default values to be used in the legend preview pane of the CVG Product Configuration dialog.

Data Level Color Assignment Lines

The remaining uncommented lines are the data level color assignment lines. The number of lines must equal the value of parameter 2. Each line explicitly assigns a color number (that is the number of the color in the CVG color palette file) to a data level. Some of these lines also assign a text label to a data level. No more than 32 labels can be specified. For color assignment, this is the lowest level (threshold level) to be assigned the color specified. This is the data level where the color used transitions to a new color. This color is used for higher levels until a different color is assigned or until the maximum data level in the product. No more than 128 colors can be assigned.

The first token in each line is the raw data level (for method 5 this is an unsigned integer type). The second token is the number of the color in the CVG palette file configured for the product. The characters on the remainder of the line (if not all blank spaces) specify the legend text label to be displayed. This label is either the title of a flag value or the decode value of a numerical data level.

The legend text label can be specified in three ways. If **`$scale_offset`** is entered, the result of the Scale Offset formula is used to decode the data level and displayed with the format specified by parameter 7. If **`$data_level`** is entered, the raw data level is displayed with the format specified by parameter 7. Otherwise the entry is treated as plain text and displayed directly (e.g. `2.000`). The labels can contain no more than 8 characters (including the decimal point) regardless of whether they are derived from simple text or decoded values using **`$scale_offset`**.

It should be noted that the color numbers assigned do not have to be in sequence. They may be in sequence if the design of the color palette file supports using the colors in order. **See the paragraph titled Color palettes used with Method 5 below for a full description of color palettes used.**

WARNING: A data level can appear only once in the palette file. The result of having more than one color assignment entry for a data level is unpredictable.

Flag data levels are adjacent data levels each having a color assigned in the legend file. Because of the limited space in which to draw the legend, there are limits to how close the listed numerical data levels can be for color assignments and for label assignments. The limits are based upon the range of data levels configured. The first example using **`method_5_sample_w_flags.lgd`**, **`range`** is 64501. The digital product (packet 16) example using **`hires_refl_5f.lgd`**, **`range`** is 256.

- The suggested minimum spacing between color assignments for numerical data levels is **`range/128`** which is approximately 500 in the first example and 2 in the digital product example. The color bar will disappear altogether if a spacing less than approximately **`range`** / 500 is used. No more than 128 colors can be assigned.
- The suggested minimum spacing between numerical data level legend labels is **`range/32`** which is approximately 2000 in the first example and 8 in the digital product example. The labels will overlap if too small a spacing is used. No more than 32 labels can be specified.

Example 1 - A generic radial product having 64,501 data levels (max level of 64,500)

The following is taken from **`method_5_sample_w_flags.lgd`** legend configuration file (version 2).

Type **`5`** supports any unsigned integer data type.

This file assigns colors to **`66`** data levels.

The minimum data level is **`0`** and the maximum data level is **`64500`**. By convention, the minimum is always **`0`** for scaled integer products. These include all flag and numerical values.

This product has **`1`** leading flag and **`1`** trailing flag.

If not simple text, the decoded threshold label is displayed with **`3`** decimal places.

```
5                                  type = unsigned integer
66                                 number of colors used
0                                  minimum data level used
64500                              maximum data level used
1                                  number of leading flags
1                                  number of trailing flags
3                                  digits after decimal pt
$file                              source of parameters
1000.0                             SCALE
0.0                                OFFSET
0000    0  NOTHING                 data_level  color  label
0007    1                          data_level  color
0500    2                          data_level  color
1000    3                          data_level  color
1500    4                          data_level  color
2000    5 $scale_offset            data_level  color  label
  "     "                            "         "        "
  "     "                            "         "        "
58500  61                          data_level  color
60000  62 $scale_offset            data_level  color  label
61500  63                          data_level  color
63000  64                          data_level  color
64500  65  MAX_FLAG                data_level  color  label
```

The decoding parameters are taken from this file (**`$file`**) rather than the product threshold levels (**`$product`**).

The next two lines are the values for Scale (`1000.0`) and Offset (`0.0`).

The flag value titles are specified with simple text and the numerical data level labels are decoded using Scale Offset.

Example 2 - A digital product having 256 data levels (max level of 255).

The following is taken from **`hires_refl_5f.lgd`** legend configuration file (version 1). This file is used to configure the digital reflectivity product (ID 94) and others.

Type `5` supports any unsigned integer data type.

This file assigns colors to `77` data levels.

The minimum data level is `0` and the maximum data level is `255`. By convention, the minimum is always `0` for scaled integer products. These include all flag and numerical values.

This product has `2` leading flags and `0` trailing flag.

If not simple text, the decoded threshold label is displayed with `0` decimal places.

```
5                          type = unsigned integer
77                         number of colors used
0                          minimum data level used
255                        maximum data level used
2                          number of leading flags
0                          number of trailing flags
0                          digits after decimal pt
$file                      source of parameters
2.0                        SCALE
66.0                       OFFSET
000   0  B Th              data_level  color  label
001   1  Missing           data_level  color  label
002   2                    data_level  color
006   6 $scale_offset      data_level  color  label
011  11                    data_level  color
016  16                    data_level  color
 "    "                       "         "        "
 "    "                       "         "        "
186 186 $scale_offset      data_level  color  label
196 196                    data_level  color
206 206 $scale_offset      data_level  color  label
216 216                    data_level  color
226 226 $scale_offset      data_level  color  label
```

The decoding parameters are taken from this file (**`$file`**) rather than the product threshold levels (**`$product`**).

The next two lines are the values for Scale (`2.0`) and Offset (`66.0`).

The flag value titles are specified with simple text and the numerical data level labels are decoded using Scale Offset.

Scale Offset Limits for Method 5 CVG Legend

The CVG legend panel permits a maximum of 8 characters for the legend labels for both flag values and numerical values. If a longer label is specified it is truncated. Using method 5, there is an option to calculate the numerical labels using the Scale and Offset parameters (other parameters used are the maximum raw data level and the number of leading and trailing flag values).

The limit of 8 characters in the legend label results in a maximum and minimum value that can be represented in the legend. For example, with 0 decimal places specified, the maximum positive value that can be represented is `99999999`; with 2 decimal places specified the maximum is `99999.99`. The minimum negative value that can be represented is one less digit because of the addition of the minus sign (`-9999999` and `-9999.99`). With 6 decimal places specified the minimum positive value that can be represented is `0.000001`; with 2 decimal places the minimum is `0.000001`. The number of decimal places is applied to all of the calculated numerical labels in the legend. This is more than sufficient for all existing products. The prudent choice of unit of measure normally permits the range and precision of any meteorological parameter to be adequately represented within these limits.

These limits on legend display translate into limitations on the values of Scale and Offset that can be used to encode and decode real data values into unsigned integers. A couple of examples provide an idea of what the limits can be. For the minimum value of `0.000001`, if the Offset were `0.0`, the maximum value of Scale would be `1000000.0`. With a 256 level product, for the maximum value of `99999999` the minimum value of Scale would be `0.00000255`. With a product using the unsigned short having 65536 data levels, for the maximum value of `99999999` the minimum value of Scale would be `0.000655`. These limits change with the value of Offset and the number of decimal places configured for display.

Because the encoded values must increase as the raw data levels increase, the value of Scale must be a non-zero positive number.

During display of a legend in the preview pane, CVG provides an indication of what the product's minimum and maximum value would look like if displayed to help catch situation where the range and precision of encoded values exceed the CVG legend display capabilities.

Color palettes used with Method 5

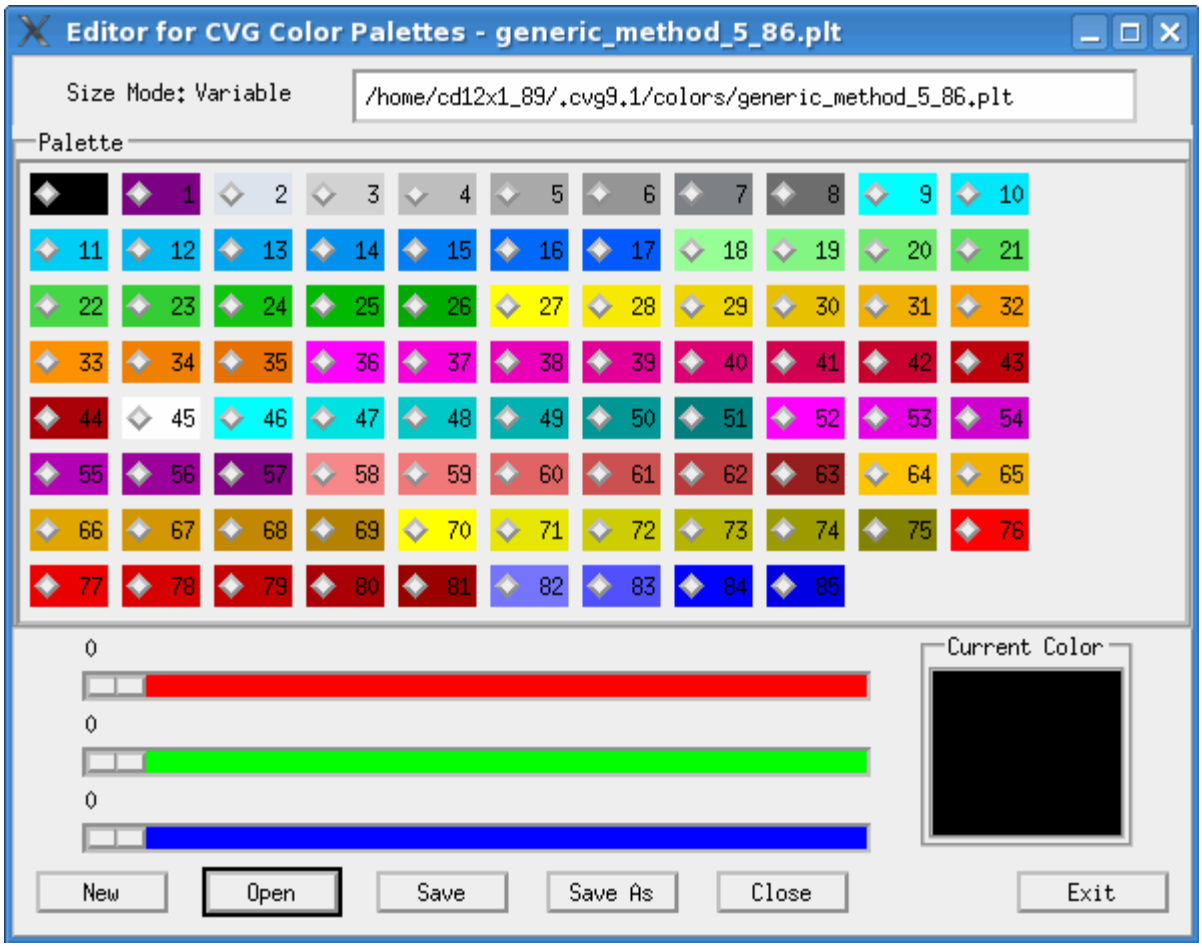
All CVG color palette files are limited to 256 defined colors. With Method 5 there is no additional restriction on the number of colors in the palette file as long as a sufficient number of colors are defined to provide a meaningful display. **Not all colors in the palette file need to be used. The significant restriction is on the Method 5 legend file which can make no more than 128 color assignments.**

The following guidance is provided for the Method 5 color palette file.

- The file must include black (background) as color 0.
- It is recommended that each color in the file be unique (reduces confusion).

As noted previously, the legend file can assign color numbers in any order. This permits a master color palette file containing up to 256 colors to be created. This master palette can be used with any Method 5 legend file which selects the desired colors for specified data levels.

CVG includes a sample master color palette file for method 5: **`generic_method_5_86.plt`**. It contains 86 colors in several sequences of related colors.



The colors have been defined so that the difference between adjacent colors in a sequence can be perceived on a good color display.

HINT for Reconfiguring Existing Digital Products

With existing digital products configured with method 2, the color palette file assigns a color directly to each data level. However, these palette files typically contain less than 128 unique colors. If reconfiguring an existing digital product with a 256 color palette file, the original palette file can be used. The color assignment lines (transition or threshold data levels) in the method 5 legend file can simply reflect the data levels in the original color palette file where the defined color changes.

Other Legend Configuration Methods for Digital Products

Configuration of Digital Products using Method 1

Method 1 can be used to quickly configure a product for display. Understanding how to state the required parameters requires some explanation. This example is taken from the **`refl_1.lgd`** legend configuration file.

256	total data levels
2	number of leading flags
0	number of trailing flags
-320	minimum value
5	increment between values
10	scale factor of min value
10	scale factor of increment

This product has **`256`** data levels. The next two lines indicate the number of leading flag values and trailing flag values (**`2`** and **`0`** in this example).

The first three lines are used by CVG to assign colors in the CVG palette file to data levels. All seven lines of the configuration file are used to calculate the value displayed and the position of the threshold labels.

The 4 parameters following the first three lines are integers that represent two real numbers: the minimum numerical value and the increment between adjacent real values.

- The **minimum value** is described by the 4th and 6th parameters. The integer in parameter 4 representing the minimum value, **`-320`**, is scaled by a factor of ***`10`** units. Therefore the minimum value is **`-32.0`** units.
- The **increment** is described by the 5th and 7th parameters. The integer in parameter 5 representing the increment, **`5`**, is also scaled by a factor of ***`10`** units. The increment is therefore **`0.5`** units.

This is admittedly awkward. However, *the minimum value and increment are stated in this fashion because this corresponds to how they are partially encoded in the threshold levels in most existing digital products*

Since two leading flag values is specified, data level 0 is the first flag value which is labeled "**`FL 1`**" and the data level 1 is the second flag value which is labeled "**`FL 2`**".

After the flag values, data level 2 represents the minimum value **`-32.0`** units. Because the increment is **`0.5`**, data level 3 represents **`-31.5`** units, etc. **`CVG`** automatically calculates the numerical labels and places them at the beginning of a color as long as the palette is chosen correctly.

Color palettes used with Method 1

`CVG` can handle palette sizes from 1-256 with the following guidance. The palette size (number of colors in the **`CVG`** palette) should never be greater than the total number of data levels (including flag values).

The easiest way ensure the calculated threshold labels are aligned with the beginning of a new color is to choose a CVG color palette file where the number of color values in the file (the palette size) is less than the total number of data levels. In addition:

- The first color must be black (background).
- Each color specified in the CVG palette file must have a unique color value (adjacent colors not identical).

If it is desired that all colors in the palette file be used, the palette is sized based upon the number of flag values and the total number of data values. That is, the number of colors specified in the CVG color palette file should be equal to:

palette_size = ceil **(** (total_num_levels - num_flags) / N **)** + num_flags

For existing products, N has been chosen to be 4. This formula can be approximated by stating that there should be a color specified for every 4 numerical data levels plus 1 color for each flag value. Simply stated, if all 256 data levels are used, a palette with 66 color values is used for products having 2 flag values, 65 color values if there is 1 flag value, and 64 colors with no flag values.

A new palette file can be created or one of the following palette files provided with CVG can be used. **`drefl_66.plt`**, **`dvil_66.plt`**, **`nws_66.plt`**, **`nws_64.plt`**, and **`refl_64.plt`** can be used for reflectivity / precipitation related products. **`vel_66.plt`**, **`vel_65.plt`**, and **`vel_64.plt`** can be used for velocity products.

Configuration of Digital Products using Method 2 & 3

Method 2 (and Method 3) can be used for precise configuration of product display. This method is more straight forward but it requires more effort. This example is taken from the **`refl_2.lgd`** legend configuration file.

This product has **`256`** data levels. The next two lines indicate the number of leading flag values and trailing flag values (**`2`** and **`0`** in this example).

As with Method 1, the first three lines are used by CVG to assign color in the color palette file to data levels. The remaining lines in the legend configuration file are used to define and place the threshold labels.

```
256          total data levels
2            number of leading flags
0            number of trailing flags
0 B Th       data_level   text label
1 Missing    data_level   text label
2  -32       data_level   text label
62  -2       data_level   text label
98  16       data_level   text label
134  35.5    data_level   text label
170  52      data_level   text label
206  70      data_level   text label
230  82      data_level   text label
```

After the first three lines, each line contains a pair of values. The first token is an integer representing the raw data level in the byte (0-255) that will be labeled. The rest of the line is a text string that will appear next to the color corresponding to that data level.

In this example, data level 0 (the first leading flag) is labeled "**`B Th`**" for below threshold. Data level 1 (the second flag) is labeled "**`Missing`**". The first space after the integer data level is ignored. Additional spaces, if present, are part of the label string.

The flag values are adjacent data levels. The numerical data level threshold labels will overlap if configured with insufficient spacing.

Currently, there is room for only 8 characters to be displayed next to the color bar. The remaining labels correspond to the decoded numerical data values which must be calculated manually.

The advantage of this method is control. Flags may be labeled meaningfully and numerical labels can be created at any point desired. The disadvantage of this method is the labels will not automatically be aligned with the beginning of a color on the color bar. To accomplish this, a palette having the same size as the total number of data levels normally should be used.

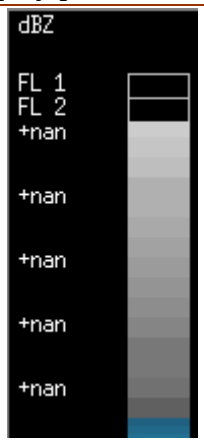
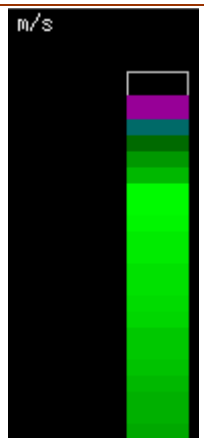
Color palettes used with Method 2 & 3

CVG can handle palette sizes from 1-256 with the following guidance. The palette size (number of colors in the CVG palette) should never be greater than the total number of data levels (including flag values).

The palette size for method 2 should be equal to the total number of data levels including flag values and:

- The first color must be black (background).
- Each color specified in the CVG palette file corresponding to a flag value must have a unique color value (the first two colors in our example).
- The remaining colors specified in the palette file do not have to be unique. Adjacent colors are normally assigned identical color values which effectively reduces the number of colors that must be interpreted visually.
- The colors chosen must change at the data levels corresponding to the threshold labels defined in the digital legend configuration file.

A smaller palette sized for label Method 1 could be used for Method 2 as well but is not recommended because of loss of control over the positioning of the color bands and therefore the position of the threshold labels.

COMMON CONFIGURATION ERROR: For Legend File Type 1, 2, and 3, **cvg** cannot detect mixing methods in the setting of the Legend File Type and the structure of the digital legend file. The resulting legend display provides an indication of this error.	
 **Using a method 2 legend file with the Legend File Type set at 1 -** results in all legend labels having a value of **+nan**	 **Using a method 1 legend file with the Legend File Type set at 2 -** results in all blank labels

Volume 4. CODE Utility Guide

Document 3. Archive II Disk File Ingest - play_a2 Tool

The ORPG play_a2 Archive II Utility

NOTE: There are two minor issues with Archive 2 playback of which users should be aware. The following do not cause any problem with correct data ingest.

1. There are two situations where the volume date and time displayed at the terminal output from **play_a2** is not correct. Both involve volume data files created with the **save_volume_file** utility.

    a. Volume files produced by the **save_volume_file** utility can have a volume date - time display that is late by less than a minute.
    b. The volume date time will differ significantly if the volume files were created by an early version of the **save_volume_file** utility on a little Endian platform (Linux). This includes some of the initial message 31 volume files being used for early super resolution testing and development.

2. The output of **play_a2** may include the following error message:

```
Bad data: Message type 0
```

Message type 0 is an unused segment of a more recent version of metadata included in the Archive 2 data stream which should be ignored by **play_a2**. Disregard the error message.

Part A. Introduction

The ORPG utility **play_a2** provides a capability for running an ORPG using Archive II disk files as a source of radar data. This provides a convenient source of specialized input data for testing. **play_a2** has several options for selecting which data to play and can feed short test cases or operate in a continuous play mode.

The CODE distribution includes a small collection of Archive II disk files from several radar sites that can be used by **play_a2** to drive the ORPG.

Part B. Creating Archive II Disk Files with play_a2

Description

See the man page for **`play_a2`** for information about creating Level II data files for ingest.

Part C. Input Archive II Disk Files into the ORPG with play_a2

See the man page for **play_a2** for information about using the interactive mode of data ingest.

SPECIAL NOTE: The environmental variable **AR2_DIR** must be defined for normal operation of the **play_a2** utility. The directory specified by **$AR2_DIR** is the default location of Archive II disk files to be used as input for the ORPG. Ideally, the most commonly used input data would be placed in **$AR2_DIR** and other test cases in subdirectories in the same tree structure. However, this normally cannot be done with the new **play_a2** utility because **play_a2** continues down subdirectories recursively when playing data files in the top directory. For a more detailed explanation see the NOTE under *Specifying Files to be Played – Using Command Line Options*.

Description

play_a2 reads WSR-88D Level II archive data from compressed files on disk and ingests the data into the ORPG. It can be used to read Archive II data directly from 8mm tape drives. The default behavior is to write the data in real-time to **$ORPGDIR/ingest/resp.0** (the standard buffer file for ORPG data input) including all radial data messages and RDA status messages.

play_a2 can read Level II bzip compressed files (**.bz2** file extension), Level II gzip compressed files (**.gz** file extension) from NCDC, and LDM files with the **.raw** file extension.

Note regarding ingest of historical radar data: When ingesting historical radar data into the ORPG, the Generation Data and Time are the current time and the Volume Data and Time are based upon when the data was recorded. **play_a2** has a switch to force the Volume Date Time to the current date time.

By default, all files conforming to the standard naming convention (described below) are read from the directory defined by the environmental variable **AR2_DIR**. An alternate directory may be specified with a command line option. Command line options also may be used to limit which files are read: the user may specify the filename prefix; a starting volume file in the series; and the number of volumes to read.

Alternatively, a "play list" may be used to specify which files to read, in which case the files will be read in the specified order and do not need to follow a standard naming convention.

Specifying Files to be Played - Using Command Line Options

Filename must follow the format "**[<PREFIX>_]<TIME_STAMP>[<SUFFIX>]**". The optional **<PREFIX>** (can be user specified) is typically a series of alphabetic characters (no numerals) followed by an underscore. The radar site 4 letter ID (.e.g, **KYUX**) is a convenient prefix. The **<TIME_STAMP>** is a series of digits that may include imbedded colons "**:**", underscores "**_**", or dashes "**-**". The **<TIME_STAMP>** must be unique for the files in a directory and must be of equal length in all filenames for the files to be played in order (leading zeros can be used). The **<SUFFIX>** must end in "**.bz2**", "**.Z**", or "**.gz**" indicating

the compression type. When used to create volume files the **play_a2** utility produces compatible filenames.

If using command line options, the directory to search for files is determined as follows:

- If an absolute path is specified with the **-d DIR** option, that path is used.
- If a relative path is specified with the **-d DIR** option, the path is formed as **$AR2_DIR/DIR**.
- If the **-d** option is not used, the directory is **$AR2_DIR**.

Either the **-r PREFIX** option or the **-s first_file** option is used to determine the first file read. If neither option is provided, all files will be played in time stamp order.

NOTE: **play_a2** (the new ORPG utility) does have a slightly different behavior than the retired CODE utility. **play_a2** proceeds recursively into subdirectories when playing back data files. This eliminates the storage of data files in multiple directory levels unless either the **-r PREFIX** option or the **-s first_file** option is used in the command line. As long as the data files in the subdirectories do not have the same prefix or name as specified for the parent directory they will not be played

To limit the number of files played to N, use the **-n N** option.

Specifying Files to be Played - Using a Play List File

Filenames do not need to follow any defined convention.

The **-d** option will be ignored if provided

For each file listed in the play list:

- When the full path to the filename name is specified, it will be used.
- When a relative path is provided (including a bare filename),
    - if a **PATH** is specified on line #1 of the play list it will be pre-pended to the individual line listing, or
    - if a **PATH** is not specified on line #1, **$AR2_DIR** will be pre-pended.

If the **PATH** specified on line #1 of the play list is not an absolute path, **$AR2_DIR** will be pre-pended.

Known Problems

- When NOT using the play list, files are sequenced based on the time stamp in the file name. If files from multiple radars are submitted for playback, and those files overlap in time, the volumes of those radars will be interleaved during playback.
- Aborting may require multiple ctrl-C's

Command Line Options

```
PLAY LIST METHOD OPTIONS

-p play_list    File listing which volume files to play, otherwise the
                command line options -s, -r, -n, -d are used to
                indicate which files to play. Either an absolute path
                or a path relative to the current directory can be
                specified. (see EXAMPLES for sample play_list formats)

---------------------------------------------------------------
COMMAND LINE METHOD OPTIONS

-s first_file   Play files starting with this filename that contain
                the same name prefix (see EXAMPLES).  Not compatible
                with the -p option.
                Default: Play all files in the directory.

-r PREFIX       Play files whose name begin with the prefix specified.
                Not compatible with the -p option.
                Default: Play all files in the directory.

-n N            Play only N volumes. Not compatible with the -p option.
                Default: Play all volume files in the sequence.

-d DIR          Directory containing files to play. Either a
                complete path is specified or a path relative to
                $AR2_DIR is specified.  Not compatible with the -p
                option.
                Default: The value of $AR2_DIR

---------------------------------------------------------------
GENERAL OPTIONS

-T              Simulates the proper delay between volume scans.  This
                should be used to reproduce the most accurate real
                time playback speed.

-R              Resets the playback data time to the current time. This
                results in the Volume Date Time in the product being the
                current time rather than the data collection time.  The
                product generation Date Time is always the current time.

-x speed        Multiplier for playback speed. A positive real number.
                For example "2" is double speed and "0.5" is half speed.
                Default: Play in real-time

-c              Continuous loop mode. When the last file in the directory
                is ingested play continues with the first file.  play_a2
                can be stopped with ctrl-C.
                Default: OFF

-t              No files ingested into the ORPG, only prints which
                files would have been played.
                Default: Normal Execution

-h              Show usage information
```

Command Line Examples

Example 1 - Read all volume files in the default directory (**$AR2_DIR**) continuously:

```
play_a2 -T -c
```

Example 2 - Read all volume files in the subdirectory **$AR2_DIR/test_case1** once and stop:

```
play_a2 -T -d test_case1
```

Example 3 - Read 10 volumes from the default location, starting with volume TAPE_1999:08:27:06:15:00.ar2.bz2, and play back at 1.5x speed:

```
play_a2 -T -s TAPE_1999:08:27:06:15:00.ar2.bz2 -n 10 -x 1.5
```

Play List Examples

Example 1 - Read the volume files specified in the file **test_volumes** without stopping (the file **test_volumes** must be in the current directory):

```
play_a2 -T -p test_volumes -c
```

Example 2 - Read the volume files specified in the file **/data/test_volumes** once and then stop:

```
play_a2 -T -p /data/test_volumes
```

Note the different behavior: When specifying the directory with the command line option **-d**, relative paths are relative to **$AR2_DIR**. However, when specifying the play list file with the **-p** option, relative paths are relative to the current directory.

Sample Play List Files

The following "play list" files are equivalent if the **AR2_DIR** variable is set to **/noaa/home**

Sample "play list" file with the optional PATH specifier (absolute path):

```
PATH = /noaa/home/TEST_FILES
volume_file1
volume_file2
volume_file3
```

Sample "play list" file with the optional PATH specifier (relative path):

```
PATH = TEST_FILES
volume_file1
volume_file2
volume_file3
```

Sample "play list" file without the optional PATH specifier:

```
TEST_FILES/volume_file1
```

```
TEST_FILES/volume_file2
TEST_FILES/volume_file3
```

Volume 4. CODE Utility Guide

Document 4. Product Distribution with nbtcp

This document describes how to use **`nbtcp`** in a CODE ORPG environment. **`nbtcp`** simulates a Class 1 user and communicates with an ORPG to initiate product distribution to a selected directory. The result is a collection of individual product binary files for selected products from a running ORPG. This is one method of testing the request parameters for selected products.

Part A. Overview of nbtcp and CODE

The Narrowband TCP/IP (**nbtcp**) test tool is designed to send Class 1 (and Class 2) user messages to an RPG and receive product messages from the RPG. Though originally designed to test the TCP/IP product interfaces, **nbtcp** is a convenient tool for collecting desired product messages from an RPG at run time. The RPG can be any RPG accessible by a TCP/IP network or an RPG running on the platform that **nbtcp** is being used.

When started, **nbtcp** connects to the specified RPG and sends a Routine Product Set (RPS) list message which is a list of products that the RPG should routinely distribute to **nbtcp**. The user must create a custom RPS list from a master RPS list that **nbtcp** always creates at startup.

The product messages are stored as individual files in the directory specified as a command line parameter. If a directory is not provided the product messages are discarded. The directory will also contain some files that are not final products: the General Status Message (GSM), Request Response Message (PRR), the Alert Adaptation Parameter Message (AAP), and the Alert Message (AM).

nbtcp can also send One Time Request messages, Alert messages, and Bias Table messages. The One Time Request Message could be useful for products using customizing parameters in the request message. The other messages are normally not needed for CODE users.

In the CODE environment, **nbtcp** is compiled and installed with the ORPG. If running an ORPG from a binary distribution, **nbtcp** must be installed separately from a tools CD.

nbtcp *Configuration Directory*. The **nbtcp** Configuration Directory is the location where

- **nbtcp** reads the **product_attr_table** file and creates the **master_rps_list.dat** file.
- The user places the custom RPS list file and the edited **one_time_req.msg** file.

The following instructions are based upon:

- using **nbtcp** while logged on to a CODE ORPG account,
- that CFG_DIR is defined in the standard way (= **`~/cfg`**), and
- that TOOLS_DIR is NOT defined, so **nbtcp** uses CFG_DIR as the configuration directory.

The TOOLS_DIR variable can be used to change the location of the configuration directory, since CFG_DIR cannot be modified. In this case

- The **product_attr_table** file, the custom RPS list, and the edited **one_time_req.msg** files must be copied to this new location
- the **master_rps_list.dat** file will be created in that location

Part B. Configuring nbtcp for the CODE Environment

1. Check that the CFG_DIR variable is set as follows (if not modify the **`~/orpg_env_cshrc`** file.
   CFG_DIR = $HOME/cfg (e.g. **`/home/code_9/cfg`**).
   a. The *ORPG configuration directory* is always determined by the value of CFG_DIR.
   b. Unless TOOLS_DIR is defined, this is also the ***nbtcp*** *configuration directory*.

2. Go to the ORPG configuration directory, **`~/cfg`**, and create backups of the following files
   cp tcp.conf tcp.conf.CODE
   cp comms_link.conf comms_link.conf.CODE
   cp service_class_table service_class_table.CODE

3. Edit the **comms_link.conf** file. Link number (LN) = 0 is used for radar data input by **`play_a2`**. Normally the first unused link is 1. The **comms_link.conf** file installed with CODE includes a commented out line that can be used for an **nbtcp** connection.
   a. Uncomment the line having A link number (**LN**) = 1, a line rate (**LR**) = 100000, and a comm manager (**CS**) = **`cm_tcp_`**.
   b. Change the comm manager from **`cm_tcp_`** to **`cm_tcp`**
   c. Add 1 to the value of the number of links, **number_links** (typically changing 1 to 2).
   d. Confirm the following values for the uncommented line. **LN**=**1**, **UN**=**1**, **CN**=**1**, **DN**=**1**, **PN**=**0**, **LT**=**Dedic**, **LR**=**100000**, **CS**=**cm_tcp**, **MPS**=**128**, **NS**=**2**, **LS**=**0**, **DEN**=**1**, **CLASS**=**98**, **TOUT**=**0**, **AW**= .
   e. Save the **comms_link.conf** file

4. Edit the **tcp.conf** file. For the **TCP_link_specification** block, choose a link that is not being used and renumber to the link number that was enabled in **comms_link.conf**.
   a. For CODE you can change link number **27** to **1**.
   b. The **S/C** should be **server**, and the **DED/DIAL** should be **DED**.
   c. Record the port number (**4491**) for later use.
   d. Change the **IP/Host** name to the host name of the computer on which the CODE RPG is installed or enter the actual IP address of the computer.
   e. Record the **password** (the default is **passwd**).
   f. Save the **tcp.conf** file.

5. Edit the **service_class_table** file. The only reason for a CODE user to edit this file is to increase the number of product requests permitted for the applicable class. Class 98 permits 160 requests (**n_req_prods 160**). If the custom RPS list being used results in more than 160 products being requested, this value must be increased.

6. For working purposes, copy three template message files (**alert_req.msg**, **bias_table.msg**, and **one_time_req.msg** from the **`~/tools/cfg`** directory into the chosen **nbtcp** Working Directory (in our case into **`~/data`**) and into the **nbtcp** Configuration Directory: **`CFG_DIR(~/cfg)`**. If these files are actually used, the modified copies must be placed into the **nbtcp** Configuration Directory.

Part C. Operating nbtcp in the CODE Environment

Basic instructions for running **`nbtcp`**

nbtcp *Working Directory*. This is the directory from which the **nbtcp** command is executed.

- The log file **`nbtcp.log.<port_num>`** is placed into the working directory.
- it is convenient to place the product files in a subdirectory of the working directory.
- The following procedures use **`~/data`** as the ***nbtcp*** *Working Directory*.

1. The **nbtcp** configuration changes made in the previous section are not applied until after the next RPG startup using the "**-p**" option: **mrpg -v -p startup**

2. The RPG must be running before starting **nbtcp**. The RPG should be started and data ingested using 'play_a2' following normal CODE procedures. For example:
   a. Execute '**mrpg -v -p startup**' and '**play_a2**'
   b. From a separate terminal window execute '**hci**' to launch the ORPG user interface.

3. In another terminal window, start **nbtcp**. Change to the desired **nbtcp** Working Directory (**`~/data`** in our case) and execute the following command. The format of the **nbtcp** command is
   nbtcp -p <port_num> -s <site> -d <directory> -w <password> <comp name>
   where **–p** is the port number of the line configured, **-s** is the 4 letter site id (used in some of the file naming conventions), **-d** is the directory to store the products (a subdirectory named **products** in our case), **-w** is the password for the line number, and the final parameter is the host name (must be in the **`/etc/hosts`** file) of the computer you are connecting to. The IP address can be used instead.
   a. '**cd ~/data**'
   b. '**nbtcp -p 4491 -s KMLB -d ./products -w passwd nimbus**'
   Since the default port for **nbtcp** is **4491** and the default password is **passwd,** those options can be omitted: '**nbtcp -s KMLB -d ./products nimbus**'

4. When **nbtcp** starts, a master RPS list file (**master_rps_list.dat**) is created / recreated in the **nbtcp** Configuration Directory (**`~/cfg`** in our case), which contains all the products contained in the **`product_attr_table`** file. A custom RPS list must be created from this file and entered at the next prompt. The header line in the master RPS list file must **not** be included in the customized RPS list. *See instructions later in this document.*

5. You will be prompted to **Enter RPS list filename:** Enter the name of the custom file you created (e.g., **my_rps_list.dat)**.

6. You are prompted to select one of the 6 file naming conventions for the stored product files. *An example and a description of each format are provided later in this document.*
   Select the product files naming convention:
   1. ProdMne_MsgCode_ElevCut_Site_id-Day-Year:Hr:Min:Sec"Z"
   2. ProdMne_MsgCode_ElevCut_VSDate_VSTime_VS#_Site_id
   3. MsgCode_ProdMne_VS#_ElevCut_Site_id

4. VS#_ElevCut_MsgCode_ProdMne_Site_id-Day-Year:Hr:Min:Sec"Z"
5. VS#_MsgCode_ProdMne_ElevCut_Site_id
6. "b"VS#_ProdMne_MsgCode_ElevCut_Site_id

Enter the number of the selection (**1 - 6**).

7. A message "**connecting to <computer name or IP address>**" appears.

8. An **nbtcp Tool Interface** should appear with the prompt

1 - Send One Time Request to Dedicated RPG
2 - Send Bias Table
3 - Send Alert Request
4 - Send RPS List
5 - Request RPS List
6 - Toggle product save flag (current state: save products)
9 - Terminate Tool

Enter selection:

(This step is not really needed unless you again change the list). To send / resend your RPS list enter **4** at the prompt. A message **Processing Send RPS List** appears and then another prompt.

Enter Selection From List:
1 - `my_rps_list.dat` file
2 - Enter new RPS filename
3 - Return to Main Menu

Enter **1** which should contain the name of your RPS list. A message **Sending RPS List** appears.

9. **nbtcp** can be monitored in several ways,
   a. Open yet another terminal and execute: '**mon_nb 1**' This will show the time and the product requests that are made, along with product information for the configured narrow band line (**`1`** in our case).
   b. A log containing errors can be found in a file named **`nbtcp.log.<port_num>`** in the **nbtcp** Working Directory (**`~/data`** in our case).
   c. From the **hci** you can look at the status of the link by clicking on **Comms** under the **Users** heading. A "*Product Distribution Comms Status"* window appears showing the line number being used and information about its status. This window allows you to **Disconnect/Connect** to the line, by clicking on the line first to select it, then clicking on the **Disconnect or Connect** buttons.

11. To terminate **nbtcp**,
   a. return to the terminal window with the **nbtcp Tool Interface**. Then **Enter selection: 9** to terminate tool. A message **Tool is terminating** appears.
   b. if **nbtcp** is in a state without the Tool Interface prompt, use '**control-C**' to terminate.

12. Use '**control-C**' to end **play_a2** and **mon_nb.**

The product files received from the RPG are stored in the designated product directory (**`~/data/products`** in our case).

Product File Name Formats

Choosing a format for the product file name involves some trade-offs. None of the file name directory sort orders is completely consistent. The product distribution time cannot be used to relate the product with the ingested volume data (if historical data is used) and the volume date and time are simple integers.

Format 1: **`DR_094_E3_KMLB_11-14-2006_14:42:26Z`** [Note 2] Filenames sorted by product mnemonic, product code, elevation [Note 1], and partially on distribution date-time [Note 2]. The date-time order breaks down when starting a new year.

Format 2: **`DR_094_E3_8473_34291_V002_KMLB`** [Note 3] Filenames sorted by product mnemonic, product code, elevation [Note 1], and volume date-time.

Format 3: **`094_DR_V002_E3_KMLB_8473_34291`** [Note 3] Filenames sorted by product code, volume sequence number (1-80) and elevation [Note 1]. Will not be in time order if more than 80 volumes of data have been ingested.

Format 4: **`V002_E3_094_DR_KMLB_11-14-2006_16:42:21Z`** [Note 2] Filenames sorted by volume number (1-80), elevation [Note 1], and product code. Will not be in time order if more than 80 volumes of data have been ingested.

Format 5: **`V002_094_DR_E3_KMLB_8473_34291`** [Note 3] Filenames sorted by volume number (1-80), elevation [Note 1], and product code. The files will not be played in time order if more than 80 volumes of data have been ingested.

Format 6: **`bV002_DR_094_E3_KMLB_8473_34291`** [Note 3] Filenames sorted by volume number (1-80), product mnemonic, product code, elevation [Note 1], and volume date-time.

Note 1: The elevation sort would work if a leading **0** was used on elevations less than **10**.

Note 2: The product distribution date-time used in Formats 1 and 4 are in real-time. If ingesting historical data, the distribution time has no relationship to the time of the data (volume time). For this reason Format 1 is not recommended for most users. Format 4 contains the volume sequence number which can be related to the data ingested if a small sample is used. This date time text is derived from the transmission date (HW 2) and time (HW 3 & 4) in the message header block.

Note 3: The integer volume date and integer volume time used in Formats 2, 3, 5, and 6 are intended for machine processing and are not human readable. The volume date is taken from HW 21 of the product description block and the volume time from HW 22 & 23 of the product description block.

Creating a customized RPS list.

When **nbtcp** starts, it always creates a new "**master_rps_list.dat**" file in the in the **nbtcp** Configuration Directory (**`CFG_DIR`**, **`~/cfg`** in our case). This file contains a list of all final products configured in the **product_attr_table** file also found in the **nbtcp** Configuration Directory. The list of products includes the default values of the product dependent parameters used in the product request message.

NOTE: The standard procedure to configure a new development product in CODE is to use a "snippet" **`product_attr_table`** file in the **`~/cfg/extensions`** directory. If you wish **nbtcp** to create an entry in the **master_rps_list.dat** file, the main configuration file, **`product_attr_table`**, in the ORPG configuration directory (**`~/cfg`**) must be modified instead of using a snippet.

The user must edit the "**master_rps_list.dat**" file to create a custom RPS list file, with a different name, which will contain only the desired products to use in the RPS list sent to the RPG. The custom RPS list file must be placed in the **nbtcp** Configuration Directory (**`~/cfg`** in our case).

a. The master RPS list header information must be removed.

b. Each line in the file must represent a product request. Comments and blank lines are not permitted.

c. Even though labeled "Prod ID", the first column represents a request number and is the same as the line number. If the request numbers are not in order with no breaks, some product requests may be missed.

d. The column alignment must be retained. If not, results will be unpredictable.

e. Some products may not be produced with the default product dependent parameters. See Volume 2 Document 2 Section III and Volume 2 Document 4 Section III of the CODE Guide for additional information. Table IIa in the RPG to Class 1 User ICD documents the product dependent parameters for existing products.
   Note: The permissible values for the elevation parameter are not the same as the default product generation table (see *Request message elevation parameters* in this document).

Making One-Time Requests with **nbtcp**

The One Time Request Message is useful for products having customizing parameters in the request message. The template message **one_time_req.msg** in the **nbtcp** working directory can be modified and placed into the **nbtcp** Configuration Directory (**~/cfg**) in our case). The initial contents of the file include:

```
Product {
    prod_code   85
    flag        1
    num_prods   1
    interval    1
    vol_date    0
    vol_time   -1
    params     900 300 1250 600 UNU UNU
}
```

Product{ is the beginning of the definition and the closing bracket **}** the end. For a basic one time request, you only need to be concerned with three fields. **prod_code** is the product code of the product. **vol_time** value of **-1** repeats the request for the current volume. If this field is set to a specific volume time or to **-2** (most recent existing product) the elevation parameter must be the single elevation angle requested. **params** are the product dependent parameters in the request message.

The master RPS list, **master_rps_list.dat,** can be used to determine which parameters are needed for the request message and what the default values are. "**UNU**" means the parameter is not used. See Volume 2 Document 2 Section III and Volume 2 Document 4 Section III of the CODE Guide for additional information. Table IIa in the RPG to Class 1 User ICD documents the product dependent parameters for existing products.
Note: The permissible values for the elevation parameter are not the same as the default product generation table (see *Request message elevation parameters* in this document).

More than one product request can be placed in the message. One limit to the number of products is that there can be only 10 requests for an individual product code (not counting multiple elevations) in a volume time frame.

After modifying the file, save it in the **nbtcp** Configuration Directory (**~/cfg** in our case) with the name **one_time_req.msg.**

At the **nbtcp Tool Interface** prompt

1 - Send One Time Request to Dedicated RPG
2 - Send Bias Table
3 - Send Alert Request
4 - Send RPS List
5 - Request RPS List
6 - Toggle product save flag (current state: save products)
9 - Terminate Tool

Enter selection:

Enter **1** to send the one time request message to the RPG.

Request message elevation parameters

For elevation products, the third product dependent parameter is used to represent the elevations requested. The encoding of this parameter is explained in Note 9 for Table IIa in the RPG Class 1 User ICD.

The basic method, listing the elevation angle (in tenths of a degree), is used in the master RPS list. The RPG will distribute the product with the same (or closest) elevation angle. For example:

UNU UNU 15 UNU UNU UNU

represents an elevation angle of 1.5 degrees.

Special cases use bits 13 and 14 of the integer value as follows. These methods can be used in both the custom RPS list and the one time request. However, for them to work in the one-time request message the **vol_time** field must be set to **-1** (the current volume).

a. If bit 14 is set and all other bits are not set, all elevations are requested.

UNU UNU 16384 UNU UNU UNU

b. If bit 13 is set and bits 14 & 15 are not set, this is a request for all elevations up to the angle represented in the low order bits 0-12.

UNU UNU 8235 UNU UNU UNU

represents all elevations up to 4.3 degrees [8192 (bit 13) + 43].

c. If bit 13 & 14 are set and bit 15 is not set, this is a request for all elevations up to the cut number represented in the low order bits 0-12.

UNU UNU 24579 UNU UNU UNU

represents the first three cuts [24576 (bit 13 & 14) + 3].

Volume 4. CODE Utility Guide

Document 5. Additional CODE / ORPG Tools

Several additional CODE utilities and ORPG tools are documented here. The ORPG tools are a site adaptation utility **`change_radar`**. The CODE tools are a color palette editor (for **`CVG`**), and a utility to make additional background maps for use with **`CVG`**.

Section I CODE Maintained Utilities

Section II ORPG Maintained Tools

Section I CODE Maintained Utilities

Part A. Editor for CVG Color Palettes - **edit_cvgplt**

NOTE: The **cvg_color_edit** utility provided with CVG 9.0 is replaced by the more capable **edit_cvgplt** utility provided with CVG 9.1. The new editor is a full-capability interactive palette editor (edit_cvgplt) which provides the capability of creating new palette files of a specified size, opening a palette without closing the editor, and saving the edited file to a selected file name.

For the previous palette editor (**cvg_color_edit**) several sample "blank" palettes were provided in the color palette directory **~/.cvgN.N/colors** which can be edited manually or which can be modified by the editor. With the current editor (**edit_cvgplt**) it is easier to create a new palette file using the editor's New function.

Each palette file contains three rows of integer numbers with the first row representing red, the second row green, and the third row blue. The number of columns is determined by the number of colors to be specified. The sample 16 color palette file, **blank_16.plt**, looks like:

```
16
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

The standard 16 color reflectivity palette, **refl_16.plt**, looks like:

```
16
  0   0   0   0   0   0   0 255 231 255 255 214 192 255 153 255
  0 236 160   0 255 200 144 255 192 144   0   0   0   0  85 255
  0 236 245 245   0   0   0   0   0   0   0   0   0 255 201 255
```

IMPORTANT: Do not modify any files in the **~/.cvgN.N/colors** directory except the samples: **blank_8.plt**, **blank_16.plt**, **blank_32.plt**, **blank_64.plt**, and **blank_256.plt** and any personal development palettes that you may have created.

If you wish to make a permanent change to an existing development palette distributed with CVG, contact the CVG Development Lead: Brian Klein: brian.klein@noaa.gov) .

Starting the Editor

The **Editor for CVG Color Palettes** is a basic X Windows program that aids in defining color palettes. This program (**`edit_cvgplt`**) replaces a previous program (**`cvg_color_edit`**), is easier to use, and provides full interactive capability to create new palette files and edit palette files.

The program is started by executing the command

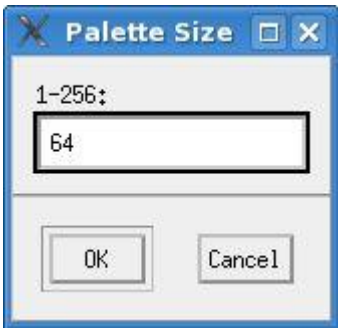
```
edit_cvgplt
```

The command can also include an option to open a named palette file when launching the program: **`edit_cvgplt  refl_16.plt`**. The filename can be either a complete path or a relative path.

Creating a new CVG color palette file

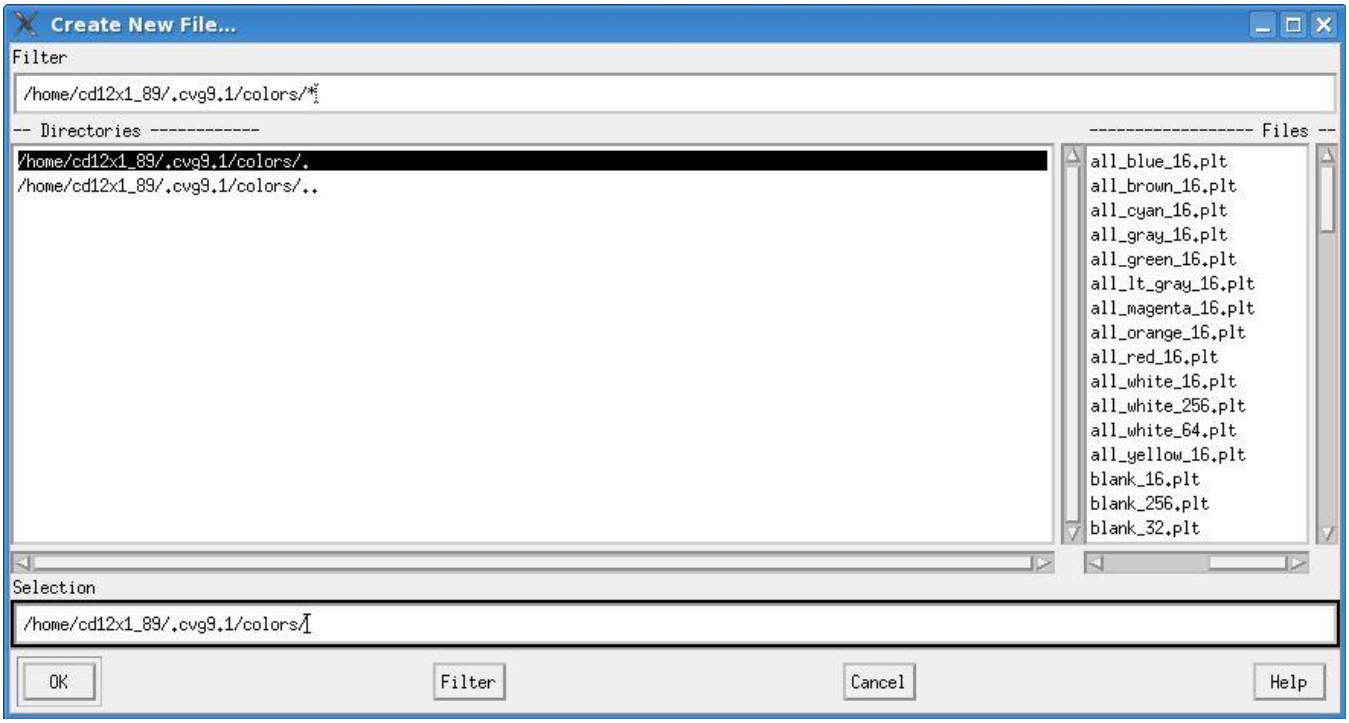
To create a new palette file click on the **New** button at the bottom of the dialog.

A prompt for the desired palette size (number of colors) pops up.



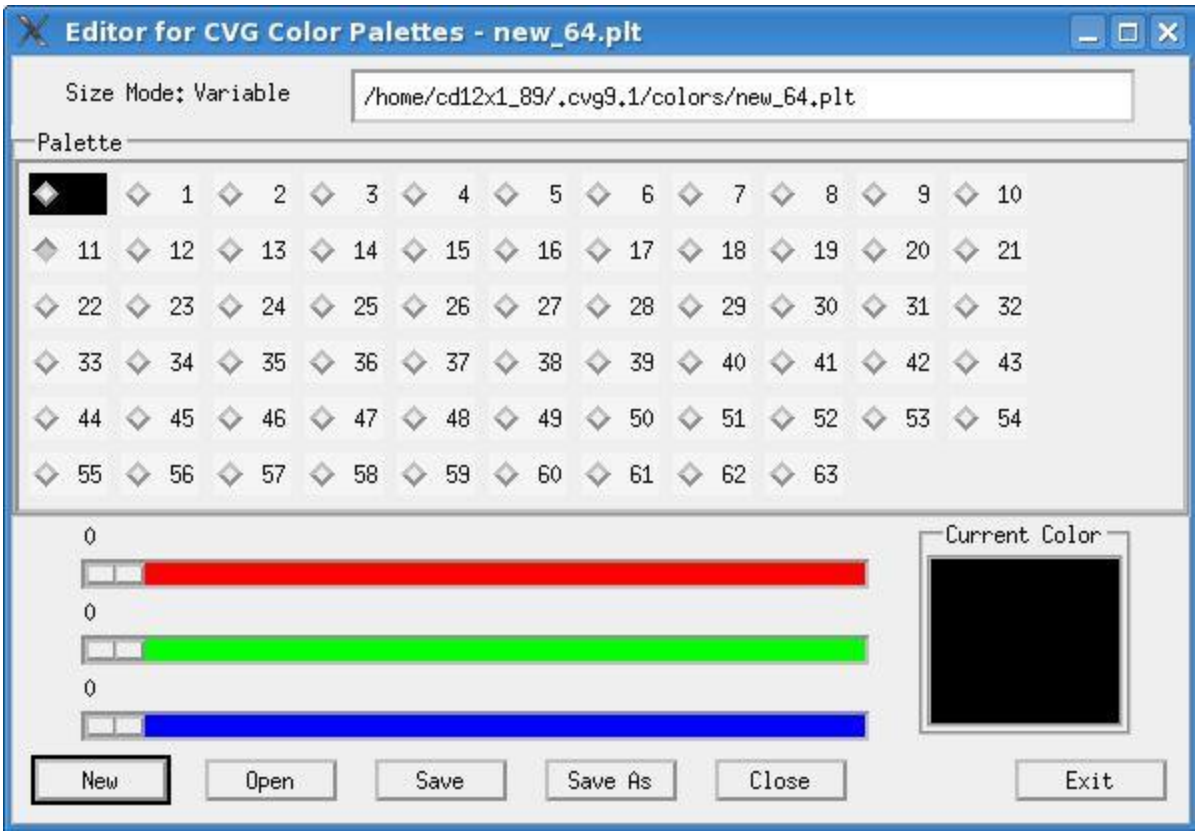
Enter the size and click on **OK**.

A file selection dialog appears with the local CVG configuration directory containing the palette files open. The directory can be changed if desired.



Enter the desired filename and click **OK**.

A new palette file is created with the first color black and the remaining colors a neutral gray.

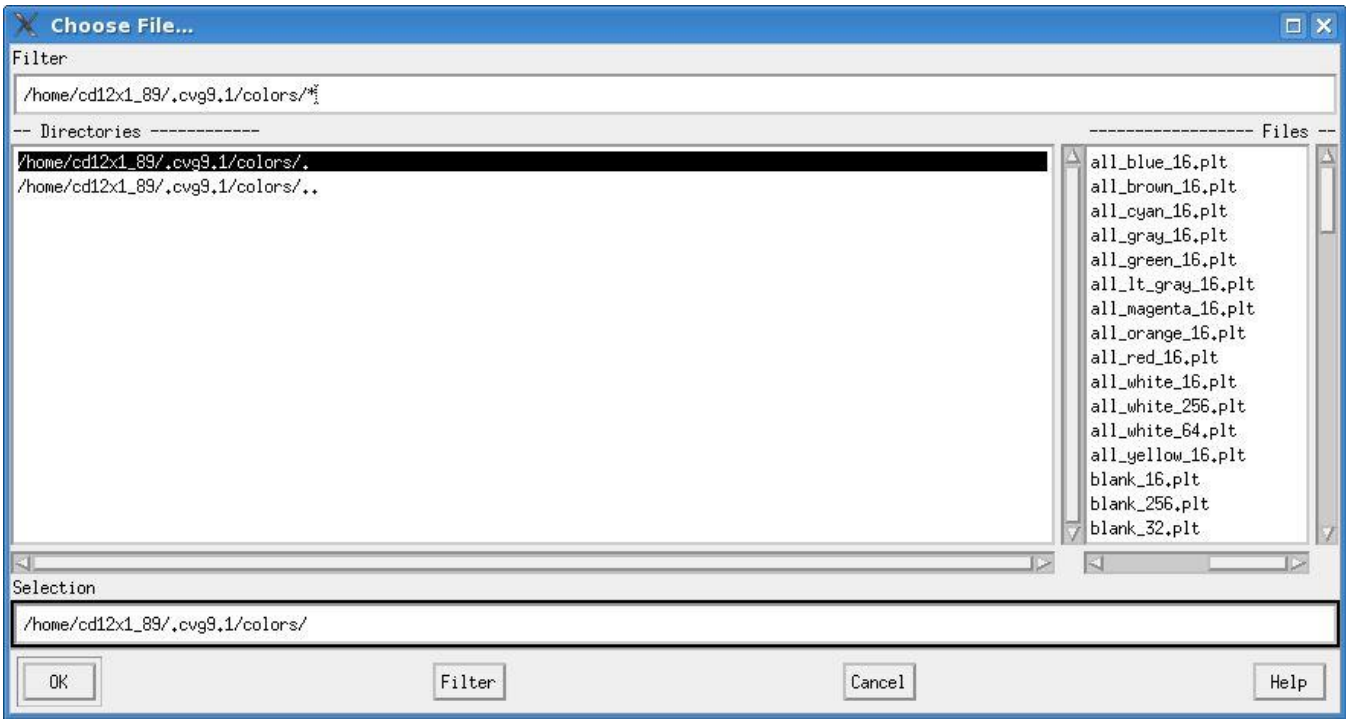


This new palette is already saved to the selected file. The file can be edited and saved as desired.

Opening an existing color palette

To open a palette file click on the **Open** button at the bottom of the dialog.

A file selection dialog appears with the local CVG configuration directory containing the palette files open. The directory can be changed if desired.

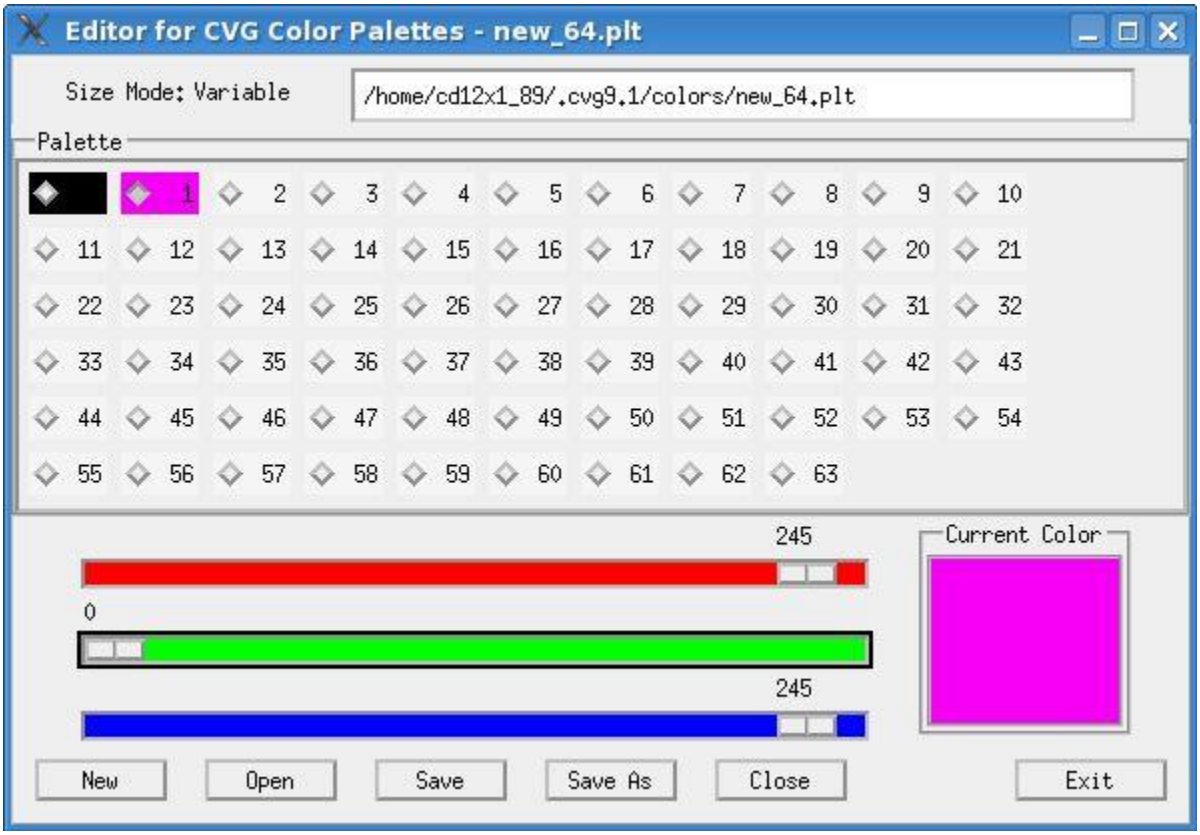


Enter the desired filename and click **OK**.

The selected palette file is opened.

Editing Colors

To modify the colors in the opened palette select the color button containing the color to be changed and then use the three color sliders for Red, Green, and Blue to set the desired color.



In this example, the just created palette file, new_64.plt is open. Button 1 has been selected and the slider bars set to create the new color R 254, G 0, B 245. The panel to the right of the sliders provides a larger area to aid in visually evaluating the color.

There is no need to save changes after each color is modified.

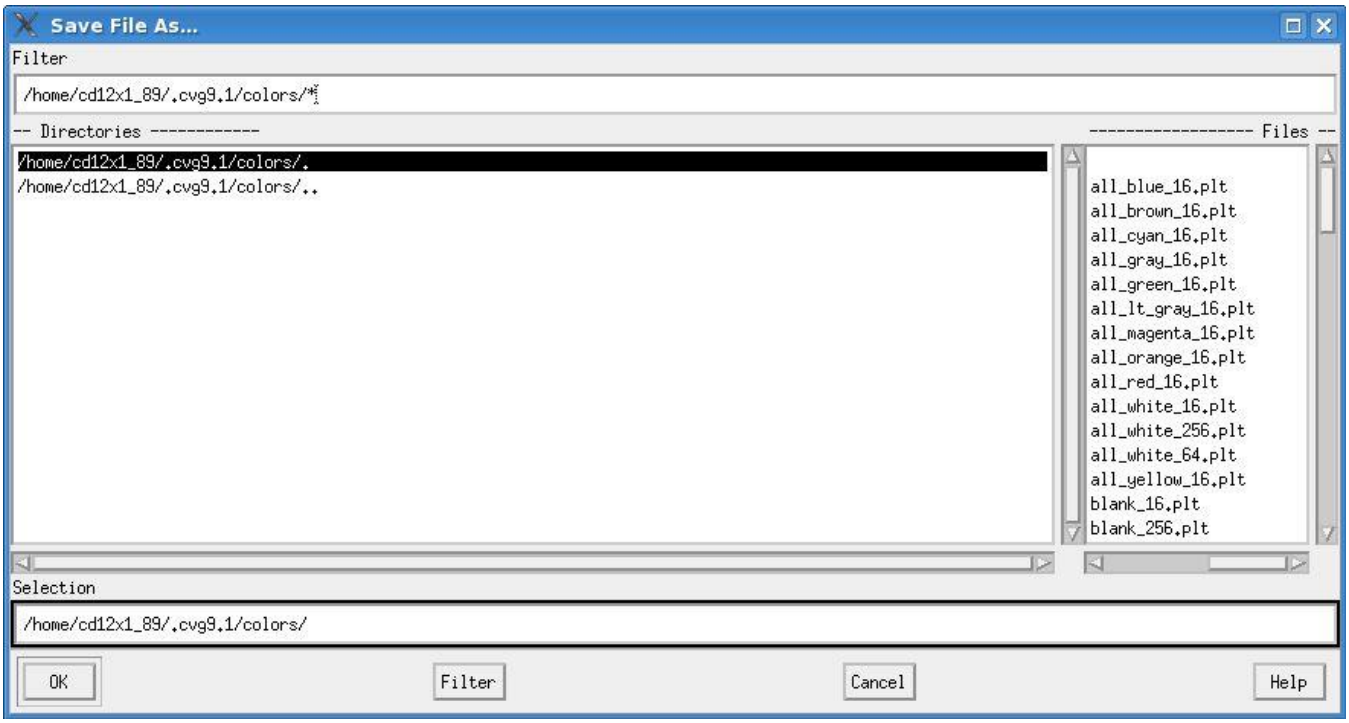
Saving the edited palette

To save the opened file with the existing filename, click on the **Save** button at the bottom of the dialog.

A prompt pops up to ensure you wish to overwrite the existing palette file.

To save the file with a different filename click on the **Save As** button at the bottom of the dialog.

A file selection dialog appears with the local CVG configuration directory containing the palette files open. The directory can be changed if desired.

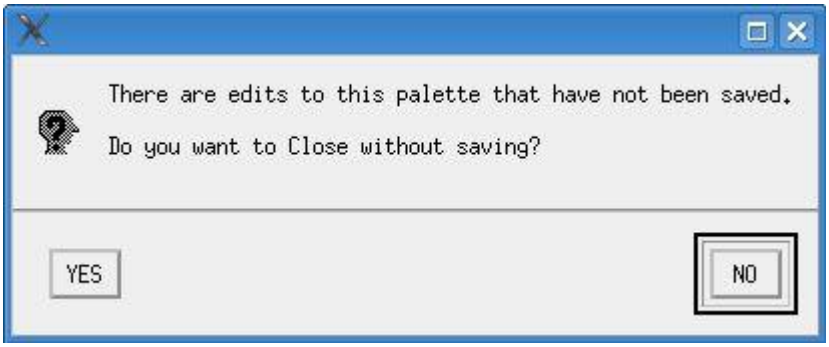


Enter the desired filename and click **OK**.

Closing the edited palette without saving

To close the open file without saving, click on the **Close** button at the bottom of the dialog.

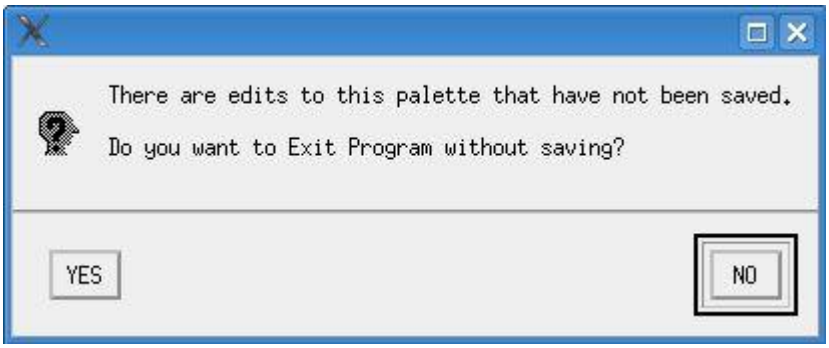
If edits have been made since last saved, a prompt appears to ensure that you wish to save without closing.



Exiting the editor

To exit the program without saving, click on the **Exit** button at the bottom of the dialog.

If edits have been made since last saved, a prompt appears to ensure that you wish to exit without closing.



The editor Size Mode

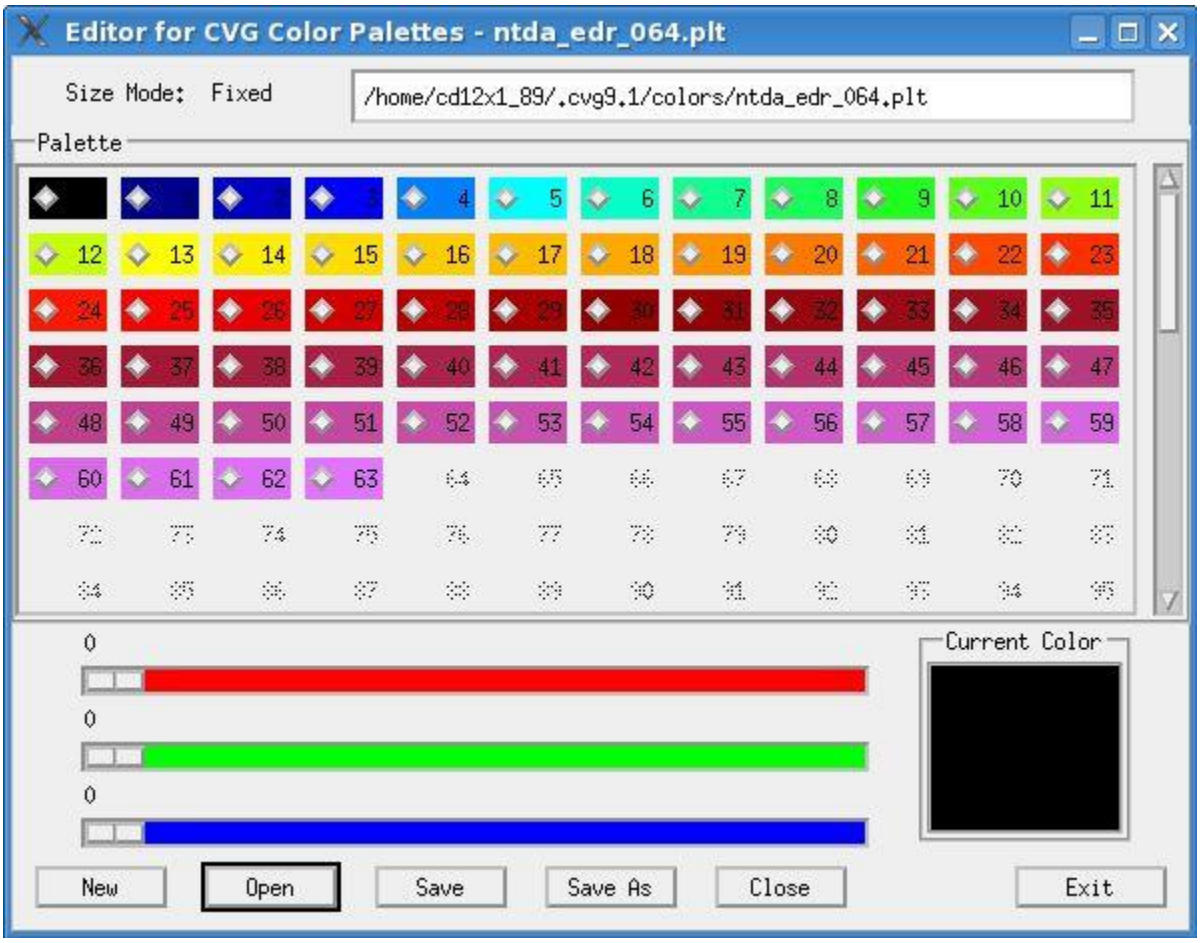
The editor can operate in two Size Modes: Variable (the default) and Fixed. All of the screen shots presented thus far have been in the Variable mode.

The primary purpose of the Fixed Mode is to improve response when remotely connected to the workstation. The Fixed Mode is selected with the **-f** switch:

```
edit_cvgplt -f
```

Variable Mode: The number of color toggle buttons is equal to the number of colors in the palette. The editor window resizes to accommodate the number of buttons.

Fixed Mode: The number of color toggle buttons is always 256 which is the maximum size of a CVG color palette. The window does not resize automatically. The toggle buttons beyond the size of the opened file are grayed out and not operational.



In this example the window has been manually resized and a vertical slider bar provides access to all of the color toggle buttons.

Part B. Background Map Utility - **map_cvg**

Beginning with Build 10

The original CODE map utility '**map**' has been retired and is no longer supported. It has been replaced by '**map_cvg**', a utility installed with CVG. **map_cvg** is much easier to use, has both a command line and graphical user interface (GUI) mode, and includes a script to automate making many maps with a single command.

Introduction

This background map utility **map_cvg** is installed with CVG and intended specifically to create background maps specifically for CVG. Map data files derived from USGS regional data files in the 'graphic' format are also installed with CVG. These files are a subset of the USGS data and are intended for use only with **map_cvg**.

Rather than making a background map for each applicable radar site, one could make a single large map covering a whole region. This is possible but a large map induces significant delay in the display of the product which becomes unacceptable during animation.

The major improvements over the previous utility are:

- Using range from the radar and radar location as an input parameter rather than calculated Latitude / Longitude limits to define the extent of the map.
- Providing an option to use a standard map size and name and to look up the location from an input ICAO identifier.
- Eliminating the 'quirks' in data entry.
- Providing a command line mode and a script that can automate making many maps at once.
- The map data includes type and rank identifiers to permit CVG to display features in different colors and to select subsets for display at runtime.

Getting Started

Whether using the command line mode or the interactive graphical user interface, use is simplified if **map_cvg** is executed from the directory in which the map data files are located. With a normal local installation, the data files are installed into the **~/tools/cvg_map** directory. With a standard global installation, the data files are installed into the **/usr/local/share/cvg_map** directory.

When initially installed, the map data files are in a bzip2 compressed format. They are automatically uncompressed the first time **map_cvg** (or either of the associated script files) is used.

The data file **us_map.dat** is used when making maps for sites in the contiguous 48 states. **ak_map.dat** is used for Alaskan sites and **hi_map.dat** for Hawaiian sites. Other locations (Korea, Taiwan, Puerto Rico, etc.) are not supported by the USGS data.

With both the command line and interactive modes, there are two methods of specifying input parameters to **map_cvg**.

- Method 1 - This is the easiest way to create maps. The required parameters are the name of the input data file and the 4 letter ICAO identifier of the operational radar site.
- Method 2 - This method provides additional flexibility. The required parameters are the name of the input data file, the name of the output map file, the Latitude and Longitude of the radar, and the range of the map.

When using the 4 letter ICAO identifier of the radar site as an input for Method 1, **map_cvg** looks up the Latitude and Longitude of the site. Identifiers for the standard operational sites in the United States are supported. If **map_cvg** reports and error in finding the location, Method 2 must be used.

The Command Line Mode

Ensure you are aware of the contents of the Getting Started paragraph before attempting to use this utility.

From the directory containing the map data, execute **map_cvg** with arguments appropriate for either Method 1 or Method 2 as described below.

= Method 1 =

Method 1 uses only two arguments. The name of the data file and the 4 letter ICAO identifier of the radar site. A standard range of 250 NM is used. The resulting map filename is **<icao>_cvg250_lnux.map** (on a Solaris platform the name would be **<icao>_cvg250_slrs.map**).

```
map_cvg -I us_map.dat -D kemx

where
   -I (Infilename) = Input data filename (complete / relative pathname)
   -D (ID) = Radar ID (e.g. KLWX) - uses std site Lat/Long, 250 range,
                                    and standard map name
```

= Method 2 =

Method 2 requires both input and output filenames, the Latitude and Longitude of the radar, and the desired range of coverage as parameters.

```
map_cvg -I us_map.dat -LA 41.263 -LO -112.448 -R 300 -M salt_lake.map

where
   -I (Infilename) = Input data filename (complete / relative pathname)
   -LA (Lat) = Latitude of the radar
   -LO (Lon) = Longitude of the radar (all entries assumed negative)
   -R (Range) = Cutoff range (in NM) for map image, 250 NM  recommended
   -M (Mapfilename) = Output map filename (complete / relative pathname)
```

= Using the script =

The script **create_all_cvg_maps** can be edited to provide the capability to create many maps at once. The script is installed in the same location as the data files: in **~/tools/cvg_map** for a local installation and in **/usr/local/share/cvg_map** for a global installation.

To use the script, un-comment the lines corresponding to the sites for which maps are desired (uses Method 1). The script must be executed from the installed location (the directory containing the data files).

```
cd9_1_3:cvg_map/        69 > ./create_all_cvg_maps

CREATING CVG BACKGROUND MAP FILES

This script must be executed from the directory containing
the CVG map data files (us_map.dat, etc.) which are normally
installed in the /home/cd9_1_3/tools/cvg_map directory.

This script can create all or selected cvg map files.
The data are limited to US NEXRAD sites.

If you wish to customize the map filenames, select a map range
other than 250 NM, or specify a custom center Lat Long, then
select 'n' at the prompt and execute 'map_cvg' with no arguments.

Simply uncomment the lines for the NEXRAD sites
for which you wish to create a map file, then execute
this script and enter 'y'

If you have not modified the script or are in the incorrect
directory, enter 'n'

DO YOU WISH TO PROCEED, 'y/n'

y
Creating standard map for radar ID  kabr
Map file: 'kabr_cvg250_lnux.map' finished.
Creating standard map for radar ID  kama
Map file: 'kama_cvg250_lnux.map' finished.
Creating standard map for radar ID  kapx
Map file: 'kapx_cvg250_lnux.map' finished.
Creating standard map for radar ID  karx
Map file: 'karx_cvg250_lnux.map' finished.
Creating standard map for radar ID  katx
Map file: 'katx_cvg250_lnux.map' finished.
cd9_1_3:cvg_map/        70 >
```

The Interactive Mode - Graphical User Interface

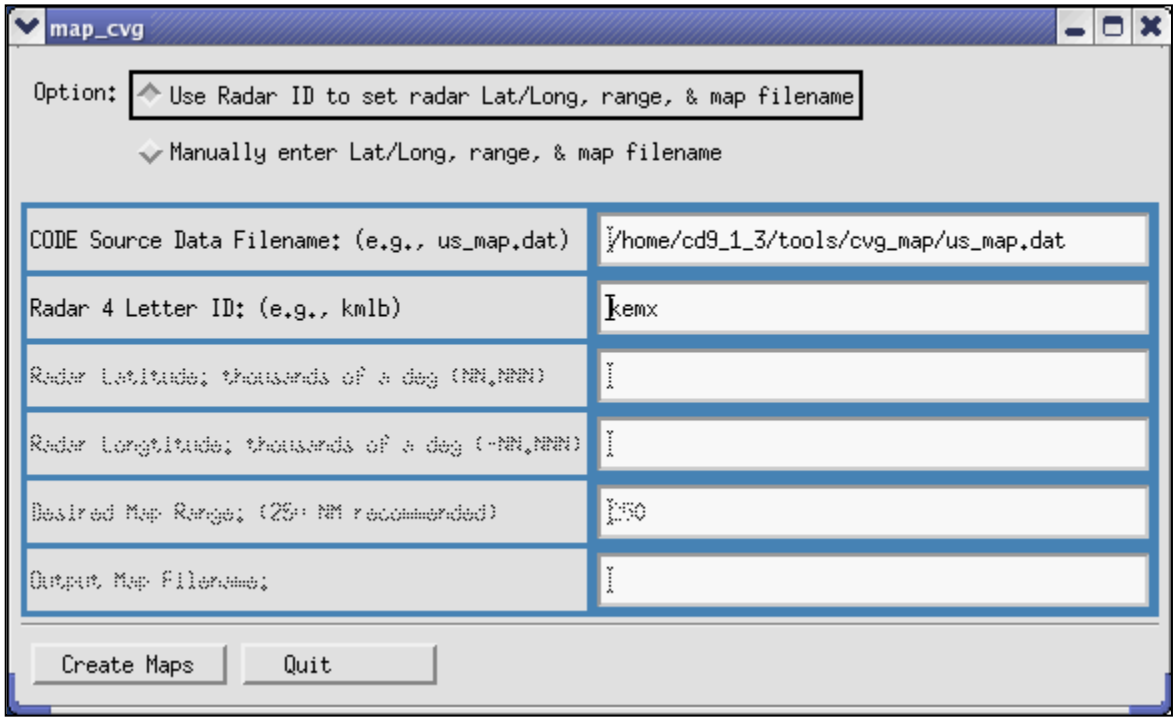
Ensure you are aware of the contents of the Getting Started paragraph before attempting to use this utility.

= Launching the graphical utility =

From the directory containing the map data, execute **`map_cvg`** with no additional arguments. This launches the utility in the graphical user interface mode.

= Method 1 =

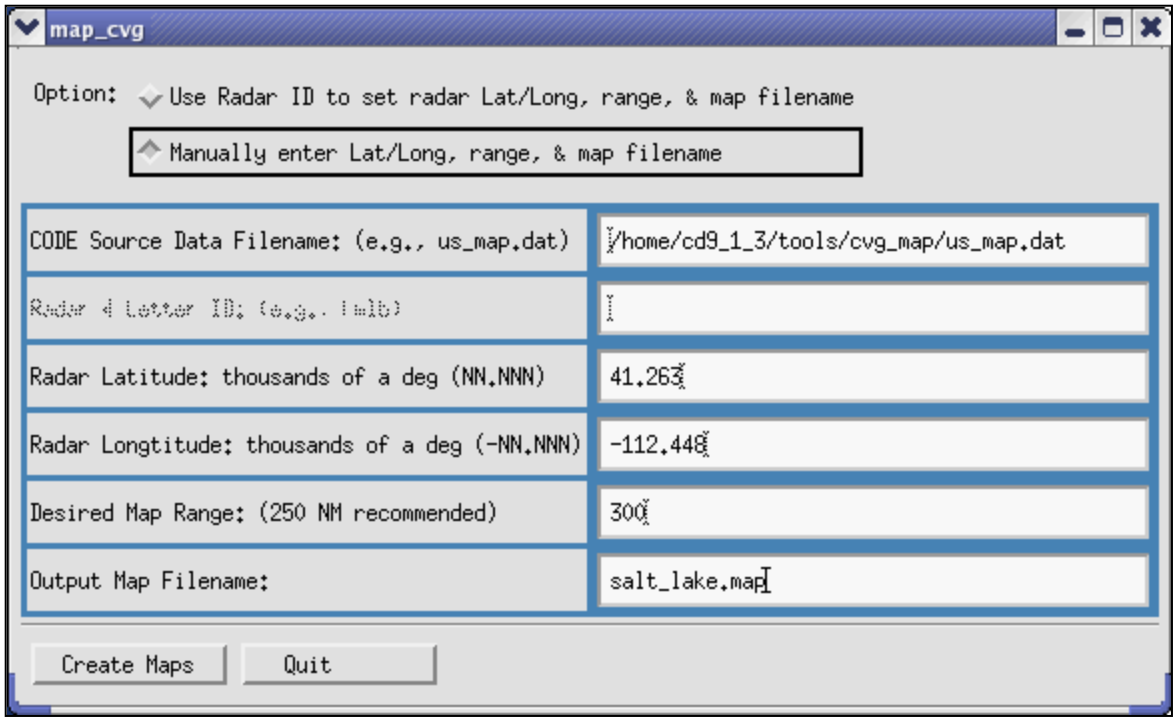
The utility opens with the first check box for Method 1 selected.



Only the two input boxes used for Method 1 are active. The Source Data Filename can be either a complete or relative pathname.

= Method 2 =

To use Method 2 in order to specify the map location, size, and filename, click on the second check box.



The five input boxes used for Method 2 are active. The Source Data Filename and the Output Map Filename can be either a complete or relative pathname.

Section II ORPG Maintained Tools

BUILD 12: Added procedures for adding site data into the change_radar.dat file.

Part A. Site Adaptation Data Tool - **change_radar**

This tool modifies the **site_info.dea** configuration file in order to change basic site information to conform to the location and elevation of the site that produced the basedata being ingested into the ORPG.

Instructions for using **change_radar**

1. For **change_radar** to work as intended with the CODE algorithm development environment the environmental variable **ORPG_NONOPERATIONAL** must be defined (this is accomplished during CODE setup in Volume 1 of the CODE guide).

2. In order to change site adaptation data, execute the **change_radar** script from the command line while logged in to the account into which the ORPG is installed. The four letter ICAO identifier is the primary parameter. Two additional parameters are useful in the development environment.

   -S
   This optional flag prevents an automatic shutdown of the RPG software (if running).

   -R
   This flag prevents an automatic restart of the RPG software when the script is finished. **This flag should always be used in the development environment because the restart command used does not include the -p flag.**

   For example, if the input data (Archive II tape, Archive II disk files, or BDDS on a LAN) are from Melbourne, FL, the command would be:

   ```
   change_radar -r kmlb -S -R
   ```

3. If the output of the script is the following error message:

   ```
   ICAO kmud ISN'T FOUND IN /home/code12/tools/cfg/change_radar.dat
   ```

   Either the four letter ICAO identifier was entered incorrectly or the data needs to be added to the **change_radar** data file. You can see a list of all supported sites by executing

   ```
   change_radar -l
   ```

   If the desired site is not in the list, the site data can be added using the instructions that follow.

4. If the output of the script looks like this:

```
SITE ADAPTATION DATA HAS BEEN CHANGED TO THE FOLLOWING:

RPG ICAO:       kmlb
RPG ID:         302
RDA LATITUDE:   28113
RDA LONGITUDE:  -80654
RDA ELEVATION:  116
```

   you have successfully modified site adaptation data.

   The changes are reflected in the following fields in **site_info.dea**
   site_info.rpg_name = KMLB
   site_info.rda_lat = 28113
   site_info.rda_lon = -80654
   site_info.rda_elev = 116
   site_info.rpg_id = 302

5. In order to replace the existing binary adaptation data files, they must be erased before the next ORPG start. This can be accomplished by using the **-p** option with the ORPG start command:

```
mrpg -p startup.
```

The **change_radar** utility has other options including the capability to list all supported radars and to interactively enter the desired site data. See the man page or execute: **change_radar -h**.

Adding New Site Data

If the data / products are from a radar site that is not an operational WSR-88D site, this radar's basic adaptation data will not be in data file used by `change_radar`. This could be a non-operational site, a non-WSR-88D site or experimental radar.

The data file for the **change_radar** tool is **~/tools/cfg/change_radar.dat**. Each line contains the minimum site adaptation data needed to display products. Amount other things, this information places the correct site ID, latitude, longitude, and height into the products header. This is required for correct background map display and to display the radar ICAO identifier and radar type in CVG.

For example, the line for KMLB (Melbourne FL) is:

```
kmlb,28113,-80654,116,302,MELBOURNE
```

When manually editing this file it is recommended no extra whitespace be added.

kmlb - the four letter ICAO identifier.
28113 - the latitude (28.113 deg)
-80654 - the longitude (-80.654 deg)

`116` - the antenna elevation in feet MSL.
`302` - the site ID

WARNING: The latitude and longitude must be entered to include the thousandth of a degree or the values will not be decoded correctly (3 decimal places of precision is assumed).

Volume 4. CODE Utility Guide

Appendices

Appendix A. Encoding Data into Unsigned Integers

Appendix B. CVG Color Palettes - List and Purpose

Appendix C. CVG Color Palettes - Examples in Color Editor

Vol 4. Appendices

Appendix A. Encoding Data into Unsigned Integers

The `Scale-Offset` formula

Beginning with Build 10, a new method was published for specifying the linear encoding and decoding of real data in unsigned integers contained in basedata messages. These formulas are NOT applied to any value of the scaled integer which represents a flag value (e.g.; "range folded"). They are only meaningful for the encoding and decoding of numerical data values.

```
encoded_integer = (float_value * SCALE) + OFFSET

float_value = (encoded_integer - OFFSET) / SCALE
```

The Scale-Offset formulas can be applied for decoding base data. Currently all basic moment data and Dual Polarization data fields reserve data levels 0 and 1 for flag values. In other words data levels 0 and 1 are not used for encoded numerical values. The API provides a function for decoding data.

The `Scale-Offset` formulas can be applied to product data which use unsigned integers and has data encoded in a manner described in this appendix. This would include traditional data packet 16 as well as data elements within the generic radial component. Data packet 16 contains an array of 8-byte integers (unsigned char). Beginning with Build 10, the generic radial component can support 8, 16, and 32 byte integer types. The maximum range of values in the encoded integer is limited by the type (unsigned char, unsigned short, or unsigned int) minus the integer values used for flag values.

These formulas do not include knowledge of the number of leading / trailing flag values. So

- When decoding product data values using the second formula, the user (or decoding system) must only apply the formula to the non-flag integer data values.
- When writing product data values, the product algorithm must set the flag values as needed and ensure that only valid meteorological values are encoded into the integer values contained in the product using the first formula.

Recommended Encoding Method for Unsigned Integer Arrays

Data packet 16 contains an array of 8-bit integers (unsigned char). Beginning with Build 10, the generic radial component can support 8, 16, and 32 bit integer types. Interpreting digital products depends upon a consistent method of encoding data into unsigned integers. Many of the existing products using data packet 16 use an encoding method that is consistent with the encoding of the basic base data moments: reflectivity, velocity, and spectrum width.

Though not required by the *ICD for RPG to Class 1 User*, the recommended method of representing real values in unsigned integer arrays (including data packet 16 and the generic radial component) is as follows:

1. The encoding method for products of the same type should be the same.

2. Whether the numerical real values are negative or positive, they are encoded into the integer values where the real value increases with the integer value. This results in the **`Scale`** parameter having a non-zero positive value.

3. The first (or lowest) integer value used is always 0.

4. The interval between the adjacent encoded numerical values is a constant. That means the encoded real numerical values increase linearly as the integer value increases.

5. Flag values are special purpose data values that are not numerically encoded. If the data contain flag values, they are not intermixed with encoded numerical values.

    a. Leading Flags - (if any) are represented by the beginning integer values (0, 1, ...). Many existing products have one or two leading flags. The first numerical value is first integer value after the last leading flag.

    b. Trailing Flags - (if any) are represented by the integer values immediately after the highest numerical value. Most current products use all available integer data values and have no trailing flag values. A few products use all of the available integer data values (in an 8-bit integer) and have one trailing flag represented with the integer value 255.

Not all products use this method of encoding data into an 8-bit integer. Product ID 134, DVL, uses a linear method for part of the data range and a non-linear method for the remainder. In addition the **`Threshold`** fields are used in a unique manner (see 'Providing Decoding Parameters in the Product' below).

8-bit Example

One example of using the recommended method is product ID 87, DBV. Binary value **0** represents a "Below Threshold" flag and binary value **1** represents "Range Folding". Binary **2** represents the minimum value of -63.5 meters/second. Binary **255** represents the maximum value of 63 meters/second. The real increment between numerical values is 0.5 meters/second.

Providing Decoding Parameters in the Product

The Product Description Block in the WSR-88D final product contains 16 Threshold Level data fields. These threshold fields were originally intended to explicitly define the threshold labels to be displayed for the 4-bit run length encoded products which were intended for display on the original display device. These 16 threshold fields (halfwords 31 - 46 in the final product message) are also used to provide decoding parameters for "digital products" which have real data values encoded into integer data levels.

Threshold Level Fields - The **Scale-Offset** Parameter Method (**Recommended**)

The new **Scale-Offset** formula can be used to encode and decode any product having a linear increment between encoded data values. The following threshold fields are being used by future Dual Polarization products 159 (DZD), 161 (DCC), and 163 (DKD) to describe the **Scale-Offset** coding.

Halfword	Field	Use
HW 31	**Threshold 1**	the SCALE in IEEE floating point format
HW 32	**Threshold 2**	
HW 33	**Threshold 3**	the OFFSET in IEEE floating point format
HW 34	**Threshold 4**	
HW 36	**Threshold 6**	the **highest data level having meaning, including flag values**
HW 37	**Threshold 7**	the number of leading flag values (can be 0)
HW 38	**Threshold 8**	the number of trailing flag values (can be 0)

Unlike the formula provided for the Original Method, the **Scale-Offset** formulas do not include knowledge of the number of leading / trailing flag values because the flag values do not affect the numerical coding directly. So

- When decoding product data values using the second formula, the user (or decoding system) must only apply the formula to the non-flag integer data values.
- When writing product data values, the product algorithm must set the flag values as needed and ensure that only valid meteorological values are encoded into the integer values contained in the product using the first formula.

The following formulas can be useful in encoding / decoding and display of a product.

The number of numerical data levels is

```
(HW_36_value + 1) - HW_37_value - HW_38_value
```

The lowest numerical data level is encoded into integer value:

```
0 + HW_37_value
```

The highest numerical data level is encoded into integer value:

```
HW_36_value - HW_38_value
```

The first trailing flag (if it exists) is represented by integer value:

```
HW_36_value - HW_38_value + 1
```

The encoding formula is:

```
integer_data_level = (real_value * HW_31_32_value) + HW_33_34_value
```

The decoding formula is:

```
real_value = (integer_data_level - HW_33_34 _value) / HW_31_32_value
```

8-bit Example

Using the same product as in the 8-bit example, product ID 87 (DBV) data fields would be encoded and decoded as follows.

To encode non-flag data values (that is integer values 2 - 255):

```
encoded_integer = (float_value * 2.0) + 129.0
```

To decode the (non-flag) integer values in the product:

```
float_value = (encoded_integer - 129.0) / 2.0
```

where
SCALE = 2.0
OFFSET = 129.0

Threshold Level Fields - The Original Parameter Method **(Not recommended for new development)**

The Legacy digital products (and many products added since) had a specific, though incomplete, method of providing information in the Product Description Block to aid in decoding integer values in data packet 16.

Halfword	Field	Use
HW 31	**Threshold 1**	contains the minimum value (encoded)
HW 32	**Threshold 2**	contains the increment (encoded)
HW 33	**Threshold 3**	contains the number of data levels

NOTE: The number of levels field (HW 33) is not used in a consistent fashion. Sometimes it is the maximum data level and sometimes the number of levels.

The information provided in the three parameters is not sufficient to decode the product.

a. Since the minimum value and the increment are stored on a 2-byte integer, an unstated scaling factor has to be applied to shift the decimal point and convert to real numbers. Some products had different scaling factors for the minimum value and increment. **NOTE:** This 'scaling factor' is not the same as the SCALE in the **Scale-Offset** formula.

b. There is no parameter specified for how many data levels are flag values. The decoding formula using minimum value, increment, and scale factors is not applied to flag values.

The Class 1 ICD does not provide an encoding or decoding function for the original parameter method. For product following the four encoding guidelines stated above:

To obtain the encoded integer value from the real floating point value:

```
data_level = num_flags +
    [real_value - (HW_31_value/min_val_scale)] * (incr_scale/HW_32_value)
```

To obtain the decoded floating point value from the encoded integer value:

```
real_value = (HW_31_value/min_val_scale) +
                [(data_level - num_flags) * (HW_32_value/incr_scale)]
```

where:
HW_31_value is the contents of HW 31 (encoded minimum value)
HW_32_value is the contents of HW 32 (encoded increment)
min_val_scale is the scaling factor used to convert the min value into a real number
incr_scale is the scaling factor used to convert the increment into a real number
num_flags is the number of leading value flags (beginning at data level 0)

The scaling factors are typically 10, 100, or 1000 which move the decimal point to the left to convert the integer data value into a real value. **NOTE:** This 'scaling factor' is not the same as the SCALE in the **scale-offset** formula.

One aspect of this coding may not be universally understood or consistently applied. With leading flag values, which integer data level does the encoded minimum value in HW 31 correspond to? For base data arrays in the radial base data messages, in the packet 16 arrays in the 8-bit base data products, and in the ITWSDBV product, the minimum value corresponds to the first non-flag data level. It should be noted that, for some products at least, AWIPS incorrectly applies the minimum value to data level 0. This has gone unnoticed because of the small size of the error.

8-bit Example

For product ID 87 (DBV), or ITWSDBV product,

- **Threshold 1** contains **-635**, for the minimum value (**HW_31_value**)
- **Threshold 2** contains **+5**, for the interval (**HW_32_value**)
- **Threshold 3** contains **256**, for the number of levels (**HW_33_value**)

Additional information is required to decode this product.

- **min_val_scale = 10** (which converts the minimum value to **-63.5**)
- **incr_scale = 10** (which converts the increment value to **0.5)**
- **num_flags = 2** (which associates the minimum value with data level **2**)

Using the decoding formula, integer data level 2 decodes to -63.5 and data level 3 decodes to -63.0.

Issue 1

The biggest issue is that even though the *ICD for RPG to Class 1 User* describes how existing products use the **Threshold** fields in the *Product Description Block*, no formal guidance for future use has been developed.

Issue 2

The *ICD for RPG to Class 1 User* does not explicitly state how the minimum value in **Threshold 1** is applied.

Issue 3

The description for the use of **`Threshold 3`** in the *ICD for RPG to Class 1 User* is inconsistent, or at least not clear. The ICD states this value should have a range of 0-255 but that the meaning of the value is 'the number of data levels', which could be 256. This may be related to existing products not using this header field in the same manner.

Issue 4

Not all products use this method of encoding data into an 8-bit integer. For example, product ID 134, DVL, uses a linear method for part of the data range and a non-linear method for the remainder. In addition the **`Threshold`** fields are used in a unique manner.

Because the information in `Threshold 1` - `Threshold 3` is incomplete, the CODE product display tool CODEview Graphics (CVG) does not currently use the `Threshold` fields in the product header. CVG configuration files for each digital product are used to provide all information required.

16-bit Example

TBD IN A FUTURE EDITION OF CODE

As with the 8-bit data scaled integer arrays, the CODE product display tool CODEview Graphics (CVG) does not actually use the **`Threshold`** fields in the product header. CVG configuration files for each digital product are used to provide all information required.

Threshold Level Fields - Additional Parameter Methods

In addition to product 134 (DVL) which uses a unique partially non-linear method, other products do not follow either the Original Method or the Scale-Offset method. Most of the proposed Dual Pol products are being modified to use the **`scale-offset`** parameter method. .Several that do not follow either the Original Method or the Scale-Offset method are:

Products 156 (EDR) and 157 (EDC)

These products include the Scale, Offset, the number of levels, and number of leading flag values but do not encode them in the thresholds following the recommended method, which was defined after these products were developed.

Future Products 165 (DHC) and 163 (HHC)

These products use the concept of an enumerated type with a table defining the enumeration.

CVG use of Threshold Level Fields

For products using the Legacy 'original method', the information in **`Threshold 1 - Threshold 3`** is incomplete, the CODE product display tool CODEview Graphics (CVG) does not actually use the **`Threshold`** fields in the product header. CVG configuration files for each digital product are used to provide all information required. Currently CVG can use the 7 factors in the 'original method' (contained in CVG product display configuration files) for decoding the data values into the legend threshold labels in for 'Method 1' of legend configuration.

Beginning with Build 12, **for those products using Method 5 for display configuration, and having the `Scale-Offset` parameters in the product description block, CVG can use the contents of the threshold fields to decode the data levels for information display in for calculating legend threshold levels**. This permits the legend labels to be calculated dynamically as the increment in certain products changes (e.g., DSP).

For those products not using the **`Scale-Offset`** parameters in the product description block, parameters in legend configuration files will be used to decode the data levels and explicitly state static threshold labels.

NOTE: The dynamic change in legend threshold labels will NOT be provided for products like DSP if they are not modified to provide the Scale Offset parameters in the product description block.

Decoding will not be provided for any product that cannot be described with Scale Offset parameters. This includes products that do not have a linear encoding like DVIL. Both the recommended Method 5 and the original Method 2 will provide the capability to explicitly state threshold labels for products like DVIL that do not have a linear encoding.

.

CVT use of Threshold Level Fields

Beginning with Build 12,

- **for those products having the `Scale-Offset` parameters in the product description block, CODEview Text (CVT) can use the contents of the threshold fields to decode the data levels for information display.** This permits the legend labels to be calculated dynamically as the increment in certain products changes (e.g., DSP).
- **CVT can also decode any scaled offset-product whose encoding can be described using `Scale-Offset` parameters in a configuration file.**

Decoding will not be provided for any product that cannot be described with Scale Offset parameters. This includes products that do not have a linear encoding like DVIL. Both the recommended Method 5 and the original Method 2 will provide the capability to explicitly state threshold labels for products like DVIL that do not have a linear encoding.

Vol 4. Appendices

Appendix B. CVG Color Palettes - List and Purpose

1. Color Palette Associations Based on the Product Specification Document

The following associations are based upon the WSR-88D Product Specification. The palettes in the first list could be used for new 16/8-level products. For display by CVG they are associated with a product via Product ID (in the configuration entry for that product).

	Palette Name	Description	Products Associated via CVG Preferences (Prod ID)
Non-digital Product Palettes (radial & raster data)	refl_16.plt	16-level reflectivity	Base Reflectivity(2,4,6), Comp Reflect(23,25), Hybrid Scan Reflect(58), Severe Weather Reflect(60), Cross Section Reflect(84), Vertically Integrated Liquid(85), Comp Reflect AP(123,125), User Sel Comp Refl (137)
	refl_8.plt	8-level reflectivity	Base Reflectivity(3,5,7), Combined Moment(17), Comp Reflect(24,26), Layer Comp Reflect Ave(36,37,116), Layer Comp Reflect AP (44), Layer Comp Reflect Max(38,39,117), Cross Section Reflect (111), Comp Reflect AP(124,126)
	vel_16.plt	16-level velocity	Base Velocity(11,13,15), Severe Weather Velocity(62), Storm Rel Vel Map(68), Storm Rel Vel Region(69), Cross Section Vel(83)
	vel_8.plt	8-level velocity	Base Velocity(12,14,16), Cross Section Vel(112)
	sw_8.plt	8-level spectrum width	Base Spectrum Width(8,9,10), Severe Weather SW(61), Cross Section SW(82)
	precip2_16.plt	16-level precipitation	Surface Rainfall Accum 1HR(105), Surface Rainfall Accum 3HR(106), Storm Total Rainfall Accum(107), Snow Products (144,145,146,147,150,151)
	shear_16.plt	16-level shear	Severe Weather Shear(59), Combined Shear(114), Combined Shear Contour(115)
	et_16.plt	16-level echo tops	Echo Tops Contour(28), Echo Tops(29)
	vad_16.plt	8-level VAD	Velocity Azimuth Display(110)
	swp_4.plt	4-level SWP	Severe Weather Probability(63)
	rec_refl_11.plt	11-level rec reflectivity	Clutter Likelihood Reflectivity (132)
	rec_dop_12.plt	12-level rec doppler	Clutter Likelihood Doppler (133)

For display by CVG, the following palettes are inherently associated with the individual data packets. The existing associations should be normally be retained.

	Palette Name		Inherently Associated Packet Types (Pkt Code)
Palettes for Text and Special Symbols	hail.plt		HDA Hail Data(19)
	meso.plt		Mesocyclone Data(3), Correlated Shear Meso(11)
	storm_id.plt		Storm ID Data(15)
	symbol_pkt2.plt		Special Symbol(2)
	symbol_pkt20.plt		Point Feature Data(20)
	text_1.plt		Text and Special Symbols No Val(1), Vector Arrows(5), Linked and Unlinked Vectors No Val(6,7)
	text_8.plt		Text and Special Symbols Uniform Val(8), Linked and Unlinked Vectors Uniform Val(9,10), Contour Vectors Linked(0E03x), Contour Vectors Unlinked(3501x)
	tvs.plt		TVS Data(12), ETVS Data(26)
	wind_5.plt		Wind Barb Data(4)

The first entry is an example of overriding an inherent association with a data packet type. The palette all_yellow_16.plt is assigned to data packet 6.

	Palette Name		Products Associated via CVG Preferences (Prod ID)
Palettes for Special Purpose	all_yellow_16.plt		MDA Product(141)

2. Color Palette Associations for Digital Products

The following associations are not based upon the WSR-88D Product Specification. In the legacy WSR-88D, digital data arrays and 256 level products were not displayable and color palette assignments were not made. The following palettes are provided with CVG to permit graphical display of these products. The palettes in the first list could be associated with new digital products using packet code 16.

Future versions of this document will include some of the dual polarization palettes when the design is finalized.

	Palette Name	Description	Products Associated via CVG Preferences (Prod ID)
Digital Product Palettes	nws_64.plt	64-level reflectivity	expansion of the 16-level reflectivity palette
	refl_64.plt	64-level reflectivity	alternate 64-level palette - Prod 138
	vel_64.plt	64-level velocity	expansion of the 16-level velocity palette
	nws_66.plt	66-level	1 leading flag and 1 trailing flag - based upon nws_64.plt - Prod 108
	drefl_66.plt	66-level	2 leading flag values - Prod 57
	vel_66.plt	66-level	2 leading flag values - based upon vel_64.plt - Prod 87
	dvil_255.plt	255-level	2 leading flag values and optionally one trailing flag - Prod 134
	hreet.plt	200-level	Prod 135
	refl_256a.plt	256-level	sample 256 level palette for reflectivity and precipitation oriented products
	hires_refl.plt	256-level	2 leading flags - Used by AWIPS - Prod 94; Prod 153 and DP test Prod 600
	hires_vel1.plt	256-level	2 leading flags - Used by AWIPS - Prod 99; Prod 154 and DP test Prod 601
	hires_spw.plt	256-level	2 leading flags - Prod 155 and DP test Prod 602
	ntda_edr_064.plt	64-level	1 leading flag - Prod 156
	ntda_edc_008.plt	8-level	Prod 157

The colors are inherently assigned to the individual packets for display by CVG.

	Palette Name		Inherently Associated Packet Types (Pkt Code)
Palettes for Data Array Packets	dpa_rate_8.plt		Precip Rate Data Array(18)
	drefl_66.plt		Digital Radial Data Array(16) - this association should normally be overridden
	refl_66.plt		Digital Precip Data Array(17)

3. Other Color Palettes

The following color palettes are special purpose. The first is used for displaying symbols in data packet 28. The second can be used generically for configuration method 5. The third is used by the CVG map display system.

	Palette Name		Purpose
Palettes for Special Purpose	symbol_pkt28.plt		packet 28 development
	generic_method_5_86.plt		For generic use with configuration method 5.
	cvg_map.plt		non-product, for maps

The following color palettes are being used for product currently in the development stage and are subject to change.

	Palette Name		Products Associated via CVG Preferences (Prod ID)
	cc_16.plt cc_64.plt cc_256.plt		Prod 160 CC ---------- Prod 161 DCC and DP test Prod 605
	dp_precip_diff.plt		Prod 174 , Prod 175
	dpr_66v1.plt	generic radial scaled integer	Prod 176 DPR
	hc_16.plt hc_256.plt		Prod 164 HC Prod 165 DHC
	kdp_16.pt kdp_256.plt		Prod 162 KDP Prod 163 DKD
	ml_16.plt		Prod 166
	phi_64.plt generic_method_5_86.plt	debug product	Prod 340 DPH DP test Prod 604
	quality_256.plt	debug products	Prod 334 QSZ, Prod 335 QZD, Prod 336 QRH, Prod 337 QKD, Prod 338 QTZ, Prod 339 QTP
	snr.plt	debug products	Prod 341 DSN
	texture.plt	debug products	Prod 344 DTZ, 345 DTP
	zdr_16.plt zdr_256.plt zdr_raw_256.plt zdr_raw_256_bias.plt		Prod 158 ZDR Prod 159 DZD DP test Prod 603 Alternate palette for Prod 159 DZD and test Prod 603 and 703.

Note 1: The following data packet types do not have a color palette associated with them. SCIT Past Position Data(23), SCIT Forecast Position Data(24), and STI Circle Data(25). CVG displays them using the current foreground color (white).

Note 2: Several palettes provided with CVG are not associated with any product or any packet type. These include: echo_16.plt, meso_ru.plt, mda.plt

Vol 4. Appendices

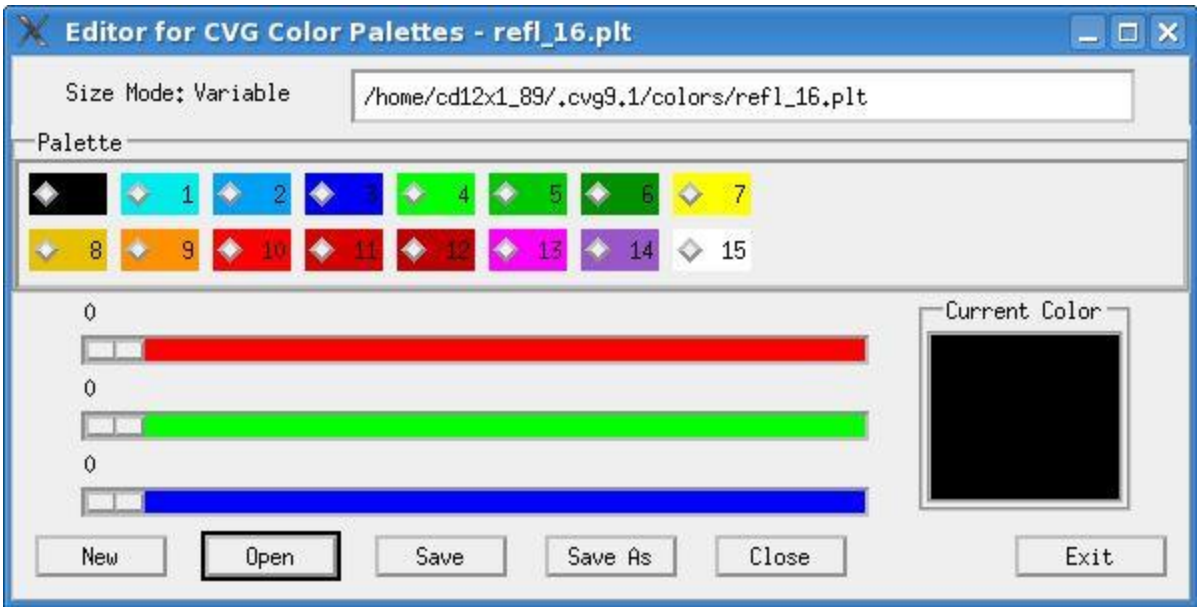
Appendix C. CVG Color Palettes - Examples in Color Editor

The purpose of this document is to display a few of the palettes provided with **cvg** for use with digital products.

Generic Color Palette

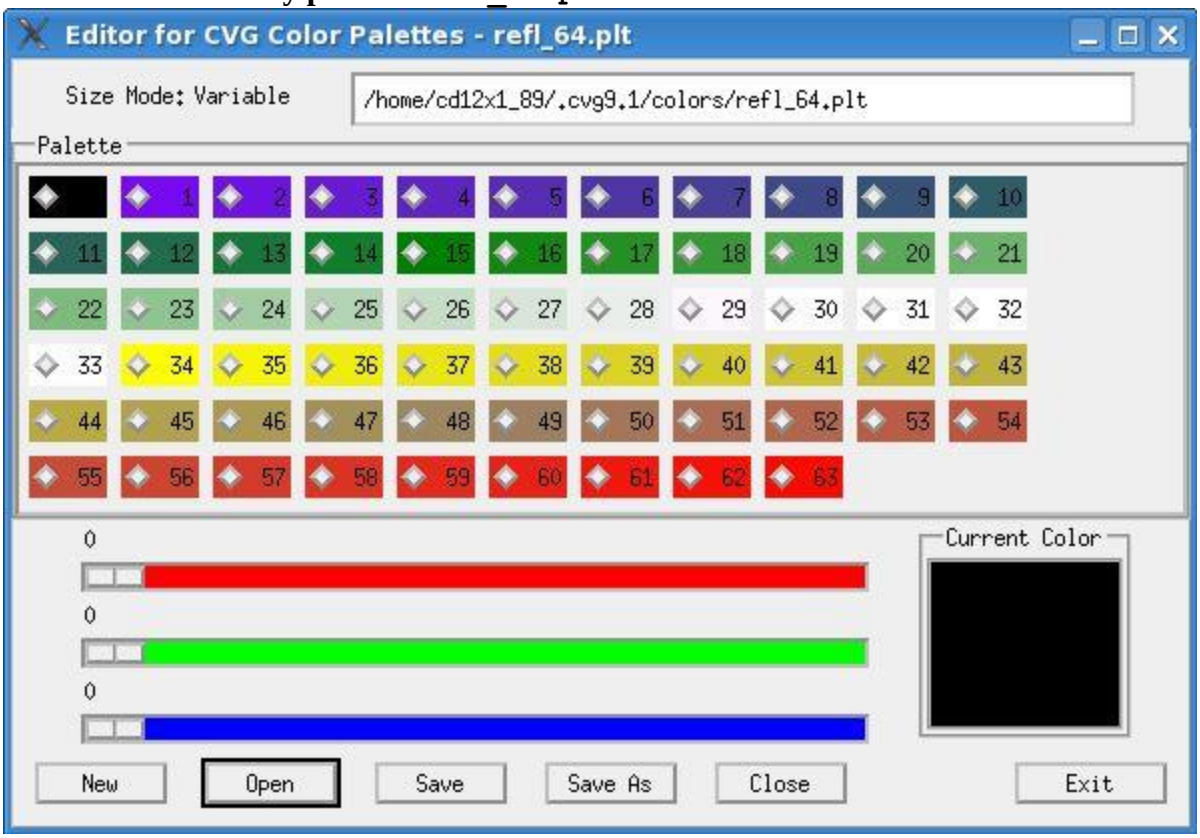
A sample generic color palette: `generic_method_5_86.plt` Designed to be used with any product configured with method 5. Within each sequence of colors, the colors are defined to enable adjacent colors to be visually distinguished on a good color monitor.

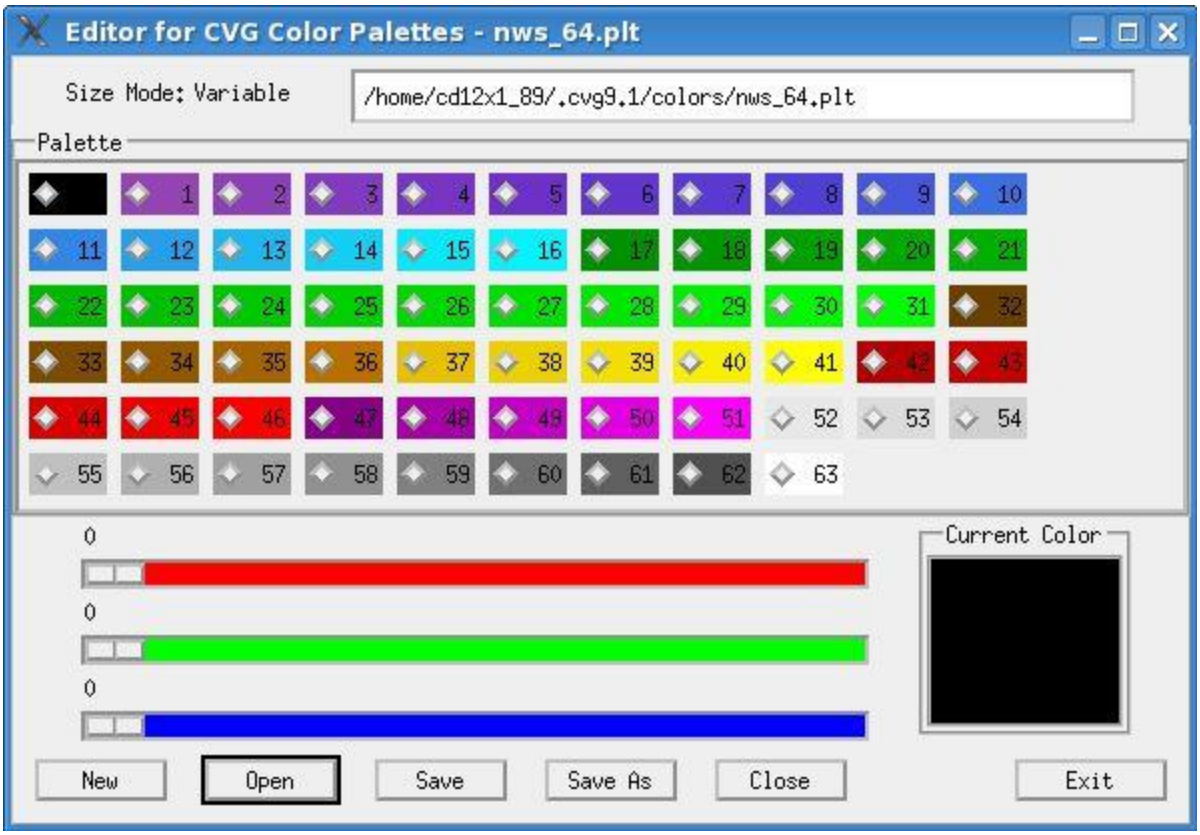
Reflectivity Related Palettes



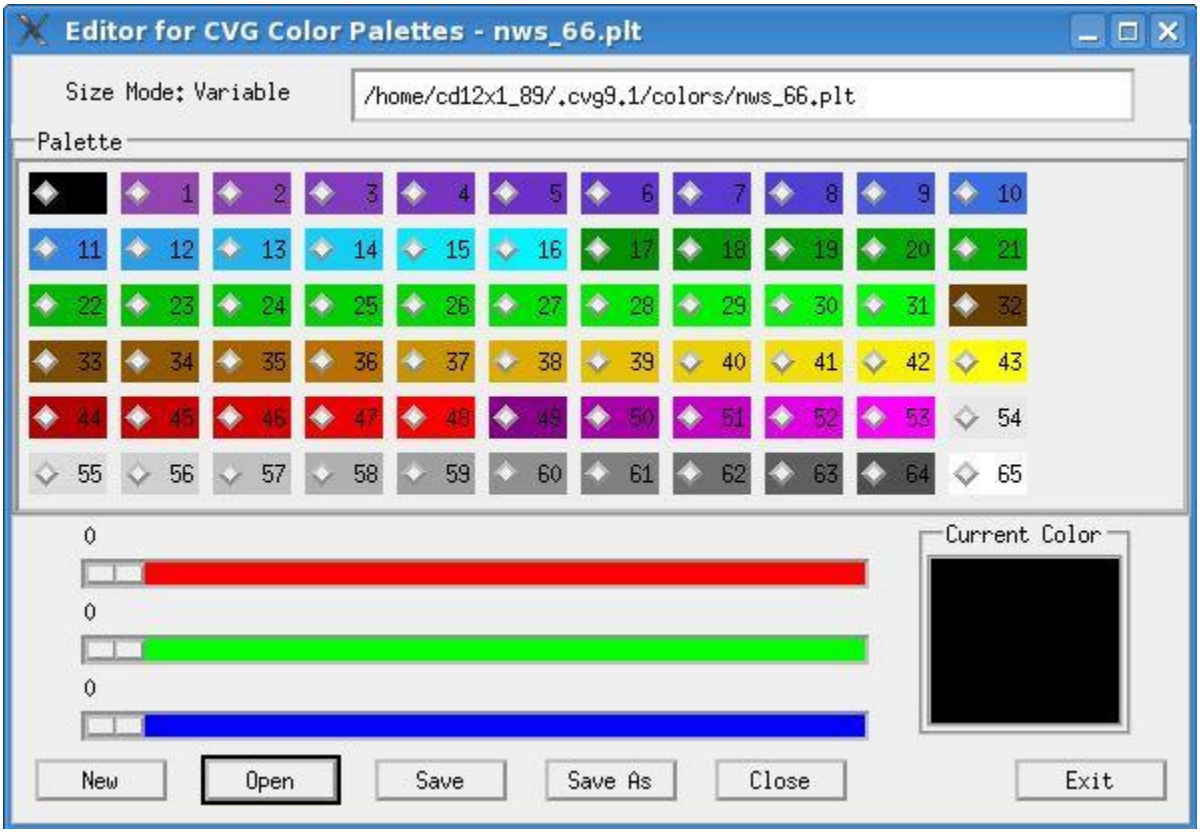
The standard 16 color reflectivity palette: `refl_16.plt`

A 64 color reflectivity palette: `refl_64.plt`

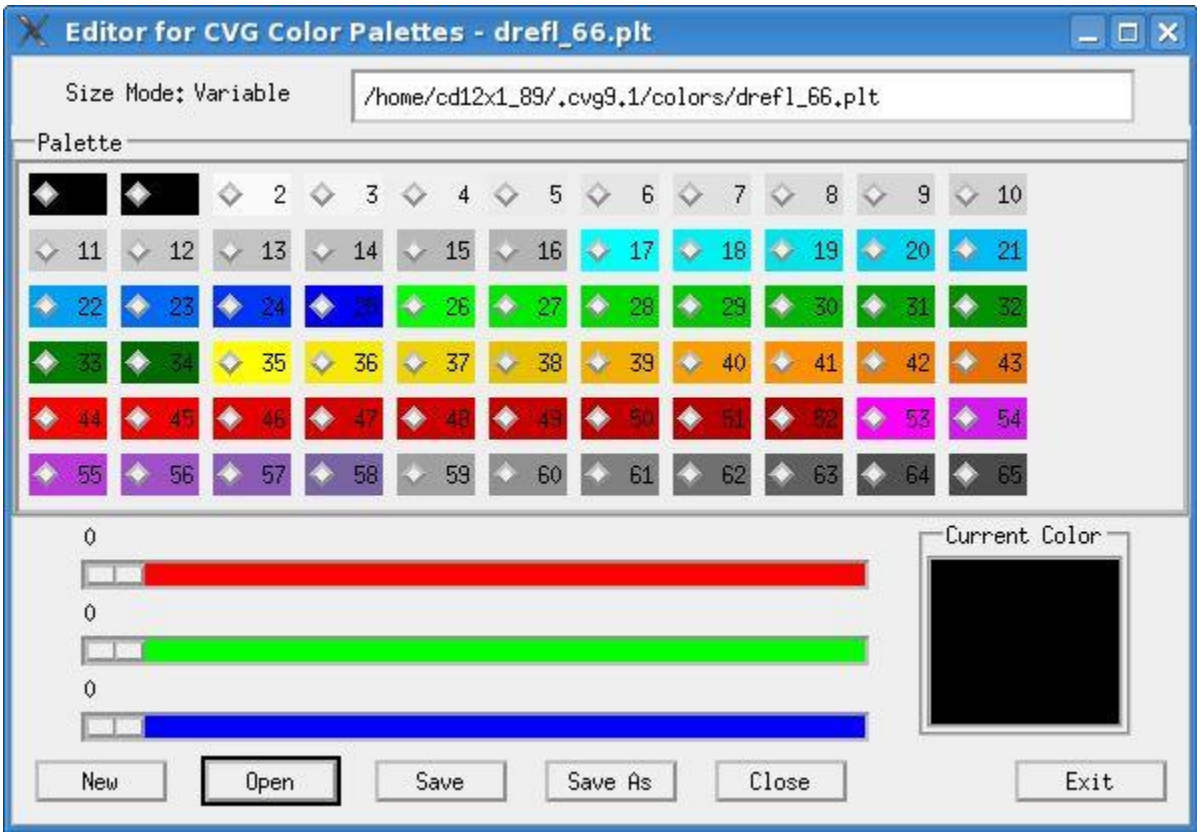




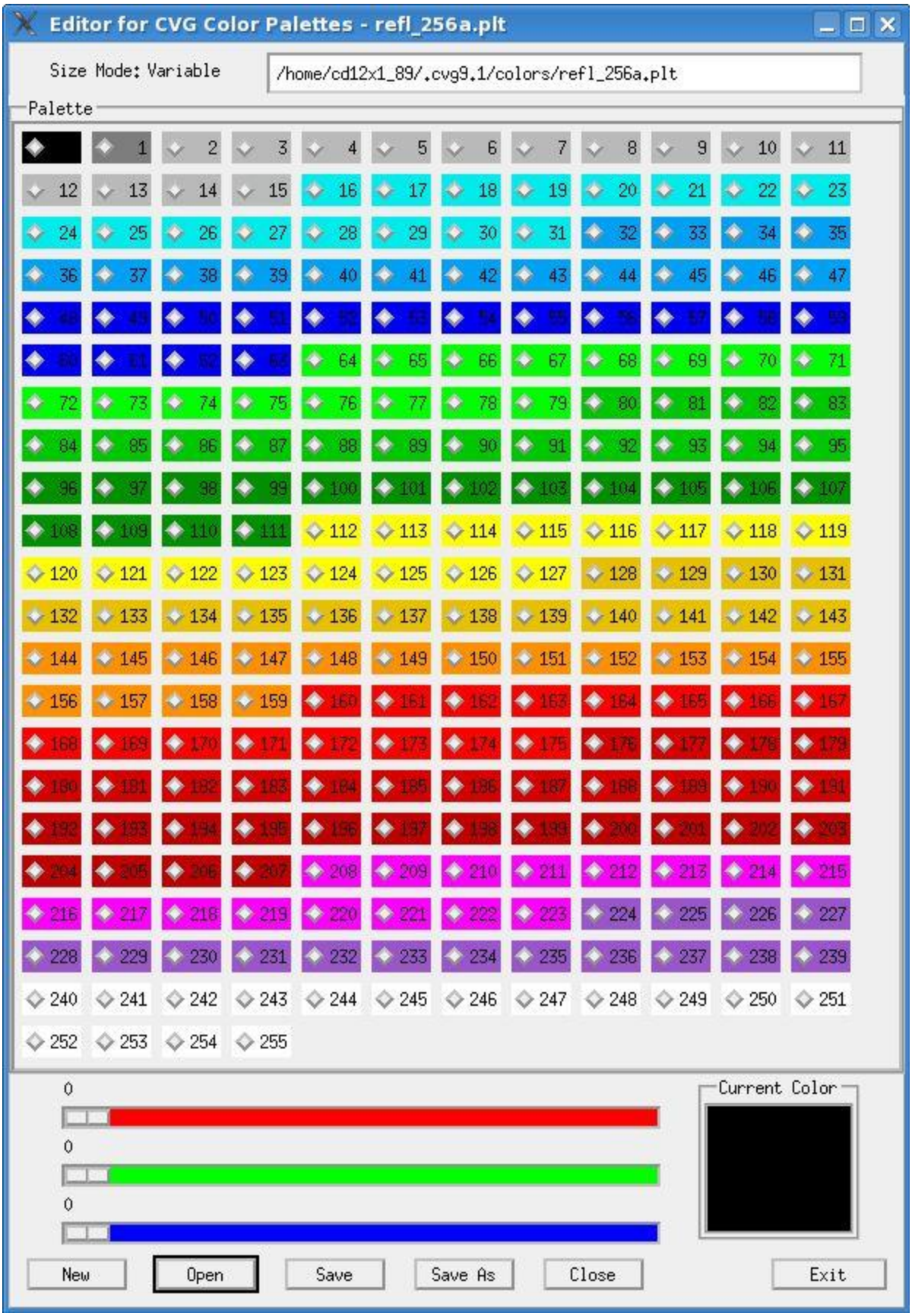
A 64 color reflectivity palette: `nws_64.plt`



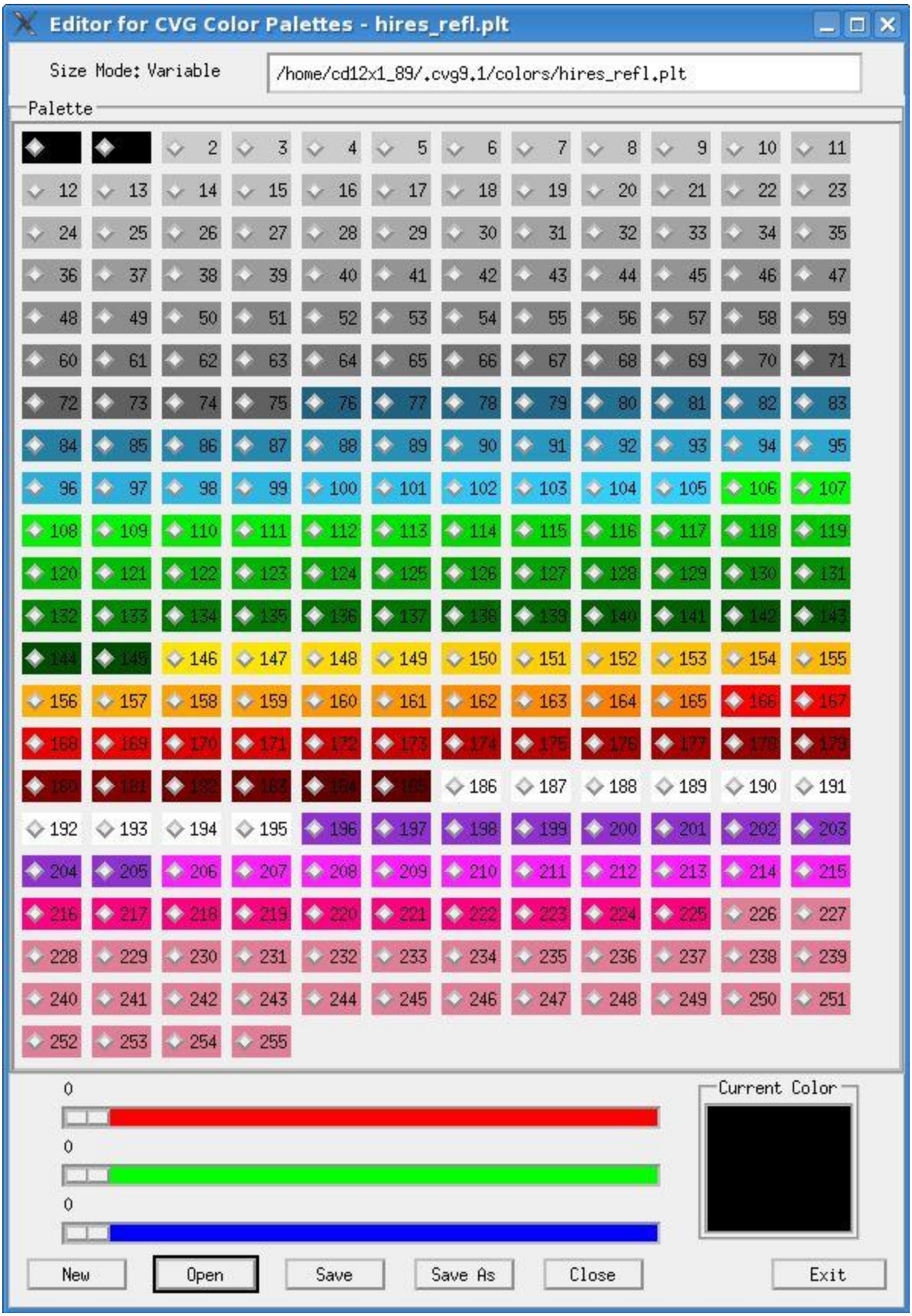
A 66 color reflectivity palette: `nws_66.plt` The colors are compatible with one leading and one trailing flag.



A 66 color reflectivity palette: `drefl_66.plt` The colors are compatible with two leading flags.

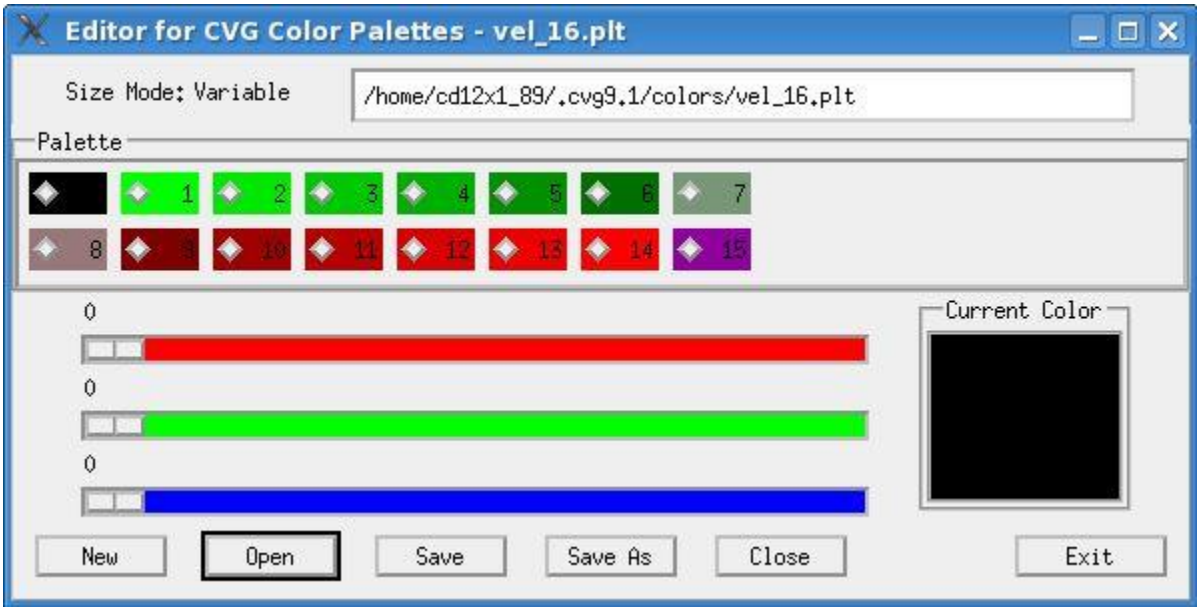


A 256 color reflectivity palette: `refl_256a.plt` The colors are compatible with two leading flags.

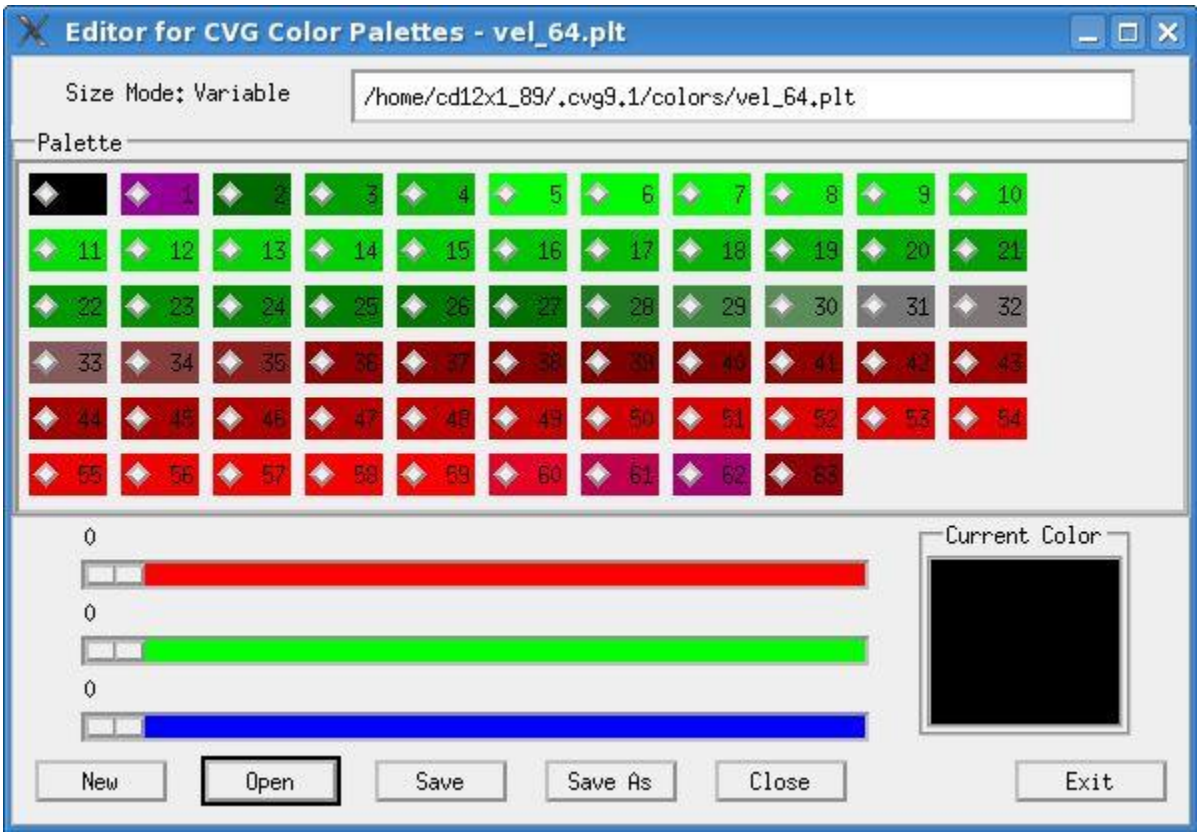


A 256 color reflectivity palette: `hires_refl.plt` Used by AWIPS for hi-res reflectivity (94)

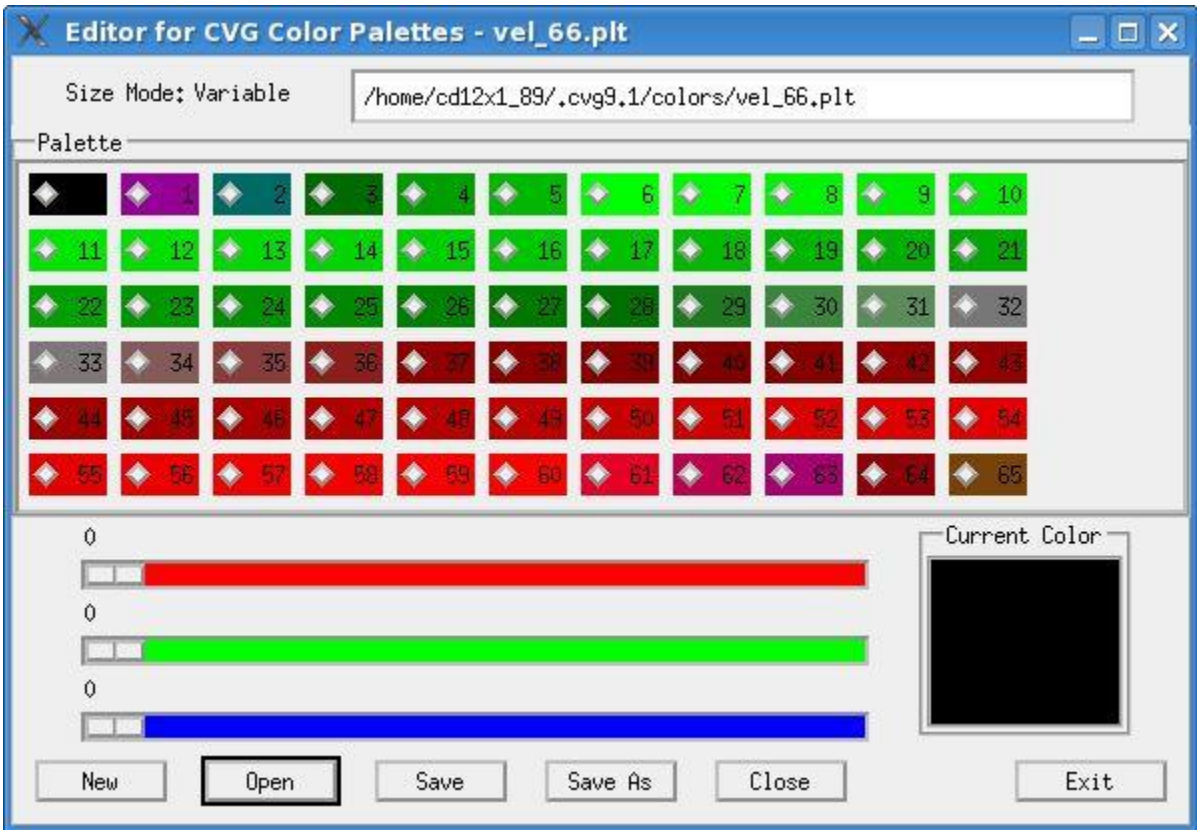
Velocity Related Palettes



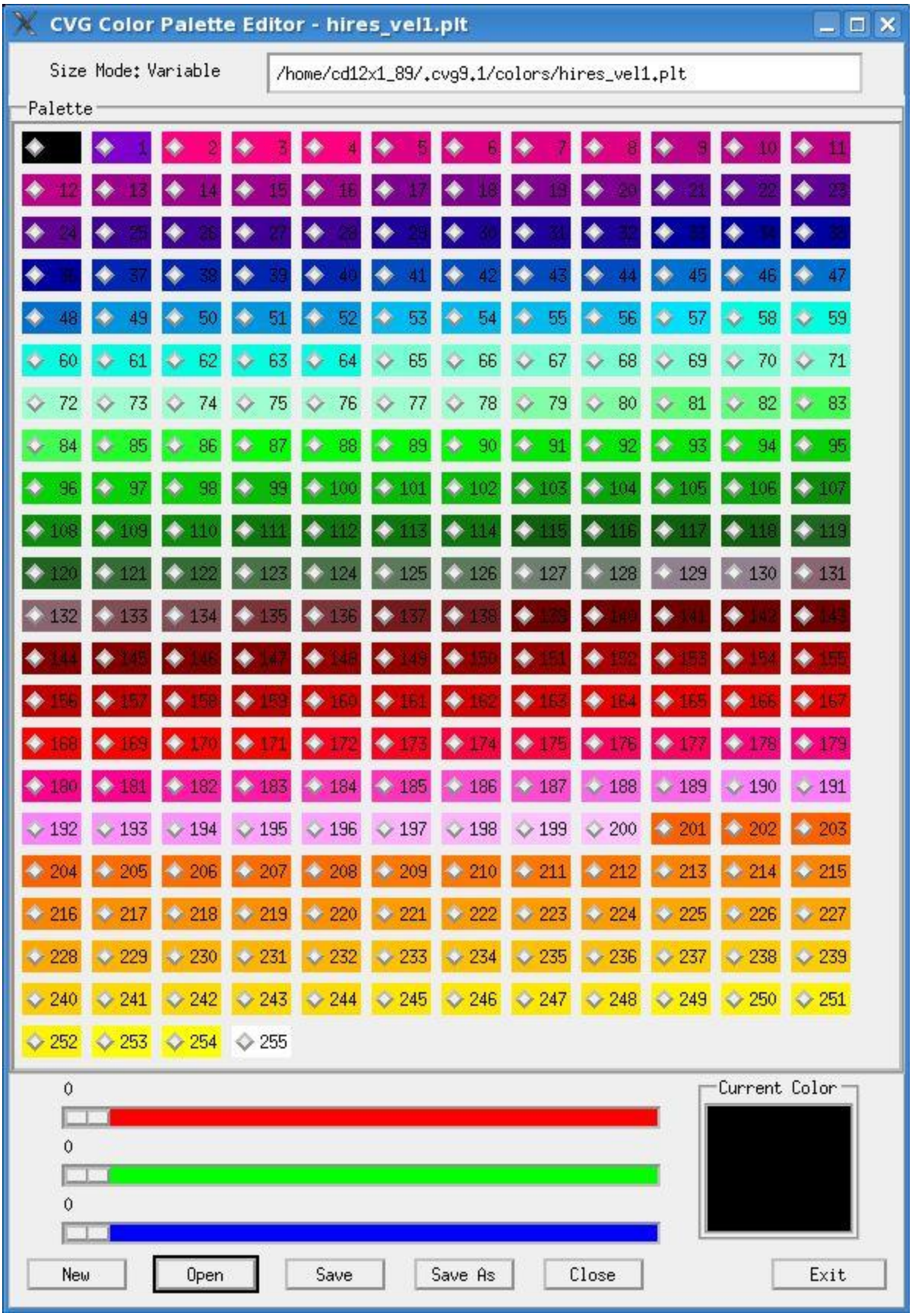
The standard 16 color velocity palette: `vel_16.plt`



A 64 color velocity palette: `vel_64.plt`

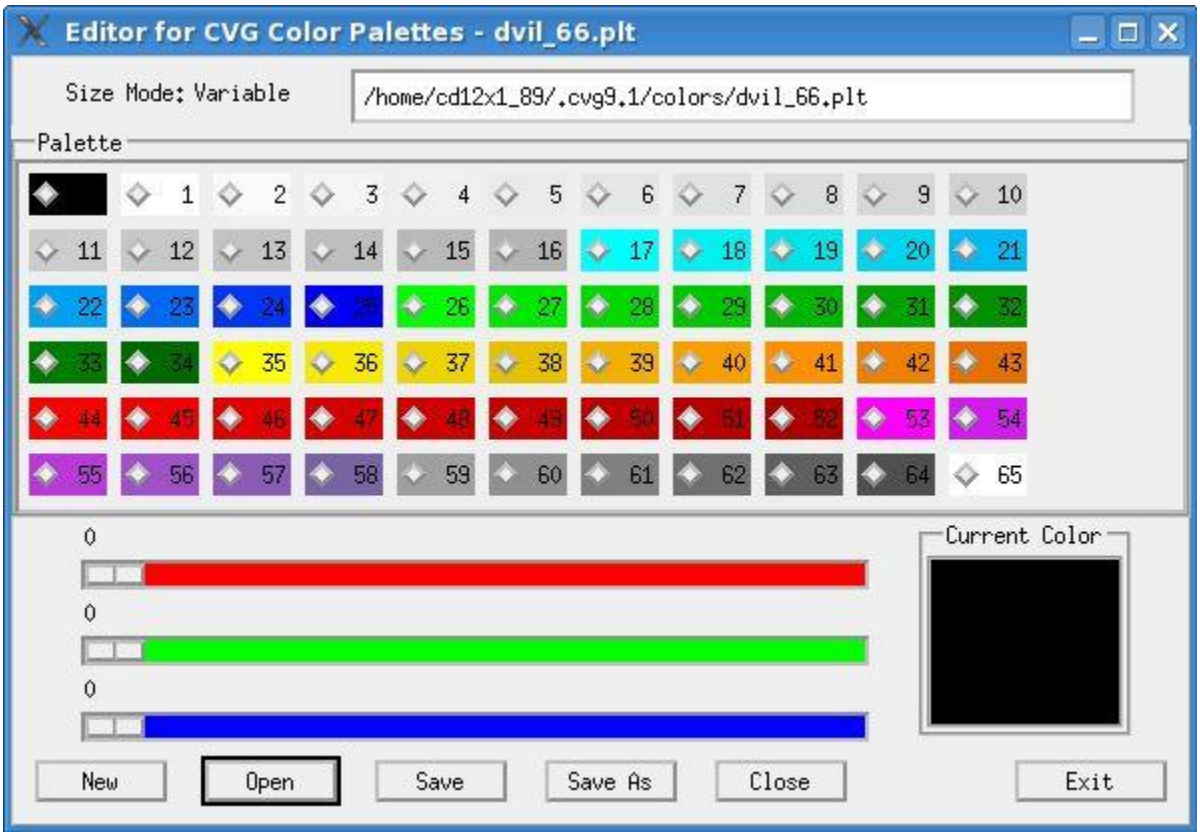


A 66 color velocity palette: `vel_66.plt` The colors are compatible with two leading flags.

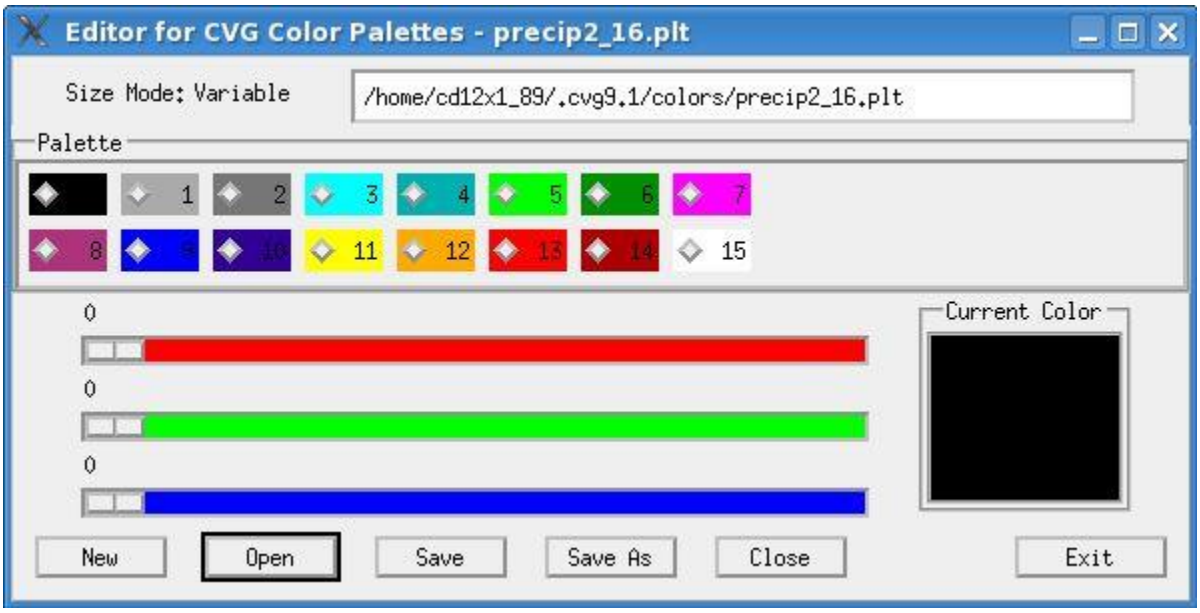


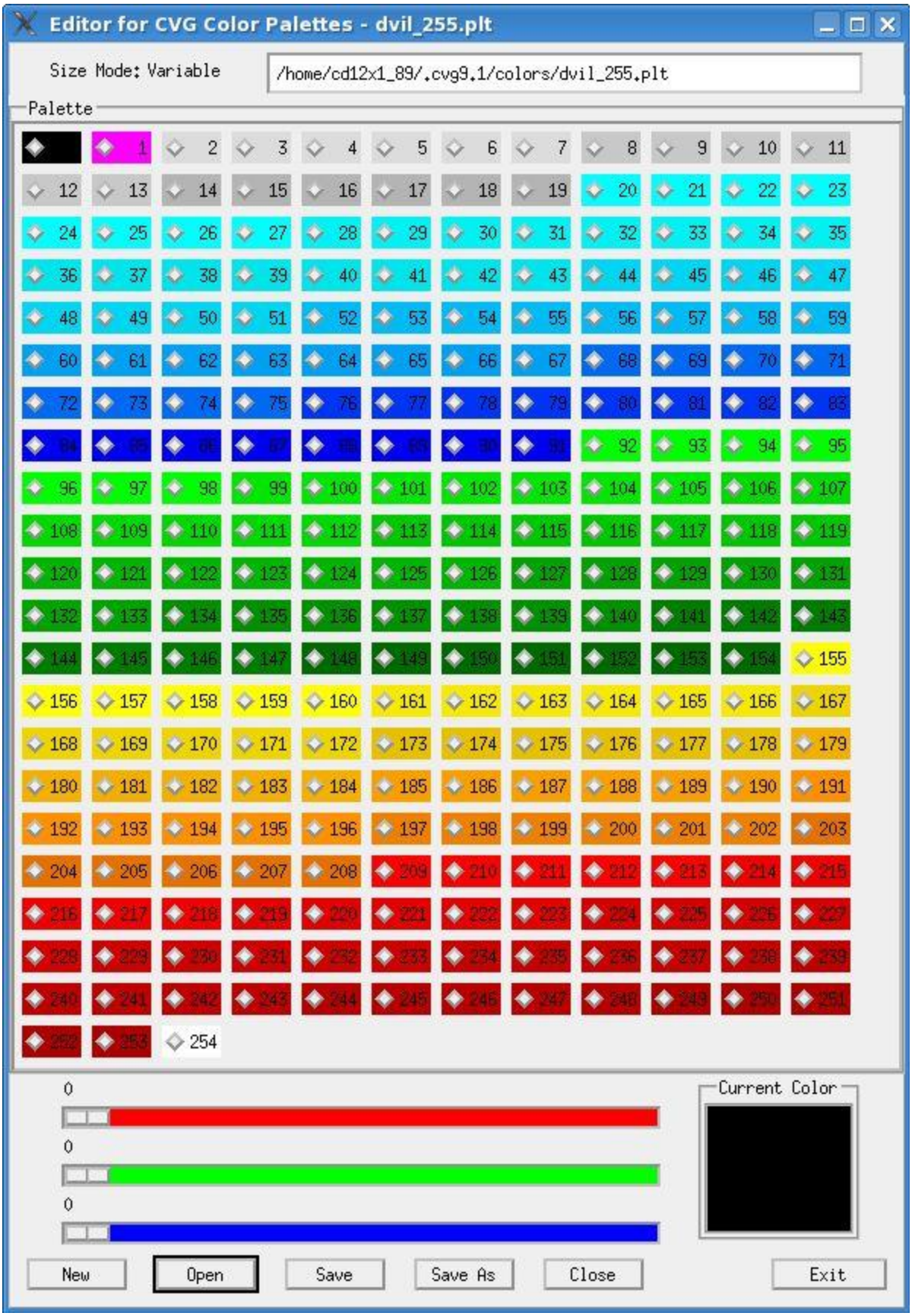
A 256 color velocity palette: `hires_vel1.plt` Used by AWIPS for hi-res velocity (99)

Special Purpose Palettes



Sample 66 color palette for HR VIL:





Sample 255 color palette for HR VIL: `dvil_255.plt` The colors are compatible with 2 leading flags and optionally 1 trailing flag.

Future versions of this document will include some of the dual polarization palettes when the design is finalized.