3. Data Formats

This chapter describes the archived data formats used by IRIS. These formats include data storage on disk files, and output tape formats. Within this chapter, all sizes and addresses are given in bytes. Also included are tables of antenna, signal processor, and general-purpose constants.

3.1 Scalar Definitions

The file **`${IRIS_ROOT}/include/sigtypes.h`** defines a few scalar data types that can be used across platforms. These types are listed in Table 3–1. They are used by the IRIS library routines for defining return values and routine arguments.

Table 3–1: IRIS Data Types

Type	Description
SINT1	Signed character
UINT1	Unsigned character
SINT2	Signed short integer
UINT2	Unsigned short integer
SINT4	Signed longword integer
UINT4	Unsigned longword integer
FLT4	Floating-point number
FLT8	Double-precision floating-point number
BIN2	Unsigned short integer
BIN4	Unsigned longword integer
MESSAGE	Unsigned longword integer

3.2 Structure Definitions

Structures are described in alphabetical order. Tables give the following information about each structure:

- Source — The name of the include file containing the C structure definition. The source files can be found in the **`${IRIS_ROOT}/include`** directories.
- Byte — The byte offset of the structure element.
- Size — The data type of the element from the above table. For structures and spare space the size in bytes is displayed here. Arrays are designated by a data type followed by a number in square brackets.

- Contents — A brief description of the structure element. Structure elements which are themselves structures are identified with angle brackets such as "<ymds_time>". Note, however, that "<spare>" indicates reserved spare space, not a special structure.

3.2.1 cappi_psi_struct Structure

Source: **headers.h**

Byte	Size	Contents
0	4	Shear flags, same as shear_psi_struct
4	SINT4	Height of CAPPI (cm. above reference)
8	UINT2	Flags: bit0=make pseudo CAPPI bit1=Velocity is horizontal winds
10	BIN2	Azimuth smoothing for shear
12	char[12]	VVP name to use for shear correction
24	UINT4	Max age for vvp shear correction in seconds

3.2.2 catch_psi_struct Structure

Source: **headers.h**

Byte	Size	Contents
0	UINT4	Flags in low 16-bits; From 8.09: RAIN1 flags in upper 16-bits
4	UINT4	Hours of accumulation
8	SINT4	Threshold offset in 1/1000 or mm
12	SINT4	Threshold faction in 1/1000
16	char[12]	Name of RAIN1 product to use
28	char[16]	Name of catchment file to use
44	UINT4	From 8.09: Seconds of accumulation in low 16-bits
48	UINT4	From 8.09: RAIN1 min Z
52	UINT4	From 8.09: RAIN1 span in seconds
56	UINT4	From 8.09: Average Gage correction factor in low 16-bits

3.2.3 catch_results Structure

Source: **product.h**

Byte	Size	Contents
0	char[16]	Name of catchment area, null terminated
16	UINT4	Number of catchment area
20	BIN4	Latitude of label
24	BIN4	Longitude of label
28	SINT4	Area of catchment in 1/100 of square km
32	SINT4	Number of pixels in the catchment area

36	SINT4	Number of pixels scanned in the catchment area
40	UINT4	Flags
44	UINT2	Rainfall accumulation in catchment, DB_FLIQUID2 format
46	54	<spare>
100	UINT2[96]	Rainfall accumulation for each input, DB_FLIQUID2 format

3.2.4 color_scale_def Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	**iflags**: Bit 8=COLOR_SCALE_VARIABLE Bit 10=COLOR_SCALE_TOP_SAT Bit 11=COLOR_SCALE_BOT_SAT
4	SINT4	**istart**: Starting level
8	SINT4	**istep**: Level step
12	SINT2	**icolcnt**: Number of colors in scale
14	UINT2	**iset_and_scale**: Color set number in low byte, color scale number in high byte.
16	UINT2[16]	**ilevel_seams**: Variable level starting values

This structure appears at the end of the product_configuration structure. It holds information configured into the product about how to quantize the data into color levels for display. We may remove this structure in the near future because IRIS can override all this at display time. This is only used if the user selects "Use Default Scale" in the Color Scale Tool.

There are two was of specifying the color scale: A fixed scale has uniformly spaced numbers fully specified by a start and step value. A variable scale is specified by a set of 16 individually controlled data seams. The **iflags** bit COLOR_SCALE_VARIABLE indicates which mode is in effect. In both cases, the number of colors can be anywhere from 2 to 16, and optionally zoomed or split to effectively give up to 32 colors. Other bits in **iflags** specify what to do with data off the end of the scale. There are two choices, either saturate at the end (used the last color), or threshold (do not display). For example a typical treatment of reflectivity would be to saturate the high end of the scale (so as not to lose the strong features), and threshold at the low end (to inhibit display of very weak speckles).

The number of main colors and labels in the legend is set by **icolcnt**. To select which subset of colors is used for the scale, fill in the low byte of **iset_and_scale**. The upper byte is not used in the product header.

For fixed spacing, set the **istart** and **istep** to the start and step values desired. These values are integers which are 10 times the physical units as discussed in the **color_setup** utility chapter of the *IRIS Utilities Manual*. The labels of **istart**, istart+istep, istart+2*istep, etc. will show up on the right side of the IRIS legend.

There are two special cases: Width assumes that the starting value is zero, and velocity assumes that the middle of the scale is at zero. If the velocity step is set to zero it means automatically scale to fit the Nyquist velocity.

For variable spacing, the numbers in the array **ilevel_seams** control the starting values for each of the colors. The first seam is the bottom of the first color. The top of the 16th color is computed by adding the step between the 15th and 16th seam to the 16th seam. These seams are data values, using the 16-bit versions of the data. For shear and height data, for which there is no 16-bit version, the 8-bit version is used.

3.2.5 cross_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	BIN2	Azimuth angle of line from left to right
2	10	<spare>
12	SINT4	East coordinate of center (in cm. relative to radar)
16	SINT4	North coordinate of center (in cm. relative to radar)
20	SINT4[15]	User miscellaneous

3.2.6 dsp_data_mask Structure

Source: **sigtypes`.h`**

Byte	Size	Contents	
0	UINT4	Mask word 0	
4	UINT4	Extended header type	
8	UINT4	Mask word 1	Contains bits set for all data recorded.
12	UINT4	Mask word 2	See parameter DB_* in Table 3–6 for
16	UINT4	Mask word 3	bit specification.
20	UINT4	Mask word 4	

3.2.7 extended_header_v0 Structure

Source: **`ingest.h`**

Byte	Size	Contents
0	SINT4	Time in milliseconds from the sweep starting time
4	SINT2	Calibration Signal level
6	14	<spare>

3.2.8 extended_header_v1 Structure

Source: **`ingest.h`**

Byte	Size	Contents
0	SINT4	Time in milliseconds from the sweep starting time
4	SINT2	Calibration Signal level
6	BIN2	Azimuth (binary angle)

8	BIN2	Elevation (binary angle)
10	BIN2	Train Order (binary angle)
12	BIN2	Elevation Order (binary angle)
14	BIN2	Pitch (binary angle)
16	BIN2	Roll (binary angle)
18	BIN2	Heading (binary angle)
20	BIN2	Azimuth Rate (binary angle/second)
22	BIN2	Elevation Rate (binary angle/second)
24	BIN2	Pitch Rate (binary angle/second)
26	BIN2	Roll Rate (binary angle/second)
28	BIN4	Latitude (binary angle)
32	BIN4	Longitude (binary angle)
36	BIN2	Heading Rate (binary angle/second)
38	SINT2	Altitude (meters above MSL)
40	SINT2	Velocity East (cm/second)
42	SINT2	Velocity North (cm/second)
44	SINT4	Time since last update (milliseconds)
48	SINT2	Velocity Up (cm/second)
50	UINT2	Navigation System OK flag
52	SINT2	Radial velocity correction (same units as velocities)

3.2.9 fcast_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Correlation threshold, 0–100
4	SINT4	Data threshold
8	SINT4	Mean speed (cm/hour), zero if none
12	BIN4	Direction of mean speed
16	UINT4	Maximum time between inputs (seconds)
20	SINT4	Maximum allowable velocity (mm/sec)
24	UINT4	Flags
28	SINT4	Desired output resolution (cm)
32	UINT4	Type of input product
36	char[12]	Name of input product (space padded)

3.2.10 gage_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Time span of the rain gage data in seconds.
4	UINT4	Flags: Bit0: File has distro's, Bit1: File does not have gages.

3.2.11 gage_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	char[16]	Name of raingage, null terminated
16	BIN4	Latitude of raingage
20	BIN4	Longitude of raingage
24	UINT2	Average rainrate in DB_RAINRATE2 format for the time span
26	UINT2	Correction factor last calculated in DB_CDBZ2 format
28	UINT1	Data Quality, range 0—10
29	UINT1	Flag bits: 0 = Skip this gage for correction calculation 1 = Gage has Z/R numbers 2 = Gage does not have rainrate
30	UINT2	Z/R constant in 1/10
32	UINT2	Z/R exponent in 1/1000
34	6	<spare>

3.2.12 ingest_configuration Structure

Source: **`ingest.h`**

Byte	Size	Contents
0	char[80]	Name of file on disk
80	SINT2	Number of associated data files extant
82	2	<spare>
84	SINT4	Total size of all files in bytes
88	12	<ymds_time> Time that volume scan was started, TZ spec in bytes 166 & 224
100	12	<spare>
112	SINT2	Number of bytes in the ray headers
114	SINT2	Number of bytes in extended ray headers (includes normal ray header)
116	SINT2	Number of task configuration table
118	SINT2	Playback version number
120	4	<spare>
124	char[8]	IRIS version, null terminated
132	char[16]	Hardware name of site
148	SINT2	Time zone of local standard time, minutes west of GMT
150	char[16]	Name of site, from setup utility
166	SINT2	Time zone of recorded standard time, minutes west of GMT
168	BIN4	Latitude of radar (binary angle: 20000000 hex is 45° North)
172	BIN4	Longitude of radar (binary angle: 20000000 hex is 45° East)
176	SINT2	Height of ground at site (meters above sea level)
178	SINT2	Height of radar above ground (meters)
180	UINT2	Resolution specified in number of rays in a 360° sweep

182	UINT2	Index of first ray from above set of rays
184	UINT2	Number of rays in a sweep
186	SINT2	Number of bytes in each gparam
188	SINT4	Altitude of radar (cm above sea level)
192	SINT4[3]	Velocity of radar platform (cm/sec)(east, north, up)
204	SINT4[3]	Antenna offset from INU (cm) (starboard, bow, up)
216	UINT4	Fault status at the time the task was started, bits: 0:Normal BITE 1:Critical BITE 2:Normal RCP 3:Critical RCP 4:Critical system 5:Product gen. 6:Output
220	SINT2	Height of melting layer (meters above sea level) MSB is complemented, zero=Unknown
222	2	<spare>
224	char[8]	Local timezone string, null terminated
232	248	<spare>

3.2.13 ingest_data_header Structure

Source: **`ingest.h`**

Byte	Size	Contents
0	12	<structure_header> The size stored in here is the total size of the file. That is 76 + 4*Irtotl + Iwritn*raysize.
12	12	<ymds_time> (3.2.71) Date and time that sweep was started TZ specified in ingest_configuration
24	SINT2	Sweep number, origin 1
26	SINT2	Resolution specified in number of rays in a 360° sweep
28	SINT2	Index of first ray from above set of rays
30	SINT2	"Irtotl" Number of rays (and pointers) expected in the file
32	SINT2	"Iwritn" Number of rays actually in the file
34	BIN2	Fixed angle for this sweep (binary angle)
36	SINT2	Number of bits per bin for this data type
38	UINT2	Data type in the file: 0=extended header 1=Total power (dBZ) 2=Reflectivity (dBZ) 3=Velocity 4=Width 5=ZDR
40	36	<spare>

3.2.14 ingest_header Structure

A file containing the ingest_header structure is written for each volume scan. This file is updated each time a sweep has finished. This allows product generators to get data from the ingest files as soon as a sweep is completed. Note on RVP6 and 7 the GPARM is read just after the processor is configured. It will contain configuration fault bits, but not the results of a noise sample taken before that task. On the RVP8 starting in release 8.09 it is read with the first valid ray of the task.

Source: **`ingest.h`**

<structure_header> 12 Bytes
<ingest_configuration> 480 Bytes
<task_configuration> 2612 Bytes
Spare 732 Bytes
GPARM from DSP 128 Bytes
Reserved 920 Bytes

3.2.15 max_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	4	<spare>
4	SINT4	Bottom of interval in cm
8	SINT4	Top of interval in cm
12	SINT4	Number of pixels in side panels
16	SINT2	Horizontal smoother in side panels
18	SINT2	Vertical smoother in side panels

3.2.16 ndop_input Structure

Source: **`headers.h`**

Byte	Size	Contents
0	char[12]	Task name
12	char[3]	Site code
15	UINT1	Flags

3.2.17 ndop_psi_struct Structure

Source: **headers.h**

Byte	Size	Contents
0	3*16	<ndop_input>
48	SINT4	Time window
52	SINT4	Cappi height (cm above reference)
56	SINT4	Output resolution (cm)
60	BIN4	Minimum permitted crossing angle
64	UINT4	Flags: Bit 0=Make diagnostic products

3.2.18 ndop_results Structure

Used for both the NDOP and FCAST products. The change rate is always zero in NDOP products. The SQI is unused in FCAST products. The SQI is a function of the variance of the calculated velocity normalized such that random vectors at half the Nyquist velocity would produce an SQI of zero.

$$SQI = 1 - \left[\sqrt{\frac{VarianceEast + VarianceNorth}{2}} \right] / Vnorm$$

Source: **product.h**

Byte	Size	Contents
0	UINT2	Velocity East in cm/second (DB_VEL2 format)
2	UINT2	Velocity North in cm/second
4	UINT2	Change rate in db/min (DB_CDBZ2 format)
6	UINT1	Signal Quality Index * 256
7	5	<spare>

3.2.19 one_protected_region Structure

Source: **setup.h**

Byte	Size	Contents
0	SINT4	East center from radar in cm
4	SINT4	North center from radar in cm
8	SINT4	East-West size in cm
12	SINT4	North-South size in cm
16	UINT2	Orientation angle in binary angle
18	2	<spare>
20	char[12]	Name of the region, all spaces means unused

3.2.20 ppi_psi_struct Structure

Source: **headers.h**

Byte	Size	Contents
0	BIN2	Elevation angle

3.2.21 product_configuration Structure

Source: **`product.h`**

Byte	Size	Contents
0	12	<Structure_Header>
12	UINT2	Product type code: 1:PPI 2:RHI 3:CAPPI 4:CROSS 5:TOPS 6:TRACK 7:RAIN1 8:RAINN 9:VVP 10:VIL 11:SHEAR 12:WARN 13:CATCH 14:RTI 15:RAW 16:MAX 17:USER 18:USERV 19:OTHER 20:STATUS 21:SLINE 22:WIND 23:BEAM 24:TEXT 25:FCAST 26:NDOP 27:IMAGE 28:COMP 29:TDWR 30:GAGE 31:DWELL 32:SRI 33:BASE 34:HMAX
14	UINT2	Scheduling code: 0:hold; 1:next; 2:all
16	SINT4	Number of seconds to skip between runs
20	12	<ymds_time> (3.2.71) Time product was generated (UTC)
32	12	<ymds_time> (3.2.71) Time of input ingest sweep (TZ flex)
44	12	<ymds_time> (3.2.71) Time of input ingest file (TZ flexible)
56	6	<spare>
62	char[12]	Name of the product configuration file
74	char[12]	Name of the task used to generate the data
86	UINT2	Flag word: (Bits 0,2,3,4,8,9,10 used internally) Bit1: TDWR style messages Bit5: Keep this file Bit6: This is a clutter map Bit7: Speak warning messages Bit11: This product has been composited Bit12: This product has been dwelled Bit13: Z/R source0, 0:Type–in; 1:Setup; 2:Disdrometer Bit14: Z/R source1
88	SINT4	X scale in cm/pixel
92	SINT4	Y scale in cm/pixel
96	SINT4	Z scale in cm/pixel
100	SINT4	X direction size of data array
104	SINT4	Y direction size of data array
108	SINT4	Z direction size of data array
112	SINT4	X location of radar in data array (signed 1/1000 of pixels)
116	SINT4	Y location of radar in data array (signed 1/1000 of pixels)
120	SINT4	Z location of radar in data array (signed 1/1000 of pixels)
124	SINT4	Maximum range in cm (used only in version 2.0, raw products)
128	2	<spare>
130	UINT2	Data type generated (See Section 3.8 for values)

132	char[12]	Name of projection used
144	UINT2	Data type used as input (See Section 3.8 for values)
146	UINT1	Projection type: 0=Centered Azimuthal, 1=Mercator
147	1	<spare>
148	SINT2	Radial smoother in 1/100 of km
150	SINT2	Number of times this product configuration has run
152	SINT4	Z/R relationship constant in 1/1000
156	SINT4	Z/R relationship exponent in 1/1000
160	SINT2	X-direction smoother in 1/100 of km
162	SINT2	Y-direction smoother in 1/100 of km
164	80	<product_specific_info>
244	28	<spare>
272	48	<color_scale_def>(3.2.4) Color scale definition May be removed in the future.

3.2.22 product_end Structure

Source: **`product.h`**

Byte	Size	Contents
0	char[16]	Site name — where product was made (space padded)
16	char[8]	IRIS version where product was made (null terminated)
24	char[8]	IRIS version where ingest data came from
32	12	<ymds_time> Time of oldest input ingest file (only RAIN1 and RAINN, TZ flexible)
44	28	<spare>
72	SINT2	Number of minutes local standard time is west of GMT
74	char[16]	Hardware name where ingest data came from (space padded)
90	char[16]	Site name where ingest data came from (space padded)
106	SINT2	Number of minutes recorded standard time is west of GMT
108	BIN4	Latitude of center (binary angle) *
112	BIN4	Longitude of center (binary angle) *
116	SINT2	Signed ground height in meters relative to sea level
118	SINT2	Height of radar above the ground in meters
120	SINT4	PRF in hertz
124	SINT4	Pulse width in 1/100 of microseconds
128	UINT2	Type of signal processor used
130	UINT2	Trigger rate scheme
132	SINT2	Number of samples used
134	char[12]	Clutter filter file name
146	UINT2	Number of linear based filter for the first bin
148	SINT4	Wavelength in 1/100 of centimeters
152	SINT4	Truncation height (cm above the radar)
156	SINT4	Range of the first bin in cm
160	SINT4	Range of the last bin in cm

164	SINT4	Number of output bins
168	UINT2	Flag word Bit0: Disdrometer failed, we used setup for Z/R source instead
170	SINT2	Number of ingest or product files used to make this product (only on RAIN1 and RAINN)
172	UINT2	Type of polarization used
174	54	<spare>
228	UINT4	Fault status of task, see ingest_configuration 3.2.12 for details
232	UINT4	Mask of input sites used in a composite
236	UINT2	Number of log based filter for the first bin
238	UINT2	Nonzero if cluttermap applied to the ingest data
240	BIN4	Latitude of projection reference *
244	BIN4	Longitude of projection reference *
248	SINT2	Product sequence number
250	SINT2[16]	Color numbers for the up to 16 steps (picture only)
282	2	<spare>
284	SINT2	Height of radar above reference height in meters
286	SINT2	Number of elements in product results array
288	UINT1	Mean wind speed
289	BIN1	Mean wind direction (unknown if speed and direction 0)
290	2	<spare>
292	char[8]	TZ Name of recorded data
300	8	<spare>

* Note on Latitude and longitudes: Interpretation varies with product type. They are as documented for CAPPI, FCAST, MAX, NDOP, PPI, RAIN1, RAINN, SHEAR, SLINE, TOPS, TRACK, VIL, USER and WARN. For all other products, the Center location is the radar location, and the reference location is zero.

3.2.23 product_hdr Structure

Source: **`product.h`**

<structure_header> (3.2.44) 12 Bytes
<product_configuration> (3.2.21) 320 Bytes
<product_end> (3.2.22) 308 Bytes
Data Size Varies

Starting with version 7.31, IRIS allows data to be recorded in UTC on computers with a timezone set to local time. IRIS also records timezone information about the local computer to support optional displaying of different times at output time. Table

3–2 documents the way this information is stored in the product and ingest headers. For backwards compatibility, **iMinutesWest** (byte 106 in product_end) is the difference between the recorded standard time and UTC. All other field are new in version 7.31. **iLocalWest** (byte 72) is the difference between local standard time and UTC. **sTZName** is the text name for the local timezone at the radar. There are also 3 new bits recorded in the upper bits of the milliseconds field of the milliseconds field of all ymds_time structures: **UTC** means that the time is recorded in UTC. **DST** means that the time is in summer time. **LDST** means that the local time on the computer is in summer time. A 60 minute adjustment to the times may be required to correctly deal with summer times. Because the **sLocalTZName** field is new, all old data when displayed using local time will display with a timezone offset, such as "–05" for US EST. Note that ascope recorded data before 7.31 did not include the timezone, so all old data will display with "+00" for a timezone, independent of what was recorded.

Previous to 7.31, IRIS acted as if the new setup question was set to record data using local time. Some systems had their timezone set to UTC to make sure data was recorded in UTC. This can now be changed. It is never recommended to record data using summer time.

Table 3–2: IRIS Timezone Recording

System	**Record UTC in setup**	**Record Local in setup**
Computer timezone UTC	iMinutesWest=0 iLocalWest=0 sLocalTZName="UTC" UTC Flag=1 DST Flag=0 LDST Flag=0 Default Display is "UTC"	iMinutesWest=0 iLocalWest=0 sLocalTZName="UTC" UTC Flag=0 DST Flag=0 LDST Flag=0 Default Display is "UTC"
Computer timezone EST	iMinutesWest=0 iLocalWest=300 sLocalTZName="EST" UTC Flag=1 DST Flag=0 LDST Flag=0 Default Display="UTC"	iMinutesWest=300 iLocalWest=300 sLocalTZName="EST" UTC Flag=0 DST Flag=0 LDST Flag=0 Default Display is "EST"
Computer timezone EDT	iMinutesWest=0 iLocalWest=300 sLocalTZName="EDT" UTC Flag=1 DST Flag=0 LDST Flag=1 Default Display is "UTC"	iMinutesWest=300 **DO** iLocalWest=300 **NOT** sLocalTZName="EDT" **RECORD** UTC Flag=0 **SUMMER** DST Flag=1 **TIME!** LDST Flag=1 Default Display is "EDT"

3.2.24 product_specific_info Structure

Source: **`headers.h`**

Byte	Size	Contents
0	10	<cappi_psi_struct> (if CAPPI)
0	44	<catch_psi_struct> (if CATCH)
0	80	<cross_psi_struct> (if XSECT or USERV)
0	48	<fcast_psi_struct> (if FCAST)
0	20	<maximum_psi_struct> (if MAX)
0	2	<ppi_psi_struct> (if PPI)
0	24	<rain_psi_struct> (if RAIN1 or RAINN)
0	20	<raw_psi_struct> (if RAW)
0	2	<rhi_psi_struct> (if RHI)
0	28	<shear_psi_struct> (if SHEAR)
0	60	<sline_psi_struct> (if SLINE)
0	14	<tdwr_psi_struct> (if TDWR)

0	4	<top_psi_struct> (if TOPS)
0	52	<track_psi_struct>seconds (if TRACK)
0	4	<vil_psi_struct> (if VIL)
0	28	<vvp_psi_struct> (if VVP)
0	80	<warn_psi_struct> (if WARN)
0	40	<wind_psi_struct> (if WIND)
0	80	<spare> (if USER, OTHER, TEXT)

3.2.25 protect_setup Structure

Source: **`setup.h`**

Byte	Size	Contents
0	1024	<one_protected_region>[32] Protected region definitions

3.2.26 rain_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Minimum Z to accumulate (in 2-byte Reflectivity Format)
4	UINT2	Average gage correction factor
6	UINT2	Seconds of accumulation (Starting at 8.09)
8	UINT2	Flag word: Bit0: Apply clutter map Bit1: Apply gage correction Bit2: Clutter map was applied Bit3: Gage correction was applied
10	SINT2	Number of hours to accumulate (RAINN only)
12	char[12]	Name of input product to use (space padded)
24	UINT4	From 8.09: Span in seconds of the input files

3.2.27 raw_prod_bhdr Structure

Source: **`product.h`**

Byte	Size	Contents
0	SINT2	Record number within the file (origin 0: record 3 contains a 2)
2	SINT2	Sweep number (1 is first sweep)
4	SINT2	Byte offset of first full ray in this record (–1 if none)
6	SINT2	Ray number within sweep for above pointed to ray
8	UINT2	Flags
10	2	<spare>

3.2.28 raw_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Data type mask word 0
4	SINT4	Range of last bin in cm
8	UINT4	Format conversion flag: 0=Preserve all ingest data 1=Convert 8-bit data to 16-bit data 2=Convert 16-bit data to 8-bit data
12	UINT4	Flag word: Bit 0=Separate product files by sweep Bit 1=Mask data by supplied mask
16	SINT4	Sweep number if separate files, origin 1
20	4	Xhdr type (unused)
24	4	Data type mask 1
28	4	Data type mask 2
32	4	Data type mask 3
36	4	Data type mask 4
40	4	Playback version (low 16-bits)

3.2.29 ray_header Structure

Source: **`ingest.h`**

Byte	Size	Contents
0	BIN2	Azimuth at beginning of ray (binary angle) If dual-PRF: bit0=ray's PRF was high
2	BIN2	Elevation at beginning of ray (binary angle) If trigger blanking on: bit0=Trigger was not blanked
4	BIN2	Azimuth at end of ray (binary angle)
6	BIN2	Elevation at end of ray (binary angle)
8	SINT2	Actual number of bins in the ray
10	UINT2	Time in seconds from start of sweep (unsigned)

3.2.30 rhi_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	BIN2	Azimuth angle

3.2.31 rti_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	BIN4	Nominal sweep angle
4	UINT4	Starting time offset from sweep time, ms
8	UINT4	Ending time offset
12	BIN4	Azimuth of the first ray in the file
16	BIN4	Elevation of the first ray in the file

3.2.32 shear_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	BIN4	Azimuthal smoothing angle
4	BIN2	Elevation angle
6	2	<spare>
8	UINT4	Flag word: Bit0=Do radial shear Bit1=Do azimuthal shear Bit2=Do mean wind correction to azimuthal shear using VVP Bit3=Mean wind correction to azimuthal shear done Bit4=Unfolding done in associated VVP product Bit5=Do elevation shear
12	char[12]	Name of VVP product to use (space padded)
24	UINT4	Maximum age of VVP to use (seconds)

3.2.33 sline_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT4	Area in square meters
4	SINT4	Shear threshold (cm/sec/km)
8	UINT4	Bit flags to choose protected areas
12	SINT4	Maximum forecast time in seconds
16	UINT4	Maximum age between products for motion calculation
20	SINT4	Maximum velocity allowed (mm/sec)
24	UINT4	Flag word: Bit0=Do radial shear Bit1=Do azimuthal shear (both bits mean do combined) Bit2=Do mean wind correction to azimuth shear using VVP Bit3=Mean wind correction to azimuthal shear done Bit4=Unfolding done in associated VVP product

		Bit8=Use two elevation angles
		Bit9=Generate diagnostic output
		Bit10=Max centroid count exceeded
28	BIN4	Azimuthal smoothing angle (0=none)
32	BIN4	Elevation binary angle
36	BIN4	Elevation binary angle
40	char[12]	Name of VVP task
52	UINT4	Maximum age of VVP in seconds
56	SINT4	Curve fit standard deviation threshold in cm.
60	UINT4	Low byte: Min length of sline (unsigned 1/10 km)

3.2.34 sline_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	SINT4	East coordinate of center point, cm
4	SINT4	North coordinate of center point, cm
8	BIN4	Rotation angle to X-Y coordinates of curve fit polynomial Span is –90 to +90 degrees.
12	SINT4	X coordinate of left of curve, cm
16	SINT4	X coordinate of right of curve, cm
20	SINT4[6]	6 polynomial coefficients
44	SINT4	Standard deviation of fit
48	SINT4	Propagation speed (mm/second)
52	BIN4	Propagation direction (binary angle)
56	SINT4	Reference side wind speed (mm/second)
60	BIN4	Reference side wind direction (binary angle)
64	SINT4	Other side wind speed (mm/second)
68	BIN4	Other side wind direction (binary angle)
72	SINT4[32]	ETA in seconds for each protected area (0 if in area, –1 if not expected)
200	UINT4	Flags: Bit0=Propagation speed available Bit1=Wind speeds valid
204	796	<spare>

The polynomial curve fit is calculated as follows: First, threshold a shear product based on the shear threshold specified in the product configuration. Let *East* and *North* refer to the distance of the center of each pixel from the radar position in centimeters. This coordinate system is rotated by an angle θ clockwise about the radar location to produce a new coordinate system with distances also in centimeters. This is the "Rotation angle to X-Y coordinates of curve fit polynomial" above. I use the variables X and Y in this coordinate system. The equations for the transformation are:

$$X = North \sin\theta + East \cos\theta$$
$$Y = North \cos\theta - East \sin\theta$$

Call this transformation T_{rotate}, and the reverse transformation $T^{-1}{}_{rotate}$.

In this coordinate system, I will call the *X*-coordinate of the left most end of the line X_l and similarly the right most end X_r. Next we shift and scale the coordinate system to keep the polynomial coefficients from becoming too large. The equations for the transformation are:

$$X' = \frac{X - X_l}{X_r - X_l}$$
$$Y' = Y$$

Call this transformation T_{scale}, and the reverse transformation $T^{-1}{}_{scale}$.

In this new coordinate system, the polynomial is simple to express:

$$Y' = A_0 + A_1X' + A_2X'^2 + A_3X'^3 + A_4X'^4 + A_5X'^5$$

or

$$Y' = P[X']$$

The standard standard deviation is computed as follows: Let $[X'_i , Y'_i]$ represent the *i*th point in the data set in the rotated and scaled coordinate system, and *N* the total number of points, then the standard deviation is:

$$standard\ \ deviation = \sqrt{\frac{1}{N}\sum_{i = 1, N} [Y'_i - P[X'_i]]^2}$$

The center point is computed as follows: Let $[East_i , North_i]$ represent the *i*th point in the data set in the original coordinate system, and *N* the total number of points, then the center point is:

$$East\ \ center = \frac{1}{N}\sum_{i = 1, N} East_i$$

$$North\ \ center = \frac{1}{N}\sum_{i = 1, N} North_i$$

3.2.35 sri_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Flags Bit 0=Do profile correction Bit 3&4: Melt src: 0=Ingest, 1=Setup, 2=TypeIn Bit 5=Check for convection
4	SINT4	Total number of bins inserted
8	SINT4	Number of bins with data
12	SINT4	Number of data bins profile corrected
16	SINT2	Surface height (m above reference)
18	SINT2	Maximum height (m above reference)
20	SINT2	Melting height (m above MSL)
22	SINT2	Melting level thickness (m)
24	SINT2	Gradient above melting (1/100 dB/km)
26	SINT2	Gradient below melting (1/100 dB/km)
28	SINT2	Convective check height (m above melting)
30	SINT2	Convective check level (DB_DBZ2 format)

3.2.36 status_antenna_info Structure

Source: **`product.h`**

Byte	Size	Contents
0	BIN4	Azimuth position
4	BIN4	Elevation position
8	UINT4	Azimuth velocity
12	UINT4	Elevation velocity
16	UINT4	Command bits
20	UINT4	Command bit availability mask
24	UINT4	Status bits
28	UINT4	Status bit availability mask
32	UINT4	Bite fault flag, 0=OK, 1=Fault, 2=Critical
36	SINT4	Lowest field number generating a fault
40	UINT4	Status bits which can cause critical faults
44	UINT4[3]	Mask indicating the state of each BITE field
56	UINT4[3]	Mask indicating which BITE fields are faulted
68	32	<spare>

3.2.37 status_device_info Structure

Source: **`product.h`**

Byte	Size	Contents
0	40	<status_one_device> dsp
40	40	<status_one_device> antenna
80	720	<status_one_device>[18] output devices

3.2.38 status_message_info Structure

Source: **`product.h`**

Byte	Size	Contents
0	SINT4	Message count
4	MESSAGE	Actual message number
8	SINT4	Number of times it was repeated
12	char[16]	Process name, null terminated
28	char[80]	Text of message, null terminated
108	char[32]	Name of signal, null terminated
140	20	<serv_ymds_time> Time of message (TZ flexible)
160	UINT4	Message type: 1=info, 2=normal, 3=say
164	36	<spare>

3.2.39 status_misc_info Structure

Source: **`product.h`**

Byte	Size	Contents
0	char[16]	Radar status configuration name, null terminated
16	char[16]	Task configuration name, null terminated
32	char[16]	Product scheduler configuration name, null terminated
48	char[16]	Product output configuration name, null terminated
64	char[16]	Active task name, null terminated
80	char[16]	Active product name, null terminated
96	UINT4	Site type (IRIS style)
100	SINT4	Number of incoming network connects
104	SINT4	Number of IRIS clients connected
108	4	<spare>
112	SINT4	Number of output devices
116	UINT4	Flags: bit 0=Automatic mode switching is enabled bit 1=Status product detected a Fault bit 2=Critical message fault
120	char[4]	Node status fault site
124	12	<ymds_time> Time of active task (TZ flexible)
136	64	<spare>

3.2.40 status_one_device Structure

Source: **`product.h`**

Byte	Size	Contents
0	UINT4	Device type
4	SINT4	Unit number
8	UINT4	Status
12	4	<spare>
16	UINT4	Mode from process table
20	char[16]	String (null terminated)
36	4	<spare>

3.2.41 status_one_process Structure

Source: **`product.h`**

Byte	Size	Contents
0	UINT4	Command
4	UINT4	Mode
8	12	<spare>

3.2.42 status_process_info Structure

Source: **`product.h`**

Byte	Size	Contents
0	20	<status_one_process> Ingest process
20	20	<status_one_process> Ingfio process
40	20	<spare>
60	20	<status_one_process> Output master process
80	20	<status_one_process> Product process
100	20	<status_one_process> Watchdog process
120	20	<status_one_process> Reingest process
140	20	<status_one_process> Network process
160	20	<status_one_process> Nordrad process
180	20	<status_one_process> Server process
200	20	<status_one_process> RibBuild process
220	180	<spare>

3.2.43 status_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	200	<status_misc_info> Miscellaneous info
200	400	<status_process_info> Status of processes
600	800	<status_device_info> Status of devices
1400	100	<status_antenna_info> Status of antenna
1500	200	<status_message_info> Message information

3.2.44 structure_header Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT2	Structure identifier: Value; Description 22 Task_configuration 23 Ingest_header 24 Ingest_data_header 25 Tape_inventory 26 Product_configuration 27 Product_hdr 28 Tape_header_record
2	SINT2	Format version number (see headers.h)
4	SINT4	Number of bytes in the entire structure
8	SINT2	Reserved
10	SINT2	Flags: bit0=structure complete

3.2.45 tape_header_record Structure

Source: **`output.h`**

Byte	Size	Contents
0	12	<structure_header>
12	char[16]	Tape identification name
28	char[16]	Name of site that created tape
44	12	<ymds_time> Time that the tape was created (TZ flexible)
56	4	<spare>
60	char[8]	IRIS version when tape was initialized
68	252	<spare>

3.2.46 task_calib_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	SINT2	Reflectivity slope (4096*dB/ A/D count)
2	SINT2	Reflectivity noise threshold (1/16 dB above Noise)
4	SINT2	Clutter Correction threshold (1/16 dB)
6	SINT2	SQI threshold (0–1)*256
8	SINT2	Power threshold (1/16 dBZ)
10	8	<spare>
18	SINT2	Calibration Reflectivity (1/16 dBZ at 1 km)
20	UINT2	Threshold flags for uncorrected reflectivity
22	UINT2	Threshold flags for corrected reflectivity

24	UINT2	Threshold flags for velocity
26	UINT2	Threshold flags for width
28	UINT2	Threshold flags for ZDR
30	6	<spare>
36	UINT2	Flags: Bit 0: Speckle remover for log channel Bit 3: Speckle remover for linear channel Bit 4: Flag to indicate data is range normalized Bit 5: Flag to indicate pulse at beginning of ray Bit 6: Flag to indicate pulse at end of ray Bit 7: Vary number of pulses in dual PRF Bit 8: Use 3 lag processing in PP02 Bit 9: Apply velocity correction for ship motion Bit 10: Vc is unfolded Bit 11: Vc has fallspeed correction Bit 12: Zc has beam blockage correction Bit 13: Zc has attenuation correction
38	4	<spare>
42	SINT2	ZDR bias in signed 1/16 dB
44	276	<spare>

3.2.47 task_configuration Structure

Source: **`iris_task.h`**

<structure_header> 12 Bytes
<task_sched_info> 120 Bytes
<task_dsp_info> 320 Bytes
<task_calib_info> 320 Bytes
<task_range_info> 160 Bytes
<task_scan_info> 320 Bytes
<task_misc_info> 320 Bytes

<task_end_info> 320 Bytes
Comments 720 Bytes

3.2.48 task_dsp_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	Major mode
2	UINT2	DSP type
4	24	<dsp_data_mask> Data type mask
28	UINT4[27]	Auxiliary data definition
136	SINT4	PRF in Hertz
140	SINT4	Pulse width in 1/100 of microseconds
144	UINT2	Multi PRF mode flag: 0=1:1, 1=2:3, 2=3:4, 3=4:5
146	SINT2	Dual PRF delay
148	UINT2	AGC feedback code
150	SINT2	Sample size
152	UINT2	Gain Control flag (0=fixed, 1=STC, 2=AGC)
154	char[12]	Name of file used for clutter filter
166	UINT1	Linear based filter number for first bin
167	UINT1	Log based filter number for first bin
168	SINT2	Attenuation in 1/10 dB applied in fixed gain mode
170	UINT2	Gas attenuation in 1/100000 dB/km for first 10000, then stepping in 1/10000 dB/km
172	UINT2	Flag nonzero means cluttermap used
174	UINT2	XMT phase sequence: 0:Fixed, 1:Random, 3:SZ8/64
176	UINT4	Mask used for to configure the ray header.
180	UINT2	Time series playback flags, see OPTS_* in dsp.h
182	138	<spare>

3.2.49 task_end_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	SINT2	Task major number
2	SINT2	Task minor number
4	char[12]	Name of task configuration file
16	char[80]	Task description
96	SINT4	Number of tasks in hybrid task
100	UINT2	Task state: 0=no task; 1=task being modified; 2=inactive;

		3=scheduled, 4=running.
102	2	<spare>
104	12	<ymds_time> Data time of task (TZ flexible)
116	204	<spare>

3.2.50 task_file_scan_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	First azimuth angle (binary angle)
2	UINT2	First elevation angle (binary angle)
4	char[12]	Filename for antenna control
16	184	<spare>

3.2.51 task_manual_scan_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	Flags: bit 0=Continuous recording
2	198	<spare>

3.2.52 task_misc_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	SINT4	Wavelength in 1/100 of cm
4	char[16]	T/R Serial Number
20	SINT4	Transmit Power in watts
24	UINT2	Flags: Bit 0: Digital signal simulator in use Bit 1: Polarization in use Bit 4: Keep bit
26	UINT2	Type of polarization
28	SINT4	Truncation height (centimeters above the radar)
32	18	<spare for polarization spec>
50	12	<spare>
62	SINT2	Number of bytes of comments entered
64	BIN4	Horizontal beamwidth (starting in 7.18)

Byte	Size	Contents
68	BIN4	Vertical beamwidth (starting in 7.18)
72	UINT4[10]	Customer defined storage (starting in 7.27)
112	208	<spare>

3.2.53 task_ppi_scan_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	Starting azimuth angle (binary angle)
2	UINT2	Ending azimuth angle (binary angle)
4	UINT2[40]	List of elevations (binary angles) to scan at
84	115	<spare>
199	1	Start limit: 0=Nearest, 1=Left, 2=Right

3.2.54 task_range_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	SINT4	Range of first bin in centimeters
4	SINT4	Range of last bin in centimeters
8	SINT2	Number of input bins
10	SINT2	Number of output range bins
12	SINT4	Step between input bins
16	SINT4	Step between output bins (in centimeters)
20	UINT2	Flag for variable range bin spacing (1=var, 0=fixed)
22	SINT2	Range bin averaging flag
24	136	<spare>

3.2.55 task_rhi_scan_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	Starting elevation angle (binary angle)
2	UINT2	Ending elevation angle (binary angle)
4	UINT2[40]	List of azimuths (binary angles) to scan at
84	115	<spare>
199	1	Start limit: 0=Nearest, 1=Lower, 2=Higher

3.2.56 task_scan_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	UINT2	Antenna scan mode 1:PPI sector, 2:RHI, 3:Manual, 4:PPI cont, 5:file
2	SINT2	Desired angular resolution in 1/1000 of degrees
4	2	<spare>
6	SINT2	Number of sweeps to perform
If RHI scan:		
8	200	<task_rhi_scan_info>
If PPI sector, or PPI continuous scan:		
8	200	<task_ppi_scan_info>
If File scan:		
8	200	<task_file_scan_info>
If PPI Manual or RHI Manual scan:		
8	200	<task_manual_scan_info>
In all cases:		
208	112	<spare>

3.2.57 task_sched_info Structure

Source: **`iris_task.h`**

Byte	Size	Contents
0	SINT4	Start time (seconds within a day)
4	SINT4	Stop time (seconds within a day)
8	SINT4	Desired skip time (seconds)
12	SINT4	Time last run (seconds within a day)(0 for passive ingest)
16	SINT4	Time used on last run (seconds) (in file time to writeout)
20	SINT4	Relative day of last run (zero for passive ingest)
24	UINT2	Flag: Bit 0 = ASAP Bit 1 = Mandatory Bit 2 = Late skip Bit 3 = Time used has been measured Bit 4 = Stop after running
26	94	<spare>

3.2.58 tdwr_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	UINT4	Flags, bit0=LLWAS, bit1=WARN, bit2=SLINE
4	UINT4	Maximum range in cm
8	char[4]	Source ID

12	char[3]	Center field wind direction
15	UINT1	<spare>
16	char[2]	Center field wind speed
18	char[2]	Center field gust speed
20	UINT4	Mask of protected areas checked
24	UINT4	Warning count
28	UINT4	SLINE count
32	UINT4	Forecast time

3.2.59 tdwr_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	char[16]	Corridor name, null terminated
16	char[8]	Arena Name, null terminated
24	char[3]	Alert Type
27	1	<spare>
28	char[3]	Threshold direction
31	1	<spare>
32	char[2]	Threshold speed
34	2	<spare>
36	SINT4	Alert speed loss/gain in mm/sec, negative=loss
40	UINT4	Mask of protected areas checked for this corridor
44	UINT4	Mask of protected areas hit in this corridor
48	52	<spare>

3.2.60 text_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	char[512]	null terminated string of arbitrary text to be spoken by IRIS

3.2.61 top_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	4	<spare>
4	SINT2	Z threshold in 1/16 dbZ

3.2.62 track_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT4	Centroid area threshold in square meters
4	SINT4	Threshold level for centroid
8	UINT4	Protected area mask

12	SINT4	Maximum forecast time in seconds
16	UINT4	Maximum age between products for motion calculation
20	SINT4	Maximum motion allowed in mm/sec
24	UINT4	Flag word: Bit9=Generate diagnostic output
28	SINT4	Maximum span of track points in the file (seconds)
32	UINT4	Input product type
36	char[12]	Input product name
48	SINT4	Point connecting error allowance

3.2.63 track_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	BIN4	Latitude (32-bit binary angle)
4	BIN4	Longitude
8	SINT4	Height (cm above reference)
12	UINT4	Flags: bit0=forecast, bit1=manual, bit2=text, bit3=icon
16	SINT4	Area of centroid (1/100 of square km)
20	SINT4	Major axis of equal area ellipse (cm)
24	SINT4	Minor axis of equal area ellipse (cm)
28	BIN4	Orientation angle of ellipse
32	UINT4	Protected area mask of areas hit
36	SINT4	Maximum value of data within area
40	8	<spare>
48	SINT4	Average value of data within area
52	8	<spare>
60	SINT4	Scale factor of input data
64	SINT4	Track index number
68	char[32]	Text
100	12	<ymds_time> Time (TZ flexible)
112	SINT4[32]	ETA in seconds for each protected area (0 if in area, –1 if not expected)
240	UINT4	Data type of input data
244	8	<spare>
252	SINT4	Propagation speed (mm/second)
256	BIN4	Propagation direction (binary angle)
260	UINT4	Text size
264	UINT4	Color
268	32	<spare>

3.2.64 vil_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	4	<spare>
4	SINT4	Height of bottom of interval (cm above reference)
8	SINT4	Height of top of interval (cm above reference)

3.2.65 vvp_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT4	Minimum range to process in cm
4	SINT4	Maximum range to process in cm
8	SINT4	Minimum height to process (cm above reference)
12	SINT4	Maximum height to process (cm above reference)
16	SINT2	Number of intervals to process at
18	2	<spare>
20	SINT4	Quota number of bins per interval
24	SINT4	Wind parameters mask. See bits defined in vvp_results structure below.

3.2.66 vvp_results Structure

Source: **`product.h`**

Byte	Size	Contents	
0	SINT4	Number of data points used	
4	SINT4	Height of the center of interval (cm above reference)	
8	SINT4	Number of reflectivity data points used	
12	8	<spare>	
		Bit	Description
20	SINT2	0	Wind speed in cm/sec
22	SINT2	1	Wind speed standard deviation
24	SINT2	2	Wind direction in 1/10 of degrees
26	SINT2	3	Wind direction standard deviation
28	SINT2	4	Vertical wind speed in cm/sec
30	SINT2	5	Vertical wind speed standard deviation
32	SINT2	6	Horizontal divergence in 10**–7/sec
34	SINT2	7	Horizontal divergence standard deviation
36	SINT2	8	Radial velocity standard deviation
38	SINT2	9	Linear averaged reflectivity (DB_CDBZ2 format)
40	SINT2	10	Log averaged reflectivity standard deviation
42	SINT2	11	Deformation in 10**–7/sec

44	SINT2	12	Deformation standard deviation
46	SINT2	13	Axis of dilatation in 1/10 of degrees
48	SINT2	14	Axis of dilatation standard deviation
50	SINT2	15	Log averaged reflectivity (DB_CDBZ2 format)
52	SINT2	16	Linear averaged reflectivity standard deviation
54	30	<spare>	

3.2.67 warn_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT4	Centroid area threshold in square meters
4	SINT4[3]	Threshold levels (1/100 of user units)
16	SINT2[3]	Data valid times (seconds)
22	2	<spare>
24	char[12]	Symbol to display
36	char[36]	Names of the product files
72	UINT1[3]	Product types
75	char[1]	<spare>
76	UINT4	Protected area bit flag

3.2.68 warning_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	BIN4	Latitude (32-bit binary angle)
4	BIN4	Longitude
8	SINT4	Height (cm above reference)
12	UINT4	Flags: Bit 0: Skip data label
16	SINT4	Area of centroid (1/100 of square km)
20	SINT4	Major axis of equal area ellipse (cm)
24	SINT4	Minor axis of equal area ellipse (cm)
28	BIN4	Orientation angle of ellipse
32	UINT4	Protected area mask of areas hit
36	SINT4[3]	Maximum value of data within the area (1/100 of user units)
48	SINT4[3]	Average value of data within the area (1/100 of user units)
60	SINT4	Scale factor of input data
64	4	<spare>
68	char[16]	Text, null-terminated
84	156	<spare>
240	UINT4[3]	Data type of input data
252	SINT4	Propagation speed (mm/second)
256	BIN4	Propagation direction (binary angle)
260	40	<spare>

3.2.69 wind_psi_struct Structure

Source: **`headers.h`**

Byte	Size	Contents
0	SINT4	Minimum height (cm above reference)
4	SINT4	Maximum height (cm above reference)
8	SINT4	Minimum range in cm
12	SINT4	Maximum range in cm
16	SINT4	Number of point in range
20	SINT4	Number of points in panels
24	SINT4	Sector length in cm
28	BIN4	Sector width in binary angle
32	UINT4	Flag word: bit0=Subtract mean wind
36	UINT4	Wind parameters mask of included VVP

3.2.70 wind_results Structure

Source: **`product.h`**

Byte	Size	Contents
0	SINT4	Number of possible hits based on geometry
4	SINT4	Number of data points actually used
8	SINT4	Range of center of sector
12	BIN2	Azimuth of center of sector
14	SINT2	Velocity East in cm/second
16	SINT2	Velocity East standard deviation in cm/second
18	SINT2	Velocity North in cm/second
20	SINT2	Velocity North standard deviation in cm/second
22	10	<spare>

3.2.71 ymds_time Structure

Source: **`sigtypes.h`**

Byte	Size	Contents
0	SINT4	Seconds since midnight
4	UINT2	Milliseconds in lower 10 bits. Bit 10: Time is daylight savings time Bit 11: Time is UTC Bit 12: Local time is daylight savings time
6	SINT2	Year
8	SINT2	Month
10	SINT2	Day

3.3 Data Types

The following sections document the data formats for both ingest data (stored in the raw product) and other products. The data formats for reflectivity, velocity, width, and ZDR differ slightly in the two cases. When these data types are converted to a Cartesian product, the data values of 255 are replaced with 254, and 255 takes on the meaning of area not scanned. Polar data does not have this meaning. The name in parenthesis after each section title is the data type name used by IRIS to identify the data. Parameters for these numbers are defined in the dsp_lib.h file.

3.3.1 Extended_Header Format (DB_XHDR)

The extended header is optionally recorded, as controlled by a question in Setup. There are two versions, either extended_header_v0 or extended_header_v1. Generally, the v1 version is used for moving-platform systems, such as on ships.

3.3.2 2-byte Axis of Dilliation Format (DB_AXDIL2)

The angle in degrees is stored as a signed number in tenths of degrees.

$$angle = \frac{N}{10}$$

–1800	–180.0 degrees
0	0.0 degrees
10	1.0 degree

3.3.3 1-byte Reflectivity Format (DB_DBT&DB_DBZ)

For reflectivity data, the number in decibels is computed from the unsigned output with the formula:

$$dBZ = \frac{N - 64}{2}$$

The overall range is from –31.5 dBZ to +95.5 dBZ in half dB steps as follows. In data product files, the value of 255 indicates areas not scanned.

	Ingest Data	**Products**
0	No data available	No data available
1	–31.5 dBZ	–31.5 dBZ
64	0.0 dBZ	0.0 dBZ
128	32.0 dBZ	32.0 dBZ
129	32.5 dBZ	32.5 dBZ
254	95.0 dBZ	95.0 dBZ or above
255	95.5 dBZ or above	Area not scanned

3.3.4 2-byte Reflectivity Format (DB_DBT2&DB_DBZ2)

For reflectivity data, the number in decibels is computed from the unsigned output with

$$dBZ = \frac{N - 32768}{100}$$

The overall range is from –327.67 to +327.66 in 1/100 of a dB steps as follows.

0	No data available
1	–327.67 dBZ
32768	0.00 dBZ
32769	0.01 dBZ
65534	327.66 dBZ
65535	Reserved for area not scanned in product files

3.3.5 2-byte Deformation Format (DB_DEFORM2)

Deformation is stored in a signed 16-bits number scaled to 10**–7. Only positive numbers are possible. These number are normally displayed in units of 10**–4.

0	0 deformation
1	0.001 10**–4 deformation
32766	32.766 10**–4 deformation
32767	Area not scanned

3.3.6 2-byte Divergence Format (DB_DIVERGE2)

Divergence is stored in a signed 16-bits number scaled to 10**–7. Positive number indicate divergence while negative are convergence. These number are normally displayed in units of 10**–4.

–32768	32.768 10**–4 convergence
0	0 divergence
1	0.001 10**–4 divergence
32766	32.766 10**–4 divergence
32767	Area not scanned

3.3.7 2-byte Floating Liquid Format (DB_FLIQUID2)

Rainfall accumulations are stored in 16-bit floating-point representation of a 27-bit integer in units of 0.001 mm. This product does not have a code for thresholded data, instead that area is assumed to have zero rain. The floating-point number consists of a 4-bit exponent in the high four bits, followed by a 12-bit mantissa. It uses implied digits and soft underflow. If the exponent is 0, the mantissa consists of the value. If the exponent is 1 through 15, the 12-bit mantissa has a 1 prefixed to it and is shifted up by one less than the exponent. The table below gives some examples:

Float		**Fixed**	**Meaning**
0	—>	0	0.000 mm
1	—>	1	0.001 mm
255	—>	255	0.255 mm

1000	—>	1000	1.000 mm
9096	—>	10000	10.000 mm
22634	—>	100000	100.000 mm
34922	—>	800000	800.000 mm
50000	—>	10125312	10125.312 mm
65534	—>	134184960	134184.960 mm or above
65535	—>		Area not scanned

3.3.8 1-byte Echo Tops Format (DB_HEIGHT)

Echo tops are stored to the nearest 100 meters above the ground. Because 0 is the code for no data, values stored in the bytes can be converted to kilometers by subtracting 1 and dividing by 10. The value 254 indicates that an echo is known to exist from the data, but it could not be measured.

0	No echo tops data available
1	0.0 km
128	12.7 km
129	12.8 km
253	25.2 km or above
254	Top exists above the maximum tilt
255	Area not scanned

3.3.9 2-byte Horizontal wind direction Format (DB_HDIR2)

The angle in degrees is stored as a signed number in tenths of degrees.

$$angle = \frac{N}{10}$$

–1800	–180.0 degrees
0	0.0 degrees
10	1.0 degree

3.3.10 1-byte KDP Format (DB_KDP)

Specific differential phase (KDP) is stored in a signed 8-bit number using a log scale. The KDP angles are multiplied by the wavelength in cm then scaled using a log scale separately for both signs. The minimum value is 0.25, and the maximum value is 150.0 deg*cm/km. KDP values are defined as the one-way differential effect of the intervening weather. The table below gives some examples:

$$KDP \times \lambda = minimum \times \left[\frac{minimum}{maximum}\right]^{\left[\frac{N-129}{126}\right]}$$

Here is the conversion equation for positive values (stored value above 128):

$$KDP \times \lambda = 0.25 \times 600^{\left[\frac{N-129}{126}\right]}$$

Here is the conversion equation for negative values (stored value below 128):

$$KDP \times \lambda = -0.25 \times 600^{\left[\frac{127-N}{126}\right]}$$

Value	Meaning KDP*L	KDP for 10 cm	KDP for 5 cm
0	No data available		
1	–150.00 deg*cm/km	–15.00 deg/km	–30.00 deg/km
2	–142.58	–14.26	–28.51
127	–0.250	–0.025	–0.050
128	0.000	0.000	0.000
129	0.250	0.025	0.050
130	0.263	0.026	0.053
254	142.58	14.58	28.51
255	Area not scanned		

3.3.11 2-byte KDP Format (DB_KDP2)

Specific differential phase (KDP) in degrees per kilometer is computed from the unsigned output with:

$$KDP = \frac{N - 32768}{100}$$

The overall range is from –327.67 to +327.66 in 1/100 of a degree/kilometer steps as follows.

0	No data available
1	–327.67 deg/km
32768	0.00 deg/km
32769	0.01 deg/km
65534	327.66 deg/km
65535	Reserved for area not scanned in product files

3.3.12 1-byte LDR Format (DB_LDRH & DB_LDRV)

LDR is short for "Linear Depolarization Ratio". This is the ratio of the power received in the depolarized channel to the power received in the main channel. There are 2 flavors: If you are transmitting horizontal, then the ratio of the vertical power to the horizontal power is "LDRH". Similarly if transmitting vertical then the ratio of horizontal power to vertical power is "LDRV". Note that in simultaneous transmission LDR cannot be computed because the signal in the depolarized channel is dominated by co–polarized returns.

$$LDR(db) = \frac{N - 1}{5} - 45.0$$

Like most power ratios this is expressed in dB. The overall range is from –45 to +5.6 in steps of 0.2 dB as follows.

0	No data available
1	–45.0 dB
2	–44.8 dB
226	0.0 dB
254	+5.6 dB
255	Reserved for area not scanned in product files

3.3.13 2-byte LDR Format (DB_LDRH2 & DB_LDRV2)

Same as DB_DBZ2 format, see section 3.3.4.

3.3.14 1-byte Phi Format (DB_PHIH & DB_PHIV)

The cross channel differential phase. Same format as DB_PHIDP, see section 3.3.16.

3.3.15 2-byte Phi Format (DB_PHIH2 & DB_PHIV2)

The cross channel differential phase. Same format as DB_PHIDP2, see section 3.3.17.

3.3.16 1-byte PhiDP Format (DB_PHIDP)

PhiDP is defined as the two-way measured phase effect after the signal has travelled through the intervening weather twice. Differential phase (PhiDP) in degrees is computed from the unsigned output with:

$$\Phi_{DP} \bmod 180 = 180 * \frac{N - 1}{254}$$

The overall range is from 0 to 180 degrees in steps of 0.71 as follows.

0	No data available
1	0.00 deg
2	0.71 deg
101	70.87 deg
254	179.29 deg
255	Reserved for area not scanned in product files

3.3.17 2-byte PhiDP Format (DB_PHIDP2)

PhiDP is defined as the two-way measured phase effect after the signal has travelled through the intervening weather twice. Differential phase (PhiDP) in degrees is computed from the unsigned output with:

$$\Phi_{DP} \bmod 360 = 360 * \frac{N - 1}{65534}$$

The overall range is from 0 to 360 degrees in steps of 0.0055 as follows. In cases where the transmitter was alternating polarization, or the data was converted from 1–byte format the values will only cover 0–180 degrees.

0	No data available
1	0.0000 deg
2	0.0055 deg
65534	359.9945 deg
65535	Reserved for area not scanned in product files

3.3.18 2-byte Rainfall Rate Format (DB_RAINRATE2)

Rainfall rates are stored in 16-bit floating-point representation of a 27-bit integer in units of 0.0001 mm/hr. The floating-point number consists of a 4-bit exponent in the high four bits, followed by a 12-bit mantissa. It uses implied digits and soft underflow. If the exponent is 0, the mantissa consists of the value. If the exponent is 1 through 15, the 12-bit mantissa has a 1 prefixed to it and is shifted up by one less than the exponent. The table below gives some examples:

Float		**Fixed**	**Meaning**
0	—>	0	No data available
1	—>	1	0.0000 mm/hr
2	—>	2	0.0001 mm/hr
255	—>	255	0.0254 mm/hr
1000	—>	1000	0.0999 mm/hr
9096	—>	10000	0.9999 mm/hr
22634	—>	100000	9.9999 mm/hr
34922	—>	800000	79.9999 mm/hr
50000	—>	10125312	1012.5311 mm/hr
65534	—>	134184960	13418.4959 mm/hr or above
65535	—>		Area not scanned

3.3.19 1-byte Rho Format (DB_RHOH & DB_RHOV)

The cross channel correlation coefficient. Same format as DB_RHOHV, see section 3.3.21.

3.3.20 2-byte Rho Format (DB_RHOH2 & DB_RHOV2)

The cross channel correlation coefficient. Same format as DB_SQI2, see section 3.3.25.

3.3.21 1-byte RhoHV Format (DB_RHOHV)

RhoHV a measure of how correlated the power fluctuations in reflectivity in the horizontal receiver is to the vertical receiver. It is a dimensionless number on a scale for which 0 means no correlation, and 1 means complete correlation. Since numbers are more likely near 1, the data is scaled with a square root.

$$RhoHV = \sqrt{\frac{N-1}{253}}$$

0	Data not available at this range
1	0.0000
2	0.0629
128	0.7085
253	0.9980
254	1.0000
255	Area not scanned

3.3.22 2-byte RhoHV Format (DB_RHOHV2)

Same format as DB_SQI2, see section 3.3.25.

3.3.23 1-byte Wind Shear Format (DB_SHEAR)

Wind shear is stored to the nearest 0.2 meters per second per kilometer. This is a signed number, with positive indicating wind is increasing in velocity away from the radar with increasing range. To convert, subtract 128 and multiply by 0.2.

0	No wind shear data available
1	–25.4 m/s/km or above
128	0.0 m/s/km
129	+0.2 m/s/km
254	+25.2 m/s/km or above
255	Area not scanned

3.3.24 1-byte Signal Quality Index Format (DB_SQI)

The Signal Quality Index (SQI) is the ratio of the magnitude of R1 to the magnitude of R0. It is a dimensionless number on a scale for which 0 means pure noise, and 1 means a pure sine wave.

$$SQI = \sqrt{\frac{N-1}{253}}$$

0	SQI data not available at this range
1	0.0000
2	0.0629

128	0.7085
253	0.9980
254	1.0000
255	Area not scanned

3.3.25 2-byte Signal Quality Index Format (DB_SQI2)

Stored linearly using 16-bits as follows:

$$SQI = \frac{(N - 1)}{65533}$$

0	SQI data not available at this range
1	0.00000
2	0.00002
128	0.00194
65533	0.99998
65534	1.00000
65535	Area not scanned

3.3.26 2-byte Time Format (DB_TIME2)

Stored as time in seconds, so in minutes the conversion is as follows:

$$Time = \frac{N - 32768}{60}$$

0	Time data not available at this range
1	– 9:06:07:
32768	00:00:00
32828	00:01:00
65535	Area not scanned

3.3.27 1-byte Velocity Format (DB_VEL)

Mean velocity is with respect to the Nyquist velocity, and is expressed as follows. In data product files, the value 255 is used to indicate area not scanned. The Nyquist velocity is wavelength times PRF divided by 4. For 2:3 dual PRF mode, it is doubled; for 4:3 PRF mode, it is tripled, and for 4:5 PRF mode it is quadrupled over that of the higher PRF. For alternating polarization it is halved.

$$Velocity = \frac{N - 128}{127} \times Nyquist$$

0	Velocity data not available at this range
1	Unambiguous velocity towards the radar
128	Zero velocity
255	Unambiguous velocity away from the radar

3.3.28 2-byte Velocity Format (DB_VEL2)

Mean velocity in meters per second is computed from the unsigned output with

$$velocity = \frac{N - 32768}{100}$$

The overall range is from –327.67 to +327.66 in 1/100 of a meter per second steps as follows.

0	No data available
1	–327.67 m/s (towards the radar)
32768	0.00 m/s
32769	0.01 m/s
65534	327.66 m/s (away from the radar)
65535	Reserved for area not scanned in product files

3.3.29 1-byte Unfolded Velocity Format (DB_VELC)

This is mean radial velocity corrected for both Nyquist folding and fall speed. It is scaled to cover a fixed span of +/– 75 meters/second. In data product files, the value 255 is used to indicate area not scanned.

0	Velocity data not available at this range
1	–75.0 m/s (towards the radar)
2	–74.4 m/s
128	Zero velocity
129	+0.6 m/s
254	+75.0 m/s (away from the radar)
255	Area not scanned

3.3.30 2-byte Unfolded Velocity Format (DB_VELC2)

Mean velocity in meters per second is computed from the unsigned output with

$$velocity = \frac{N - 32768}{100}$$

The overall range is from –327.67 to +327.66 in 1/100 of a meter per second steps as follows.

0	No data available
1	–327.67 m/s (towards the radar)
32768	0.00 m/s
32769	0.01 m/s
65534	327.66 m/s (away from the radar)
65535	Reserved for area not scanned in product files

3.3.31 2-byte VIL Format (DB_VIL2)

Vertically integrated liquid is stored in 16-bits to the nearest 0.001 mm. Because 0 is the code for no data, values stored in the bytes can be converted to millimeters by subtracting 1 and dividing by 1000.

0	No VIL data available
1	0.000 mm
128	0.127 mm
129	0.128 mm
255	0.254 mm
65534	65.533 mm or above
65535	Area not scanned

3.3.32 2-byte Vertical Velocity Format (DB_VVEL2)

Vertical velocity is stored in a signed 16-bits number scaled to 0.01 meters/second. Positive number indicate upward motion, negative downward.

–32768	327.68 m/s upward motion
0	0 motion
1	0.01 m/s fall speed
32766	327.66 m/s fall speed
32767	Area not scanned

3.3.33 1-byte Width Format (DB_WIDTH)

Spectrum width is computed from the unsigned output as:

$$W = \frac{n}{256}$$

The overall range is therefore a fraction between 1/256 and 255/256. The code of 0 indicates that width data is not available at this range. To convert the width to meters per second, multiply by the unambiguous velocity. Thus the width has twice the resolution of the velocity. This unambiguous velocity is not enlarged by the dual PRF scheme, but is halved by alternating polarization. In data products, the value 255 indicates area not scanned. Note that width unambiguous velocities are not changed for dual PRF unfolding.

3.3.34 2-byte Width Format (DB_WIDTH2)

Spectral width in meters per second is computed from the unsigned output with

$$width = \frac{N}{100}$$

The overall range is from 0.01 to 655.34 in 1/100 of a meter per second steps as follows.

0	No data available
1	0.01 m/s
32768	327.68 m/s
32769	327.69 m/s

65534	655.34 m/s
65535	Reserved for area not scanned in product files

3.3.35 1-byte ZDR Format (DB_ZDR)

For differential reflectivity data, the number in decibels is computed from the unsigned output with the formula:

$$dB\ (ZDR) = \frac{N - 128}{16}$$

The overall range is from –7.94 dBZ to +7.94 dBZ in sixteenth of a dB steps as shown below. Positive ZDR means that the horizontal return is stronger than the vertical return. In data products, the value 255 indicates area not scanned.

0	No ZDR data available
1	–7.94 dB
128	0.00 dB
129	+0.06 dB
255	+7.94 dB

3.3.36 2-byte ZDR Format (DB_ZDR2)

For differential reflectivity data, the number in decibels is computed from the unsigned output with

$$dB(ZDR) = \frac{N - 32768}{100}$$

The overall range is from –327.67 to +327.66 in 1/100 of a dB steps as follows.

0	No data available
1	–327.67 dB
32768	0.00 dB
32769	0.01 dB
65534	327.66 dB
65535	Reserved for area not scanned in product files

3.4 Ingest Data File Format

Each ingest data file contains one data type from one sweep of a volume. If extended headers are recorded, they are treated as another data type and are placed in a separate file. Each of these files consists of a fixed length ingest_data_header structure followed by a variable-length ray pointer table, followed by a variable-length segment of data, as shown in Figure 3–1. The ray pointer table contains a 32-bit pointer for each ray in the file. These are byte pointers (origin one) referenced to the beginning of the data area. Thus, a pointer value of 1 refers to the first byte of the data area. Ray data are forced onto 16-bit word boundaries, so all the pointers are odd. A pointer value of 0 indicates that no data are available for that slot.

Figure 3–1: Ingest Data File Format

<ingest_data_header> 76 Bytes
Pointer Table 4 Bytes per expected ray
Data Area Undetermined Size

Ray data are partially compressed with a truncation scheme. If a ray does not contain data all the way out to the last range bin, those trailing bins are removed from the archive. This means that the storage space required for each ray varies, which requires the pointer table for quick random access. The pointer points to a ray_header structure, which is followed by an array of range bins. The count of the number of range bins could be zero, in which case the ray contains only the 12-byte ray header.

3.4.1 Ingest File Names

The IRIS ingest data for a volume scan is stored on disk in multiple files. The ingest summary file name is just the prefix with a trailing ".", but without the suffix. Each sweep and data type is stored in a separate file. These files must be named correctly, **siris** will delete any with an incorrect name. If the two-digit year is less than 50, it means add 2000, otherwise add 1900.

General Format	Example
SSSYYMMDDHHMMSS.##DD	SIG010620141921.01dBZ
Where:	
SSS—Three letter site code from Setup	SIG—Abbreviation for SIGMET
YYMMDDHHMMSS—Data time	940620141921—14:19:21 20 June 2001
##—Two-digit sweep number	01—Sweep #1
DD—data type (1–5 chars)	dBZ—Reflectivity

3.5 Product File Format

Each product is stored in a separate file in the directory specified by the **IRIS_PRODUCT** environment variable. Raw products are stored in a separate directory specified by the **IRIS_PRODUCT_RAW** environment variable. Product file names consist of the 15-character site and time to the left of the dot, and a 7-character product type code and machine generated string to the right.

The file consists of the product_hdr structure followed by the data. The product configuration structure is exactly what is in one of the product configuration files. It specifies how a product should be generated, and so includes some information not strictly required to appear in the final disk file.

3.5.1 Cartesian Product Format

The data portion of Cartesian product files is not compressed and consists of an array. The first byte contains the lower left corner of the image, the second contains the pixels to the right of that, and so on from the bottom of the window to the top. For 3-D products, the lowest 2-D image comes first.

3.5.2 FCAST Product Format

The FCAST product format is the same as other Cartesian products except that the data elements consist of the ndop_results structure, rather than a 1- or 2-byte number.

3.5.3 NDOP Product Format

The NDOP product format is the same as other Cartesian products except that the data elements consist of the ndop_results structure, rather than a 1- or 2-byte number. 3-D data is supported.

3.5.4 RAW Product Format

The raw product is a collection of all raw ingest data acquired during a run of a single task (volume scan). Whereas the ingest data files are stored on disk separately by sweep and by data moment, the raw product is a single file into which many ingest files have been incorporated. For hybrid tasks, individual raw Products are made from each of the individual tasks that make up the overall scan.

Raw product files are blocked into 6144-byte records, which match the record length used if and when the files are eventually written to tape. Mimicking the tape structure on disk permits error recovery to be built directly into the raw data format. For all other types of products, if a tape I/O error occurs within the product's records on tape, then the entire product is lost. For raw archive this would be too great a penalty, since an entire volume scan is at stake. The blocking scheme permits partial error recovery while still maintaining a one-to-one mapping between disk and tape formats.

i

Note: The records are for IRIS interpretation only and do not refer to any operating system file records.

The first 6144 byte record of a raw product holds the product header structure. Thus, the raw product begins with a product header like all the other types of products. The only difference is the zero padding out to 6144 bytes. The next record holds the ingest_header structure for the volume scan that supplied the data. Again, this structure is zero-padded to fill the entire second record. All subsequent records hold the actual data, and each record begins with the raw_prod_bhdr structure described in Section 3.2.27.

Records hold data from one and only one sweep, and the sweep number is found in each record's header. The data for the sweep is the concatenation of all the data from as many records as pertain to that sweep. When data for a sweep ends short of the 6144 byte record size, the remainder of that record is padded with zeros. Each of these sweep data sets begins with the ingest_data_header structures for each data type that was recorded. The same number of headers are found in the beginning of a sweep's data as there are data types acquired during the sweep. The list of data types recorded is specified by the data collection mask in the task_dsp_info within the task_configuration within the ingest_header. The actual rays of data are found immediately after the headers.

Rays are ordered within a sweep by the same ordering sequence used for the ingest data file pointer table. If a ray is missing from the ingest file, a zero-length ray is inserted into the product file as a placeholder. Thus, the number of compressed rays in the file is equal to the product of the number of data types recorded and the number of angles sampled. This is true even if data were not actually acquired at some of those angles. Within a ray, the recorded data are ordered by increasing data type number. See the task_dsp_info structure for a definition of the data type numbers. All data rays are compressed using the algorithm described in Section 3.5.4.1. Note that the raw product headers and the ingest data headers are not compressed. The overall organization of the file is shown below. Raw product files are blocked into 6144-byte records, and all but the first two records begin with the raw_prod_bhdr 12-byte structure.

Figure 3–2: RAW product format

```
Record #1  { <product_hdr>    0,0,0...   }
Record #2  { <ingest_header>    0,0,0...   }
Record #3  { <raw_prod_bhdr> <ingest_data_header(s)> Data... }
Record #4  { <raw_prod_bhdr>   Data...    }
   .               .            .
   .               .            .
Record #N  { <raw_prod_bhdr>   Data  0... }
Record #N+1{ <raw_prod_bhdr> <ingest_data_header(s)> Data... }
Record #N+2{ <raw_prod_bhdr>   Data...    }
   .              .            .
   .              .            .
Record #M  { <raw_prod_bhdr>   Data  0... }
```

3.5.4.1 Data Compression Algorithm

To make the best use of storage, all radar rays are compressed before being inserted into the file. The compression algorithm is 16-bit word based, and simply removes runs of zeros. This complements the signal processor, which zeros data that does not meet the threshold requirements in effect. Runs of one or two zeros are not removed because there is no benefit. The data field starts with a compression code value. The code either indicates the number of zeros that were skipped, or the number of data words that follow. In the case of a zero skipped code, it is immediately followed with another code value. In the case of a data code, the next code follows the data.

Table 3–3: Compression Code Meanings

MSB	Low-bits	Meaning
0	0	<unused>
0	1	End of ray
0	2	<unused>
0	3 – 32767	3 to 32767 zeros skipped
1	0	<unused>
1	1 – 32767	1 to 32767 data words follow

3.5.4.2 Raw Product Example

Here is an example of what the third record of a simple raw product would look like. This product is for data with only velocity recorded. Shown is the first ray of a PPI at azimuth 0 to 1 degree, elevation 0.5 degrees. It has 200 range bins, with no data for the first 100 bins, then one bin of zero velocity, then 99 bins with no data.

Table 3–4: Raw Product Example

Byte	Size	Contents	
0	12	<raw_prod_bhdr>	
12	76	<ingest_data_header>	
88	SINT2	–32762 (code for six data words to follow)	
90	BIN2	0 (starting azimuth)	Six
92	BIN2	91 (starting elevation)	words
94	BIN2	182 (ending azimuth)	referred
96	BIN2	91 (ending elevation)	to
98	SINT2	200 (number of range bins)	above
100	UINT2	3 (time)	
102	SINT2	50 (code for fifty zeros skipped)	
104	SINT2	–32767 (code for one data word follows)	
106	SINT2	128 (zero velocity value)	One data word
108	SINT2	49 (code for forty nine zeros skipped)	
110	SINT2	1 (code for end of ray)	
112			
		(Continues on to next ray)	

3.5.5 SLINE Product Format

The first 1024 bytes of the product hold a copy of the protect_setup structure described in Section 3.2.25. This is followed by an array of sline_results structures one for each shearline found. The "Number of elements in product results array" element of the product_end structure in the product_hdr indicates the number of points in the array. Generally there is at most 1 shearline found.

3.5.6 TDWR Product Format

The first 1024 bytes of the product hold a copy of the protect_setup structure described in Section 3.2.25. This structure is copied from the setup files on the integrating computer. This is followed by an array of tdwr_results structures, one for each corridor. The "Number of elements in product results array" element of the product_end structure in the product_hdr indicates the number of elements in the array. Only corridors covered by one of the input products to the integrator will be included, and corridors which are unused at generation time will normally be removed. This is followed by an array of warning_results structures copied from the WARN input product, if any. This is followed by an array of sline_results structures copied from the SLINE input product, if any. The sizes of these arrays are in the tdwr_psi_struct portion of the product header.

3.5.7 TRACK Product Format

The first 1024 bytes of the product hold a copy of the protect_setup structure described in Section 3.2.25. This is followed by an array of track_results structures, one for each track point. The "Number of elements in product results array" element of the product_end structure in the product_hdr indicates the number of points in the array. Points must be in time order, with the oldest first. Within points of the same time, they are sorted by index number. Only one data point of each index value at each time is allowed, except that multiple text points are allowed (which have index set to zero).

3.5.8 VVP Product Format

The winds produced by the VVP product are stored in an array of vvp_results structures for each height. Note that if less than 30 range bins are found in the height interval, then the whole structure is zeroed. Therefore any analysis program should check the "Number of data points used" field. Also if the calculation cannot be performed for some other reason, then the standard deviation will be set to 32767. Analysis programs should also check the standard deviations. Data fields which are turned off in the product configuration will be set to zero. Therefore the "Wind parameters mask" in the vvp_psi_struct in the product header needs to be checked also. Note that only data bins with valid velocities not near zero are included in the calculation. To compute the average reflectivity, bins must also have a valid reflectivity. Because produces a lower number of valid bins, so the number of valid reflectivity bins is also recorded.

3.5.9 WARN Product Format

The first 1024 bytes of the product hold a copy of the protect_setup structure described in Section 3.2.25. This is followed by an array of warning_results structures. The number of warning_results structures in the array is given in the product_end structure, part of the product_hdr.

3.5.10 WIND Product Format

The first 84 bytes of the product hold a vvp_results structure for the whole volume. This is followed by an array of wind_results structures. The number of wind_results structures in the array is determined by multiplying the number of points in range by the number of points in azimuth stored in the product_specific_info structure in the header.

3.5.11 Product File Names

Because IRIS product files are computer accessed, actual file names are unimportant. IRIS can handle a product with any file name up to 23 characters. When IRIS creates a product file in its own product directory, it uses the following file name syntax. The two digit year used is simply the year modulo 100. Since the product file names are never parsed to generate a full date, there is no need to ever reconstruct the correct century.

General Format	Example
SSSYYMMDDHHMMSS.PPPXXXX	SIG940620141921.TRAE090
Where:	
SSS—Three letter site code from Setup	SIG—Abbreviation for SIGMET
YYMMDDHHMMSS—Data time	940620141921—14:19:21 20 June 1994
PPP—Three-letter product type	TRA—Track product
XXXX—Characters for uniqueness	E090

When IRIS copies files to other directories using the network product output, it must generate a file name that is unique in the target directory. To see the choices available for this, see the SETUP/OUTPUT section of the *IRIS Utilities Manual*.

3.6 Tape Format

All tapes made by IRIS hold exact images of corresponding disk-based product files. The tapes are always written using fixed-length 6144-byte records, where the last tape record is padded with zeros, if necessary, to the full 6144-byte length. Products are separated on tape by end-of-file (EOF) marks. Prior to Version 5.00, tapes ended with a double EOF. Because all disk product files begin with a product_hdr structure, this is also the structure initially encountered in the first record of each product on tape. By examining the headers, you can determine what kinds of product files have been stored on the tape.

There is a special short record at the beginning of the tape that serves to identify how and when the tape was initially created by the **`init_iris_tape`** utility. This record contains the tape_header_record structure, and is followed by an EOF and the product files, if any. There are no special directory or inventory records on the tape. After a tape has been started, the only additional writing that can be done is to append more product file images to the end.

3.7 TIFF Output Format

IRIS can output images over the network in TIFF format. These files conform to the TIFF revision 6.0 standard. This standard supports many different types of images. Only a small subset of this is required. IRIS uses only baseline TIFF, and none of the TIFF extensions. Only one image is included in each file. It is a Palette Color image. The table below lists all the fields set by IRIS. The Image Description contains the 5 character product type, followed by the 12 character product configuration name. Compression is controlled by a setup question for the output device.

Table 3–5: TIFF Fields Used by IRIS

Software	305	"IRIS 5.56" for example
ImageDescription	270	"IRIS PPPPP NNNNNNNNNNNN"
DateTime	306	Time of ingest data
ImageWidth	256	Width of image in pixels
ImageLength	247	Height of image in pixels
Compression	259	Either none or PackBits
PlanarConfiguration	284	1 (Chunky)
SamplesPerPixel	277	1
Orientation	274	1 (Top Left)
RowsPerStrip	278	Height of image in pixels

3.8 Constants

Table 3–6: Data Type Constants — /include/dsp_lib.h

Extended Header is included here though it is not generated by the DSP. In general, types 0–31 could be produced by the DSP.		
DB_XHDR	(0)	Extended Headers
DB_DBT	(1)	Total power (1 byte)
DB_DBZ	(2)	Reflectivity (1 byte)
DB_VEL	(3)	Velocity (1 byte)
DB_WIDTH	(4)	Width (1 byte)
DB_ZDR	(5)	Differential reflectivity (1 byte)
DB_DBZC	(7)	Corrected reflectivity (1 byte)
DB_DBT2	(8)	Total power (2 byte)
DB_DBZ2	(9)	Reflectivity (2 byte)
DB_VEL2	(10)	Velocity (2 byte)
DB_WIDTH2	(11)	Width (2 byte)
DB_ZDR2	(12)	Differential reflectivity (2 byte)
DB_RAINRATE2	(13)	Rainfall rate (2 byte)
DB_KDP	(14)	KDP (Differential phase) (1 byte)
DB_KDP2	(15)	KDP (Differential phase) (2 byte)
DB_PHIDP	(16)	PhiDP(Differential phase) (1 byte)
DB_VELC	(17)	Corrected velocity (1 byte)
DB_SQI	(18)	SQI (1 byte)
DB_RHOHV	(19)	RhoHV (1 byte)
DB_RHOHV2	(20)	RhoHV (2 byte)
DB_DBZC2	(21)	Corrected Reflectivity (2 byte)
DB_VELC2	(22)	Corrected velocity (2 byte)
DB_SQI2	(23)	SQI (2 byte)
DB_PHIDP2	(24)	PhiDP (Differential phase) (2 byte)
DB_LDRH	(25)	LDR xmt H, rcv V (1 byte)
DB_LDRH2	(26)	LDR xmt H, rcv V (2 byte)
DB_LDRV	(27)	LDR xmt V, rcv H (1 byte)
DB_LDRV2	(28)	LDR xmt V, rcv H (2 byte)
. . .		
DB_HEIGHT	(32)	Height (1/10 km) (1 byte)
DB_VIL2	(33)	Linear liquid (.001mm) (2 byte)

Table 3–6: Data Type Constants — /include/dsp_lib.h (cont.)

DB_RAW	(34)	Raw Data
DB_SHEAR	(35)	Wind Shear (1 byte)
DB_DIVERGE2	(36)	Divergence (2 byte)
DB_FLIQUID2	(37)	Floated liquid (2 byte)
DB_USER	(38)	User type, unspecified data (1 byte)
DB_OTHER	(39)	Unspecified data, no color legend (1 byte)
DB_DEFORM2	(40)	Deformation (2 byte)
DB_VVEL2	(41)	Vertical velocity (2 byte)
DB_HVEL2	(42)	Horizontal velocity (2 byte)
DB_HDIR2	(43)	Horizontal wind direction(2 byte)
DB_AXDIL2	(44)	Axis of dilatation (2 byte)
DB_TIME2	(45)	Time in seconds (2 byte)
DB_RHOH	(46)	Rho, xmt H, rcv V (1 byte)
DB_RHOH2	(47)	Rho, xmt H, rcv V (2 byte)
DB_RHOV	(48)	Rho, xmt V, rcv H (1 byte)
DB_RHOV2	(49)	Rho, xmt V, rcv H (2 byte)
DB_PHIH	(50)	Phi, xmt H, rcv V (1 byte)
DB_PHIH2	(51)	Phi, xmt H, rcv V (2 byte)
DB_PHIV	(52)	Phi, xmt V, rcv H (1 byte)
DB_PHIV2	(53)	Phi, xmt V, rcv H (2 byte)