Software Defined Radio Forum Contribution

Committee: Technical

Title: VITA-49 Draft Specification

Source Lee Pucker
SDR Forum
604-828-9846
Lee.Pucker@sdrforum.org

Date: 7 March 2007

Distribution: Post on Web Site, Members Only

Document Summary: This defines a specification for moving data between an RF Front end and signal processing subsystem

Notes of importance: None

Impacts/Effects: Use of this document is expected to impact the work done by the Transceiver API Task group.

Action Desired: None

Action Required for Closure : None

NOTICE

This document has been prepared by VITA to assist the SDR Forum (or its successors or assigns). It is intended for discussion purposes only. It may be amended or withdrawn at a later time and it is not binding on any member of the SDR forum or on VITA.

Use of this document is governed by the SDR Forum/VITA MOU.

VITA Radio Transport (VRT) Draft Standard

VITA-49.0 – 2006

Draft 0.16
8 December 2006

This standard is being prepared by the
VITA Standards Organization (VSO)
and is

unapproved.

Do not specify or claim conformance to this document.

VSO is the Public Domain Administrator of this draft specification,
and guards the contents from change except by sanctioned
meetings of the task group under due process.

VITA Standards Organization
PO Box 19658
Fountain Hills, AZ 85269

Ph: 480-837-7486
Fax: Available upon request
URL: www.vita.com

DRAFT REVISION HISTORY

Draft No.	Date	Comments & Major Changes/Updates
D0.9	1 March 2006	Major re-write; adding Context Packet & User Defined Context Packet; changing Streaming Data Packet to add all possible formats in Data Payload.
D0.10	7 March 2006	Aaron Kaiway updates to Section 1.0; Moved two definitions from Section 4.0 to the Glossary; Bob Normoyle updates to Section 6.0.
D0.11	19 June 2006	Recombined the edited sections after the group reached a consensus on all major issues.
D0.12	21 June 2006	- Removed spectral polarity, change, and synchronization event indicators from the Data and Context packet event field. Struck out rule 6.2.5.12-5, which was an accidental copy from the Data Packet section. - Added some missing references - Changed “TS” field in Data Packet to “TSI,” which matches Context Packet. - Incorporated new compliance section. - Fixed misleading wording related to Timestamp Adjustment. Also added a rule explicitly stating how Timestamp Adjustment affects Context change timing. - Unified Figure and Table labeling. Created proper list of Figures and list of Tables. - Deleted some unused references.
D0.13	6 July 2006	- Modified Compliance section to reflect latest consensus. - Added Appendix for compliance specification.
D0.14	21 Sept. 2006	- Inserted new Introduction section. - Modified “The Interpretation of Data Items in the Data Payload” Section using Conrad Romberg’s new text. - Modified the number of data packets, and their meaning. - Modified the compliance section and moved it to the end. - Moved compliance tables into the compliance section - Changed trailer bits from “Event Tags/Logical Indicators” to “Data Qualifiers.” - Added a compliance documentation example, now Appendix A. - Added Reference Point Example Appendix - Added Spectral Fields Example Appendix - Made wording changes too numerous to document here.
D0.15	27 Nov 2006	- Changed name to VRT - Reworked the Introduction and Overview. - Added section for Information Classes and Streams. - Replaced “Vendor-Defined” with “Extension.” - Numerous text changes.
D0.16	8 Dec 2006	- Misc. changes to Introduction and to Section 7.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The VITA Radio Transport (VRT) standard defines a transport-layer protocol designed to promote interoperability between RF (radio-frequency) receivers and signal processing equipment in a wide range of applications. These include spectral monitoring, communications, radar and others. VRT enables high-precision time stamping to maintain phase coherency and time synchronization between multiple receiver channels and is an ideal building block to support data streaming for Software Defined Radio (SDR) applications.

The benefits of the VRT protocol include:

- Transport-layer **interoperability** between diverse equipment providers, reducing integration time and effort. Allows system infrastructures in which equipment from multiple vendors can be selected, so that technology insertion is simplified. This is accomplished by:
    - **Standardization of signal data interfaces** between receivers and signal processors with a wide variety of data types supported.
    - **Standardization of context information** from receivers. A standard protocol to convey a receiver's geolocation and equipment settings pertinent to signal processing applications.
- **Efficient packet** structures for a wide variety of signal data and related contextual information.
- Transport layer **multiplexing** of many signal channels onto the same link interface.
- **Scalability** from a single receiver channel to a large number of receiver channels.
- **Flexible architectures**, enabling data to be routed to any number of signal processors or FPGAs in the same chassis, between multiple chassis, or to any destination without degradation of the signal. Ideal for distributed sensor applications.
- **Coherency** between multiple receiver channels for both real-time and recorded use of data.
- High-precision **time stamping** for:
    - correlating sensor data to external events
    - synchronizing information from multiple sources
    - data recording applications where the information is analyzed in non-real time
    - precision geolocation applications such as direction finding (DF), beamforming, and time-difference of arrival (TDOA).

The primary focus of this standard is on RF signals and equipment, though VRT is also useful for the digital transport of other signals, including acoustic and video signals.

The VRT protocol is motivated by the need to reduce the complexity and expense associated with RF receiver systems. The VRT protocol enables systems to migrate from proprietary stove-pipe architectures to interoperable multi-function architectures. Figure 1-1 shows a stove-pipe architecture, on the left, using proprietary analog IF (intermediate frequency) interfaces transformed into an architecture using open VRT interfaces, on the right.

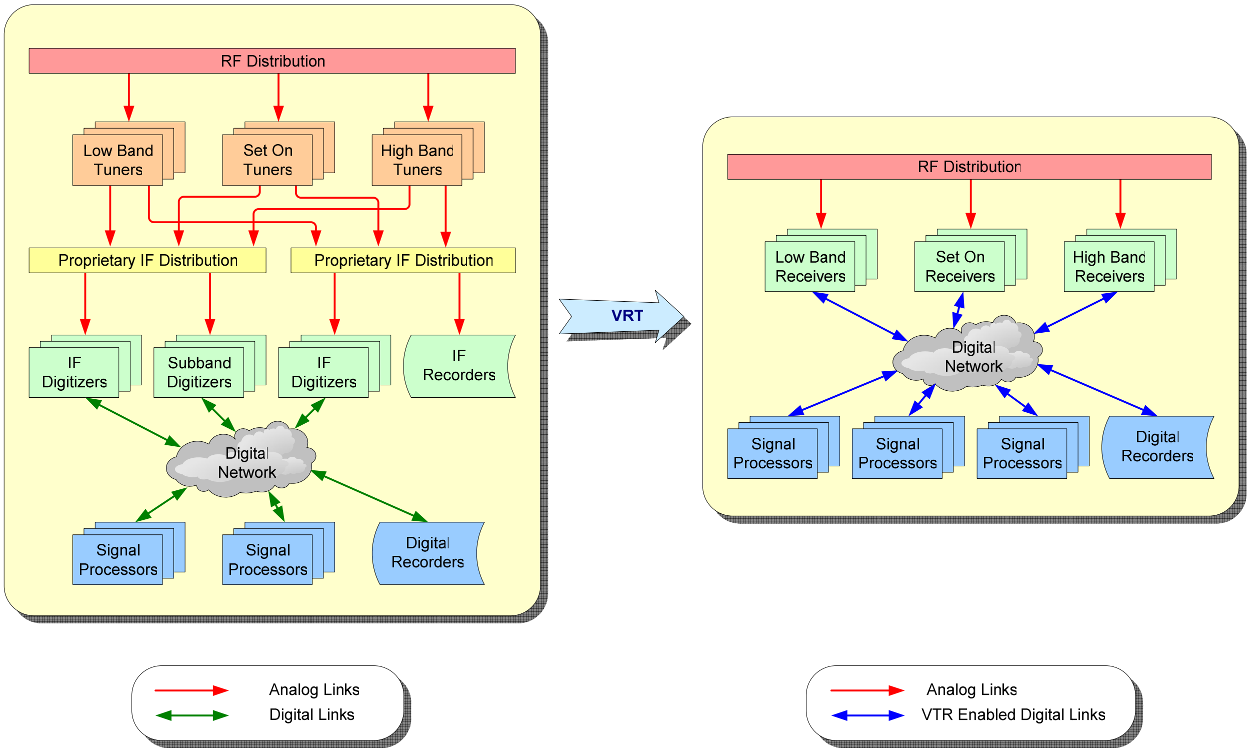


Figure 1-1: VRT simplifies traditional RF signal processing systems
The VRT enabled system (right) standardizes the system's IF distribution and allows system designers to eliminate cumbersome point-to-point analog IF distribution assemblies.

The emergence of digitized IF, and high-speed serial interconnects that can carry it, allows system designers to replace the complicated analog IF distribution networks with digital networks using commercial off-the-shelf (COTS) hardware. The digital network cloud on the right diagram represents the VRT protocol enabling time stamped signal data to be sent to any processing node via this open protocol standard. The interfaces between the sensors, the RF receivers, and the signal processors are standardized at the transport layer, enabling equipment from a variety of providers to be inserted with minimal impact on the application. The standardized interfaces also make it possible to use the same sensor and signal processing resources for multiple applications. By supporting an open architecture with the capability to provide multiple functions, VRT reduces both development and support costs.

The VRT protocol enables architectures to achieve the objectives of the military's Modular Open Systems Approach (MOSA). Specifically, VRT enables:

- Simpler and faster integration of products from multiple sources.
- Integration of new components and capabilities with minimal impact on the intrinsic architecture
- Elimination of the dependency upon a single supplier
- Incremental improvements without redesign of large portions of the system
- Adaptation to evolving requirements and threats
- Leveraging of commercial investments in new technologies into architectures that were previously stove-piped
- Reduction of life-cycle risk. Lowers cost of upgrades over the entire life cycle

The VRT protocol was created by, and has the support of, a wide industry base of organizations including RF receiver manufacturers, digital signal processor manufacturers, data recorder manufacturers, prime contractors and government agencies.

1.1 *Interoperability*

The VRT protocol provides transport-layer interoperability between equipment from multiple vendors. Complete interoperability also requires that the underlying transport, network, data-link, and physical layers also be matched between equipments. VRT is designed to be independent of these underlying layers, therefore it may run over common protocols such as TCP, UDP, Serial RapidIO, Xilinx Aurora, Race++, Serial Front Panel Data Port (S-FPDP), PCI Express, 1/10 Gigabit Ethernet and most any other interface.

1.2 *VRT Packets*

VRT is a flexible protocol that supports the packetized transport of both signal data, such as samples of a digitized IF, and related context information, such as tuner center frequency and GPS location. VRT enables data and context information to be conveyed together efficiently across digital links or networks.

1.2.1 Signal Data

The VRT protocol's primary feature is the capability to convey digitized samples of an RF, IF, or baseband signal in a standard format across a link or network. For this type of signal data, VRT supports a wide range of fixed and floating point formats for both real and complex data.

VRT also supports the transfer of other signal data in a customizable transport format. This VRT protocol 'extension' capability literally supports any type of data that needs to be conveyed to handle a wide range of applications. The customizable extension packets may also be used as a means to develop future VRT supplemental specifications.

1.2.2 Context Information

The VRT protocol provides a standard means of communicating context related to digitized RF and IF signals. The information that can be conveyed in context packets includes:

- Frequency
- Bandwidth
- A/D sample rate
- Gain settings
- Timestamp
- Timestamp delay (system latency)
- Support of multiple A/D sampling rates
- Geolocation using inertial navigation or GPS information

The VRT protocol allows context information to be sent only as needed by the application. In this way, VRT minimizes the bandwidth required for context information, therefore maximizing the amount of bandwidth available for signal data. The context information is time stamped so that changes can be precisely related to the associated data.

Some of the components of a typical RF application that can send context information via the VRT protocol include:

- Antennas
- Filters
- Amplifiers and Attenuators
- Analog downconverters
- Digital downconverters (DDCs)
- Analog-to-digital converters (ADCs)
- Fast-Fourier Transforms (FFTs)
- Channelizers
- Demodulators
- Radar Processors
- Direction Finding (DF)
- Beamforming

1.3 *Applications of VRT*

The VRT protocol can be used as an interface at many places in system architectures. For instance, it can be utilized for interfaces:

- between chips on the same board
- between modules/boards in the same chassis
- between chassis in the same location or platforms
- between platforms in a network-centric sensor application

One example of a simple VRT application is shown in Figure 1.3-1 in which a receiver translates an RF signal from an antenna to a lower IF frequency and digitizes it. The digitized signal is sent to a signal processor.

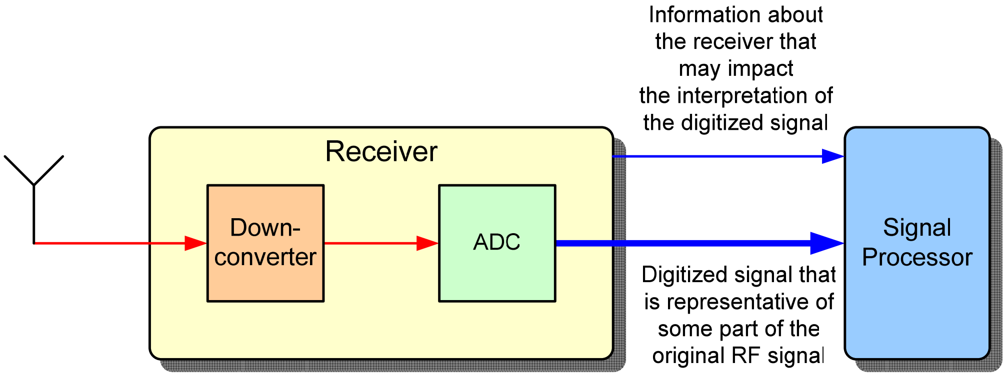


Figure 1.3-1: A simple VRT application

VRT is tailored to transport and describe signals in RF applications

In this example, VRT provides a standard data format for the ADC samples to be sent to the signal processor. VRT also provides a standardized means of conveying information, such as equipment settings, that helps give context to the signal data. For example, VRT defines a standardized means to convey the sample rate of the ADC and the center frequency of the downconverter. The system designer could replace the receiver or the processor with an equivalent VRT-enabled component with minimal impact to the system.

1.4 *Link and Processing Efficiency*

The emergence of high-speed interconnects enables the transmission of signals such as digitized IF over packet networks. Packetization comes at the expense of some overhead however. The VRT protocol can minimize this overhead by allowing the volume, the format and type of data to be configured for the minimal link utilization. Thus the overhead of the protocol can be configured to be much less than 1% of the overall signal data bandwidth to provide the same efficiency as proprietary implementations.

VRT also supports processor-efficient packets where samples are aligned at 8-, 16-, or 32-bit word boundaries to minimize or even eliminate computations required to unpack the packet payload.

1.5 *Organization of Document*

The remainder of this document is organized as follows:

Section 2:	provides information about VITA and the VITA-49.0 working group
Section 3:	provides reference information and standard VITA terminology
Section 4:	provides an overview of the VRT protocol
Section 5:	provides the requirements for compliance to the VRT standard
Section 6:	provides the rules for VRT data packet streams
Section 7:	provides the rules for VRT context packet streams
Section 8:	provides the rules for VRT information streams, which combine data and context packet streams
Appendix A:	contains an example of VRT information stream and packet stream documentation
Appendix B:	contains examples using VRT constructs
Appendix C:	contains a glossary of terms used in this document

2 VITA Information

Working Group Members

At the time this standard was completed, the Working Group for this standard had the following membership:

Name	E-Mail	Company	Status
William Accolla	<w.accolla@acqiris.com>	Acqiris	O
Larry Corsa	<lcorsa@appsig.com>	Applied Signal Technology	O
Matthew Cottrell	<mjc@appsig.com>	Applied Signal Technology	O
William Hanna	<william.a.hanna@boeing.com>	Boeing	O
Melissa Heckman	<mheckman@bustronic.com>	Bustronic Corporation	O
Ian Stalker	<ian.stalker@curtisswright.com>	Curtiss Wright	O
Jeff Smith	<jsmithoffc@aol.com>	Curtiss Wright	O
Lee Brown	<lee.brown@curtisswright.com>	Curtiss Wright	O
Paul Davis	<pdavis@systran.com>	Curtiss Wright	O
Steve Edwards	<steve.edwards@curtisswright.com>	Curtiss Wright	O
Dick Shaner	<dshaner@drti.com>	Digital Receiver Technology	S
Mary Parker	<mparker@drti.com>	Digital Receiver Technology	S
James Taylor	<james.taylor@drs-ss.com>	DRS Signal Solutions	O
Jerry Palmer	<jerry.palmer@drs-ss.com>	DRS Signal Solutions	O
Richard Sims	<Richard.sims@drs-ss.com>	DRS Signal Solutions	O
Robert Normoyle	<robert.normoyle@drs-ss.com>	DRS Signal Solutions	S
Robert Watson	<robert.watson@drs-ss.com>	DRS Signal Solutions	O
Steve Byrd	<steve.byrd@drs-ss.com>	DRS Signal Solutions	O
Mark Takatz	<mark.takatz@drs-ss.com>	DRS Signal Solutions	O
Ray Blakeman	<ray@echotek.com>	Echotek	O
Conrad Romberg	<cromberg@eclipse.sigint.com>	Eclipse Electronic Systems	S
Elwood Parsons	<elwood.parsons@foxconn.com>	Foxconn Electronics	O
Steve Byrd	<steveb@i-b-s-i.com>	Innovative Business Solutions	O
Chris Kent	<ckent@mc.com>	Mercury Computer Systems	O
Craig Lund	<clund@mc.com>	Mercury Computer Systems	O
Dean Holman	<dean@mc.com>	Mercury Computer Systems	S
Lance Jump	<ljump@mc.com>	Mercury Computer Systems	O
Marc Trepanier	<mtrepanier@mc.com>	Mercury Computer Systems	S
Randy Banton	<rbanton@mc.com>	Mercury Computer Systems	O
Stephen Pereira	<spereira@mc.com>	Mercury Computer Systems	S
M Devlin	<m.devlin@nallatech.com>	Nallatech	O
Tork Clark	<tork.clark@ngc.com>	Northrup Grumman	O
Paul Mesibov	<paul@pentek.com>	Pentek	S
John Walker	<john.walker@pentland.co.uk>	Pentland Systems	O
Dave Sherwood	<dlsherwood@praxiseng.com>	Praxis Engineering Technologies	O
Doug Endo	<dendo@raytheon.com>	Raytheon	O
Aaron Kaiway	<aaron_kaiway@spectrumsignal.com>	Spectrum Signal Processing	S
Andy Reddig	<andyr@tekmicro.com>	TEK Microsystems	O
D Bosworth	<dbosworth@tekmicro.com>	TEK Microsystems	O
Robert Patterson	<rapatter@tycoelectronics.com>	Tyco Electronics	O
Jeff Bateman	<jbateman@vmetro.com>	VMETRO	O
Terje Melsom	<terje@vmetro.no>	VMETRO	O

The following companies/representatives were on the sponsor balloting committee:

Company	Representative
DRS Signal Solutions	Robert Normoyle
DRS Signal Solutions	Richard Simms
Digital Receiver Technology	Richard Shaner
Digital Receiver Technology	Mary Parker
Eclipse Electronic Systems	Conrad Romberg
Mercury Computer Systems	Dean Holman
Mercury Computer Systems	Marc Trepanier
Mercury Computer Systems	Stephen Pereira
Pentek	Paul Mesibov
Spectrum Signal Processing	Aaron Kaiway

When the ANSI Standards Board approved this standard on Xxxx xx, 200x, it had the following membership:

Name	Company

Comments, Corrections and/or Additions

Anyone wishing to provide comments, corrections and/or additions to this proposed standard, please direct them to the task group chair or the draft editor:

Chairperson
Robert Normoyle
DRS Signal Solutions
700 Quince Orchard Road
Gaithersburg, MD, USA
20878
1 301 944 8250
robert.normoyle@drs-ss.com

Editors	
Richard Shaner	Richard Sims
Digital Receiver Technology	DRS Signal Solutions
20250 Century Blvd, Suite 300	700 Quince Orchard Road
Germantown, MD, USA	Gaithersburg, MD, USA
20874	20878
1 301 944 9265	1 301 944 8275
dshaner@drti.com	Richard.Sims@drs-ss.com

The best way to provide corrections and small additions is via marking up the specific pages and faxing them to the chair. For longer additions, the chair prefers to receive textual information via e-mail. If drawings are involved, they should be done with AutoCAD. This document is being prepared with Microsoft Word 2002 for Windows compatible computers.

VSO and Other Standards

Should anyone want information on other standards being developed by the VSO, VME Product Directories, VME Handbooks, or general information on the VME market, please contact the VITA office at the address or telephone number given on the front cover.

Change Bars

Change bars will be used in each revision to indicate modifications from the immediately previous revision.

Draft Summary

This is the preliminary draft of this standard. The original content of this draft standard was presented and agreed upon at the xxx VSO meeting. See the draft history for a summary list of the major changes made to each draft.

Issues & Concerns to be Resolved

Following are some of the issues and concerns that need to be resolved before final approval of this proposed standard:

3 References

The following publications are used in conjunction with this standard. In the event that one of these standards is revised, the revised standard **should** be used unless it conflicts with this standard.

The following standards are available from the VMEbus International Trade Association.
http://www.vita.com

- ANSI/VITA 17.1-2003, Serial Front Panel Data Port specification
- VITA-41.0-200X VXS VME bus Switched Serial Standard specification
- VITA-46.0-200X, VPX specification

The following are available from their respective maintainers:

- RapidIO^{TM} Interconnect Specification, Part 6: 1x/4x LP-Serial Physical Layer Specification
- Global Positioning System Standard Positioning Service Signal Specification, 2nd Edition, June 2, 1995
- National Marine Electronics Association, NMEA 0183, Standard for Interfacing Marine Electronic Devices, Version 3.01, January 1, 2002
- PICMG 2.18, Serial RapidIO specification
- ANSI/IEEE 802 specifications
- IEEE 754-1985 Standard for Binary Floating Point Arithmetic

3.1 *Terminology*

To avoid confusion and to make very clear what the requirements for compliance are, many of the paragraphs in this standard are labeled with keywords that indicate the type of information they contain. These keywords are listed below:

Rule
Recommendation
Suggestion
Permission
Observation

Any text not labeled with one of these keywords should be interpreted as descriptive in nature. These will be written in either a descriptive or a narrative style.

The keywords are used as follows:

Rule <Section>-<number>:

Rules form the basic framework of this draft standard. They are sometimes expressed in text form and sometimes in the form of figures, tables or drawings. All rules shall be followed to ensure compatibility between board and backplane designs. All rules use the "shall" or "shall not" words to emphasize the importance of the rule. The "shall" or "shall not" words are reserved exclusively for stating rules in this draft standard and are not used for any other purpose.

Recommendation <Section>-<number>:

Wherever a recommendation appears, designers would be wise to take the advice given. Doing otherwise might result in poor performance or awkward problems. Recommendations found in this standard are based on experience and are provided to designers to speed their traversal of the learning curve. All recommendations use the "should" or "should not" words to emphasize the importance of the recommendation. The "should" or "should not" words are reserved exclusively for stating recommendations in this draft standard and are not used for any other purpose.

Suggestion <Section>-<number>:

A suggestion contains advice, which is helpful but not vital. The reader is encouraged to consider the advice before discarding it. Some design decisions that need to be made are difficult until experience has been gained. Suggestions are included to help a designer who has not yet gained this experience.

Permission <Section>-<number>:

In some cases a rule does not specifically prohibit a certain design approach, but the reader might be left wondering whether that approach might violate the spirit of the rule or whether it might lead to some subtle problem. Permissions reassure the reader that a certain approach is acceptable and will cause no problems. All permissions use the "may" words to emphasize the importance of the permission. The lower-case "may" words are reserved exclusively for stating permissions in this draft standard and are not used for any other purpose.

Observation <Section>-<number>:

Observations do not offer any specific advice. They usually follow naturally from what has just been discussed. They spell out the implications of certain rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain rules so that the reader understands why the rule must be followed.

4 VRT Overview

VRT is a specification for a transport-layer protocol. This protocol is designed to allow radio signals, and information related to radio signals, to be conveyed in digital form from one system or module to another. The kind of information that this protocol conveys can be divided into three categories:

1. Translated RF signals (Herein called “IF Data”)[*]
2. Information derived from IF Data (Herein called “Extension Data”)
3. Metadata about IF Data or Extension Data (Herein called “Context”)

VRT sets forth rules controlling the structure and function of packets that carry these three types of information. It also specifies a way to associate together all of the different packets that carry the different portions of the information related to a signal (or set of signals). In this specification a transmitted sequence of packets that conveys one portion of this information is referred to as a “VRT Packet Stream,” or sometimes simply as a “Packet Stream.” The collection of Packet Streams needed to convey all the information being conveyed about a signal, or signals, is referred to as a “VRT Information Stream,” or simply as an “Information Stream.”

The relationship between VRT Information Streams and Packet Streams is shown in Figure 4.1-1. It depicts three systems all communicating with a fourth system over a data link, using the VRT protocol.

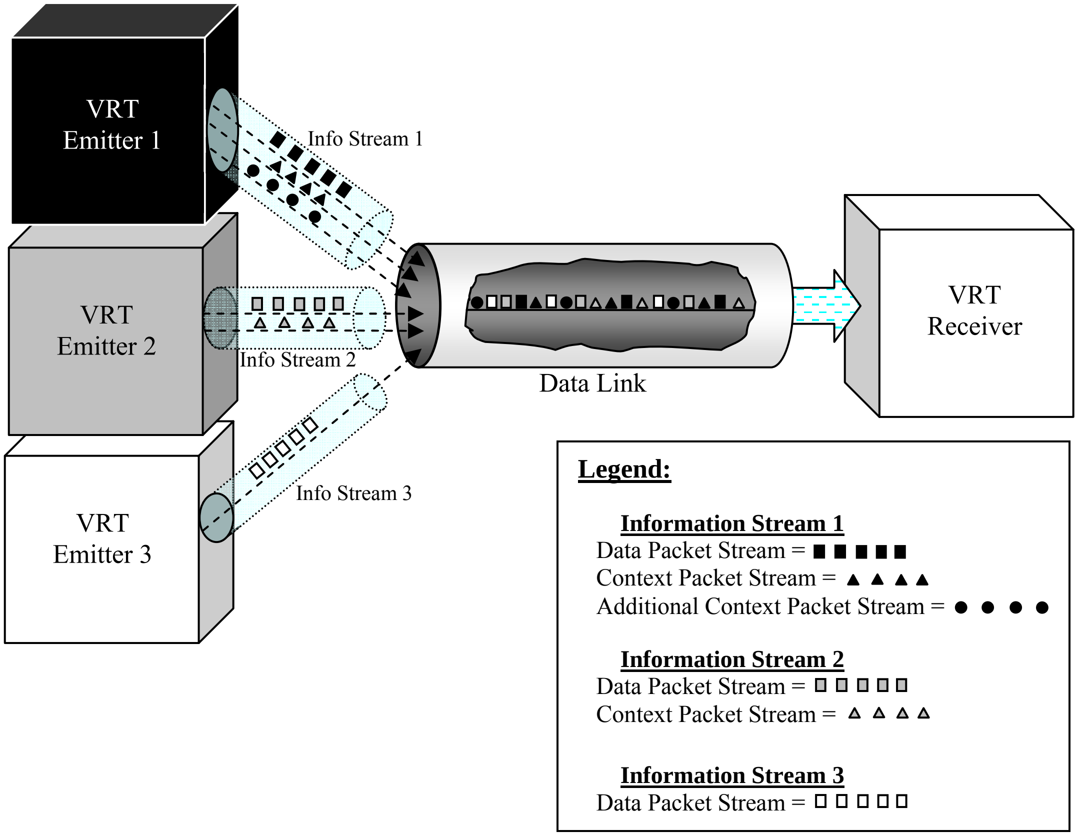


Figure 4.1-1: Three VRT Information Streams on a Data Link. Each Information Stream consists of one or more VRT Packet Streams. Each Packet Stream is dedicated to conveying a particular portion of the information to the VRT receiver.

---

[*] The term “IF Data” should not be taken to imply data that is necessarily in an intermediate state. This digitized data might be anything from baseband to directly sampled RF.

The systems that output VRT packets are called “VRT Emitters,” and the system receiving the VRT packets is called a “VRT Receiver.” By some mechanism, not shown, all the packets from the three Emitters are serialized onto the data link. The VRT Receiver receives these serialized packets.

In this example each emitter outputs one VRT Information Stream. Each Information Stream consists of one or more VRT Packet Streams, and each of these Packet Streams conveys some portion of the total information to be conveyed about a signal. The Figure depicts each Packet Stream by a string of identical shapes, each of which represents a packet. The shape of each packet is related to the kind of information it carries. Square packets carry IF Data, and triangular and round packets carry Context. Each packet’s color indicates to which Information Stream it belongs. The first Emitter outputs three Packet Streams, one for IF Data and two for Context. Each Context Packet Stream conveys some portion of the required context. For example, the first might convey operator settings for the system while the second conveys the model and serial numbers of installed modules. VRT supports putting any number of Context Packet Streams in an Information Stream.

The second Information Stream consists of two Packet Streams, one for IF Data and one for Context. Finally, the third Information Stream consists of a single Packet Stream conveying IF Data.

Definition 4.1-1: A VRT Packet Stream is a sequence of transmitted VRT compliant packets that are all used to convey the same information. A VRT Packet Stream may convey either a signal or information derived from a signal, or metadata about a signal.

Definition 4.1-2: A VRT Information Stream is a collection of VRT Packet Streams, associated together according to VRT rules. In general, each Packet Stream in an Information Stream contains packets of a different type. When taken together, these Packet Streams convey everything that is intended to be conveyed about one or more signals.

As stated, the purpose of the VRT specification is to provide a standard transport layer for the transmission of IF Data, Context, and other types of information that need to be conveyed. It sets forth rules for the creation of packet formats, and rules for the association of all the included Packet Streams in such a way that a VRT Receiver can separate the received Packet Streams into their respective Information Streams. The following sections define terms specific to VRT, and explain in more detail key constructs such as the VRT Information Stream.

4.1 *The VRT Information Stream*

A VRT Information Stream is a set of related VRT “Packet Streams.” Each VRT Packet Stream is a sequence of transmitted VRT compliant packets conveying some specific information related to a signal. For example, one Packet Stream might convey the digitized samples of a signal, while another Packet Stream conveys context such as the RF center frequency, power level, antenna azimuth, etc. Together, the collection of Packet Streams in an Information Stream conveys all the information the application requires.

As an example, Figure 4.1-2 shows a system that uses VRT transport both for inter-module and for inter-system communication. Inside this system the tuner module outputs a VRT Packet Stream conveying only IF Data to the DDC module. This Packet Stream constitutes the simplest possible Information Stream, since it consists of only a single Packet Stream. The DDC module further downconverts a portion of the IF Data it receives, and outputs this in another VRT Packet Stream to the high-speed serial interface. The DDC also outputs a Context Packet Stream along with the IF Data Packet Stream. Together these two Packet Streams constitute another VRT Information Stream. The high speed serial interface receives this Information Stream, and a Packet Stream conveying Context information from the tuner. It combines the tuner Context with the DDC Information Stream to create a new Information Stream containing all the Information it receives. It transmits this new Information Stream to the external signal processor. The VRT protocol associates together all Packet Streams comprising this Information Stream in such a way that the signal processor can sort out all the received packets, and know that all of the intended Packet Streams are being received.

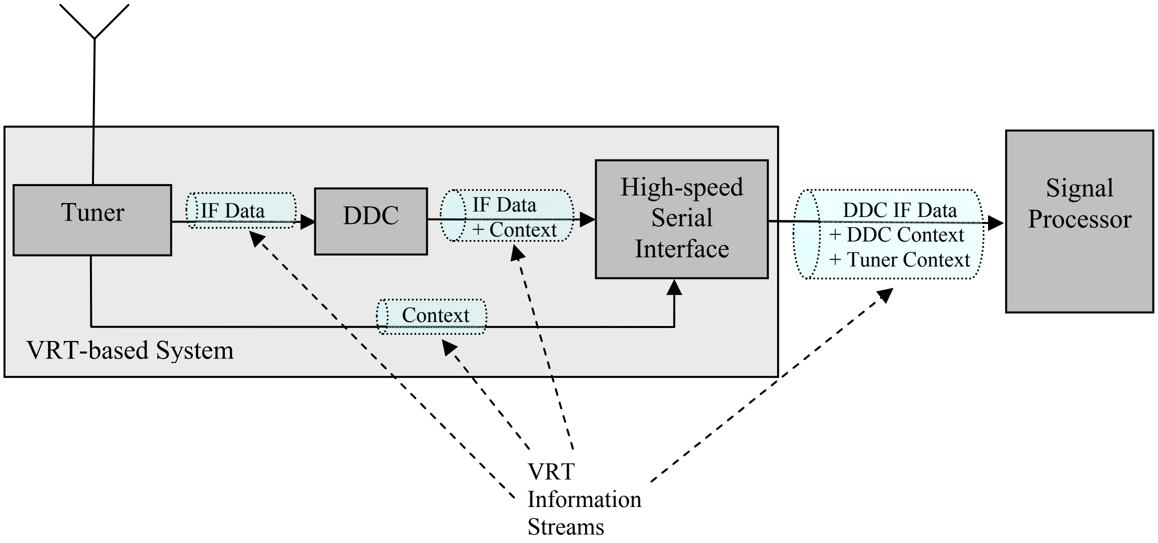


Figure 4.1-2: A System involving four VRT "Information Streams." The tuner outputs one IF Data Information Stream and one Context Information Stream. The DDC outputs one Information Stream consisting of two Packet Streams, one for IF Data and one for Context. The system outputs an Information Stream consisting of three Packet Streams, one for DDC IF Data, one for Context related to the DDC and one for Context related to the tuner.

Every Information Stream is as unique as the application behind it. The simplest Information Stream will contain only a single Packet Stream. However, the VRT standard also supports the creation of Information Streams consisting of an arbitrary number of constituent Packet Streams. Whatever the level of complexity of an Information Stream, the meaning of the constituent Packet Streams is precisely defined, either by rules within this specification, or by documentation that must be provided for VRT compliance.

4.1.1 VRT Packet Streams

An Information Stream may contain one or more VRT Packet Streams for data, or one or more VRT Packet Streams for Context, or both. There are four categories of VRT Packet Streams. They are:

1. IF Data Packet Streams
2. Extension Data Packet Streams
3. IF Context Packet Streams
4. Extension Context Packet Streams

Table 4.1.1-1 shows the relationship between the four categories of Packet Streams. As shown in the table, this specification defines two categories of VRT Packet Streams for conveying data, and two categories of VRT Packet Streams for conveying context. The IF Data Packet Streams and Extension Data Packet Streams convey data, while the IF Context Packet Streams and Extension Context Packet Streams convey Context. There are two types of VRT Packet Streams for data and two types for context in order to facilitate a relatively standard way to convey data and context for IF Data, and yet provide the flexibility required to convey the many other types of data and context that may be present in a VRT system.

Characteristics	Standard Formats	Custom Formats
Conveys Data	**IF Data Packet Stream** Conveys a digitized IF signal • Real/complex data • Fixed/floating point formats • Flexible packing schemes	**Extension Data Packet Stream** Conveys any signal or any data derived from a signal • Any kind of data • Custom packet format
Conveys Context	**IF Context Packet Stream** Conveys context for IF data • Frequency • Power • Timing • Geolocation • Etc.	**Extension Context Packet Stream** Conveys context for any signal or derived data. • Any kind of context • Custom packet format

Table 4.1.1-1: The four categories of Packet Streams. Packet Streams may convey either data or context, and may be either standard or custom in format.

The four types of VRT Packet Streams shown in Table 4.1.1-1 are defined below. The precise requirements for data and Context Packet Streams are given in Section 5 and Section 6 respectively.

Definition 4.1.1-1: An IF Data Packet Stream is a VRT packet stream that conveys IF Data samples, i.e. samples that represent a frequency-translated version of some region of RF spectrum. These samples are taken at a constant sample rate. Examples include:

- The output of an ADC (Analog to Digital Converter) that samples an analog IF signal
- The output of a DDC (Digital Down-Converter) whose input was the ADC just described

The IF Data Packet Stream forms the heart of the VRT Standard. It conveys one or more digitized IF channels.

Definition 4.1.1-2: An Extension Data Packet Stream is a VRT Packet Stream that conveys whatever data the application requires, in a format custom designed for the purpose. In typical VRT applications such data would be derived, directly or indirectly, from a region of RF spectrum, but this is not a requirement. Examples include:

- Pulse-Descriptor Words (PDWs)
- Angle of arrival information
- An FFT of a signal

The Extension Data Packet Stream is intended to extend the applicability of VRT to a wider range of applications than simple IF Data.

Definition 4.1.1-3: An IF Context Packet Stream is a VRT Packet Stream that conveys the most common types of context information related to IF Data signals. This includes:

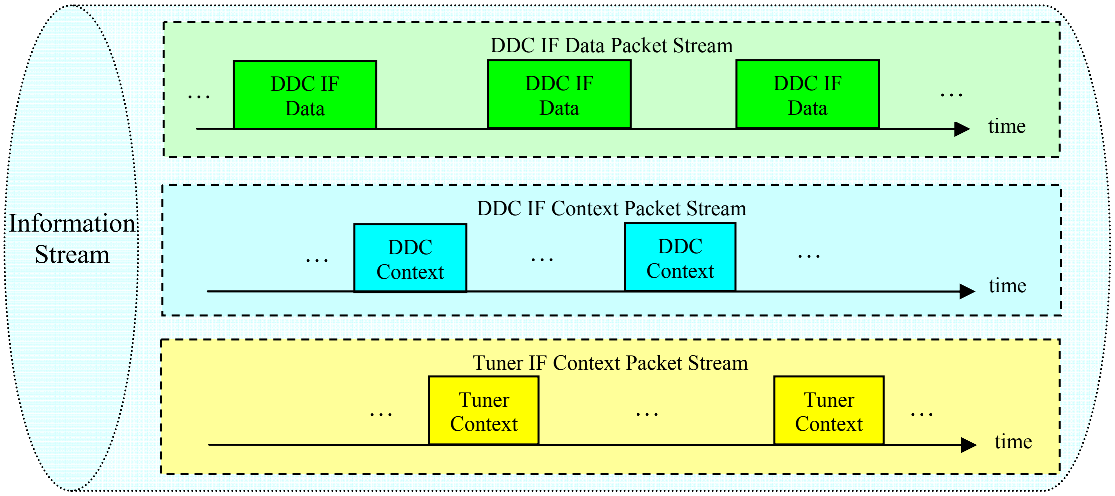
- RF frequency
- Power information
- Timing
- Other details related to the interpretation of the IF Data Packet Streams.

Definition 4.1.1-4: A Extension Context Packet Stream is a VRT Packet Stream that conveys whatever Context the application requires, in a format custom designed for the purpose. Such Context might include:

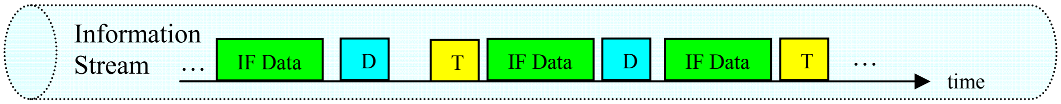
- Vendor-specific information such as equipment serial numbers or operator information
- Application-specific information that cannot be conveyed in an IF Context Packet Stream

4.1.2 Information Stream Structure

In general, a data link may carry an arbitrary number of VRT Information Streams. Each of these Information Streams is an association of some combination of the four types of Packet Streams described in the previous section. For example, in Figure 4.1-1, one IF Data Packet Stream and two IF Context Packet Streams are output from the VRT system to the signal processor. Figure 4.1.2-1 shows the three distinct Packet Streams comprising the single VRT Information Stream in that example.



4.1.2-1a: Information Stream Components.



4.1.2-1b: Information Stream Components Interleaved on a data link.

Figure 4.1.2-1: The components of an example VRT Information Stream.
This Information Stream consists of three VRT Packet Streams, each of a different type (a). The three Packet Streams are interleaved onto a data link (b). Note: In (b), the "IF Data" packets are the DDC IF Data, the "T" packets are the tuner context, and the "D" packets are the DDC context. The same data link may also carry other VRT Information Streams, not shown.

Although these three Packet Streams comprise a single Information Stream, the transmission of the packets within each Packet Stream is independent of the transmission of packets for the other Packet Streams. Typically an IF Data Packet Stream will have a relatively high rate of Packet transmission, while the associated Context Packet Stream has a relatively low rate of Packet transmission.

In order to associate the emitted VRT packets with VRT Packet Streams each emitted packet contains a "Stream Identifier." The Stream Identifier is a unique number assigned to a Packet Stream, and which is embedded in the packets of that Packet Stream. The number is identical in all the packets in the same Packet Stream. Stream Identifiers are also used to associate together all of the Packet Streams relating to an Information Stream. The method of association is explained in detail in Section 6.

In order to fully define an Information Stream, it is necessary to specify all types of packets used in that Information Stream and how the association between the Packet Streams.

4.1.3 Information Stream Specification

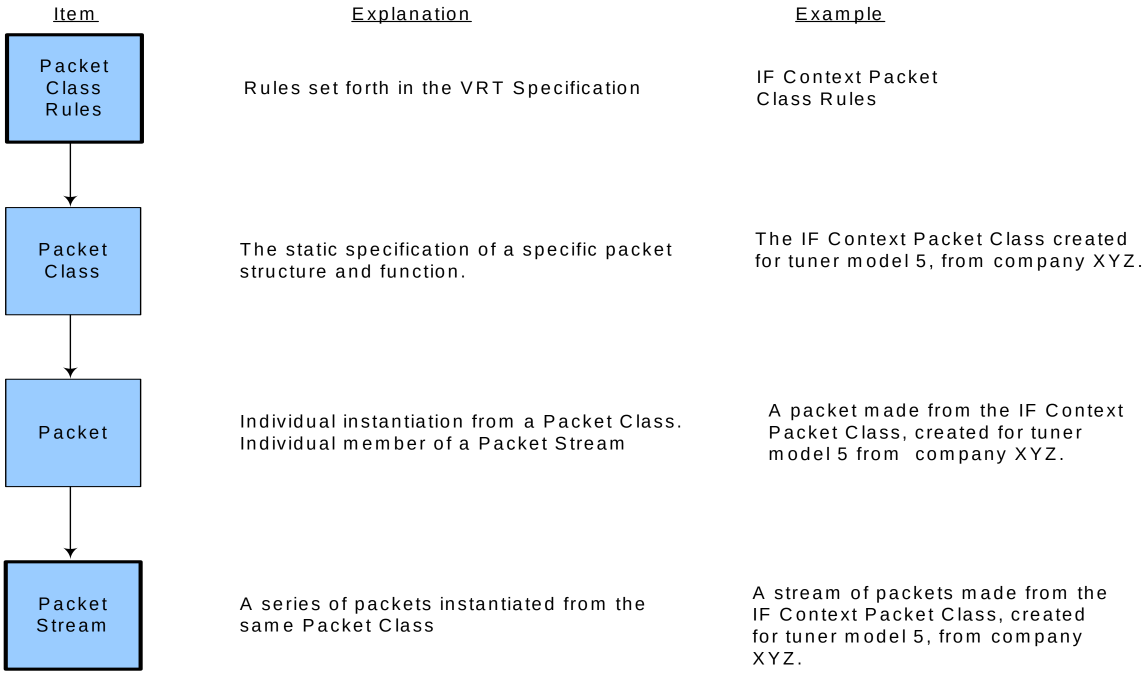
VRT sets forth rules controlling the structure and function of Packet Streams, and Information Streams. It also sets forth rules regarding how the structure and function are to be specified. For the purpose of specifying Packet Streams and Information Streams, VRT borrows somewhat the object-oriented programming concept of a "class." In the context of VRT the term "class" means a specification of the structure and function of VRT objects, namely VRT Packet Streams, and Information Streams. The corresponding classes are called "Packet Classes" and "Information Classes" respectively. A VRT Packet Class defines the structure and function of the Packets which make up a Packet Stream. A VRT Information Class defines the structure and function of the Information Stream. The following sections describe these classes in more detail.

4.1.3.1 Packet Classes

Definition 4.1.3.1-1-1: A VRT Packet Class is the specification of the structure and function of the packets in a VRT Packet Stream. Specifically, it specifies the following:

1. The name of the Packet Class
2. The structure of the Packet Class, i.e. the format and location of each field in the packet.
3. The function of the Packet Class, i.e. the meaning of each field in the packet

All VRT Packet Classes must conform to the rules set forth in this specification. These rules are different for each category of Packet Class. Section 6 presents rules for IF and Extension Data Packet Classes, and Section 7 presents rules for IF and Extension Context Packet Classes. Within the rules for each type of Packet Class, a wide variety of specific Packet Classes may be created, each tailored to the specific application. Each Packet Class may be used for the creation of one or more VRT Packet Streams. Each such Packet Stream will consist solely of packets with the structure and function predefined by the Packet Class from which it originates. Figure 4.1.3.1-1 illustrates the relationship between the VRT specification, a specific, application-dependent Packet Class, and a resulting VRT Packet Stream.



Item	Explanation	Example
Packet Class Rules	Rules set forth in the VRT Specification	IF Context Packet Class Rules
Packet Class	The static specification of a specific packet structure and function.	The IF Context Packet Class created for tuner model 5, from company XYZ.
Packet	Individual instantiation from a Packet Class. Individual member of a Packet Stream	A packet made from the IF Context Packet Class, created for tuner model 5 from company XYZ.
Packet Stream	A series of packets instantiated from the same Packet Class	A stream of packets made from the IF Context Packet Class, created for tuner model 5, from company XYZ.

Figure 4.1.3.1-1: From the VRT specification to a Resulting Application-Dependent Packet Stream.

Figure 4.1.3.1-2 illustrates a VRT Information Stream, consisting of three VRT Packet Streams, on a data link. The three Packet Streams are based on three predefined Packet Classes. Each Class serves as a template for the creation of a packet whenever information specified in that Packet Class needs to be sent.

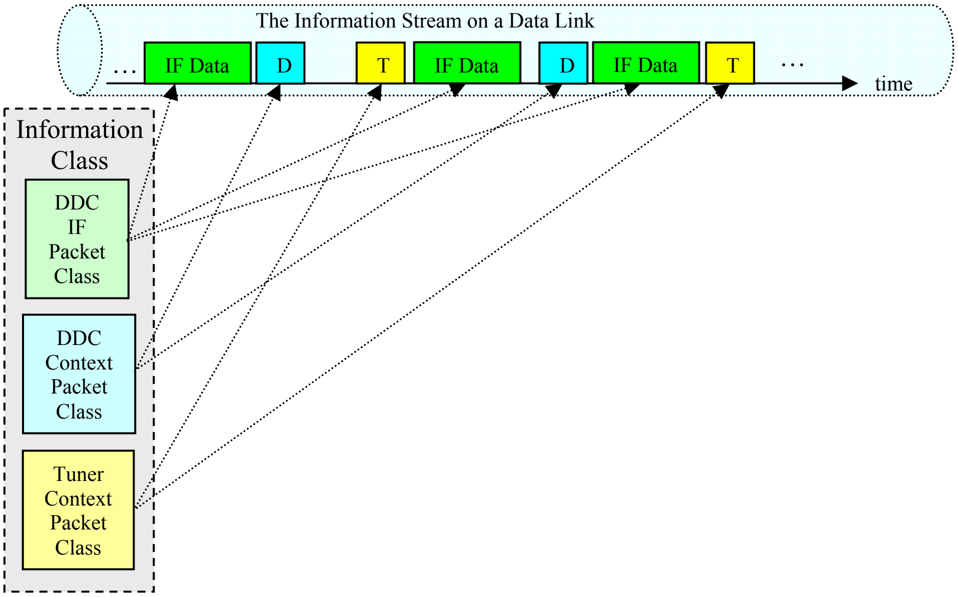


Figure 4.1.3.1-2: Three VRT Packet Streams made from three Packet Classes. Every time a DDC IF Packet is transmitted, it is based on the same Packet Class. Similarly, packets in the other two Packet Streams are based on their corresponding Packet Classes.

It should be noted that a given Packet Class may be used to create multiple Packet Streams within an Information Stream, and may be used for multiple Information Streams. The only requirement is that every packet that is created from the one Packet Class must have the same structure and function. For example, the same IF Data Packet Class may be used for *N* different output channels from a bank of DDCs. In this case the data format, data packing, etc. for the packets are all identical. The only difference from one of these Packet Streams to another is that each of the *N* Packet Streams carries data for a different channel.

4.1.3.2 Information Classes

An Information Class is a specification of the structure and function of an Information Stream. This specification includes the Packet Classes for the included Packet Streams, as depicted in Figure 4.1.3.1-2. In that figure the three Packet Classes are grouped together in one box, labeled "Information Class," because they define parts of the larger specification. In addition to the Packet Classes, the Information Class also specifies several other details of the Information Stream. Altogether there are seven components to an Information Class. They are:

1. Class Name
An Information Class is given a name as an identifier. This name does not appear in the resulting Information Stream It is a name used by system designers to refer to the type of Information Stream being created.

2. Information Stream Purpose
The purpose is the reason for which the resulting Information Stream is to be created. It may be very general, such as, "To convey any 70 MHz IF Data," or may be more specific, such as, "To convey FFT data from product X to product Y in system Z." In general, Information Streams with more constituent Packet Streams are likely to have more specific statements of purpose.

3. Names of included VRT Packet Streams
Packet Streams are also given names to be used as identifiers. These names do not appear in the Information Stream, but are used by system designers to refer to the Packet Streams within an Information Stream. It is often convenient to choose a name that indicates the purpose of the Packet Stream. These names have meaning only within the context of an Information Stream. They are reused from one Information Stream to another when multiple Information Streams are created from the same Information Class.

4. Purpose of each included Packet Stream
The purpose of a VRT Packet Stream is to convey some particular information that is part of the Information Stream. This purpose is more specific than the purpose stated for the corresponding Packet Class, which may support multiple Packet Streams, and which generally only specifies the type of information conveyed rather than the particular information conveyed. For example, if the purpose of a Packet Class is "to convey IF Data in real 16-bit samples," the purpose of a resulting Packet Stream might be, "to convey the IF Data in real 16-bit samples from tuner model 5 made by company XYZ."

5. Packet Classes
Each Packet Class specifies the structure and function of a type of Packet Stream, as previously described. For each type of Packet Stream in the Information Stream, a Packet Class is provided for use in creating that Packet Stream. The same Packet Class may be used for the creation of multiple Packet Streams.

6. Context Reference Points
Some of the information conveyed in IF Context Packet Streams expresses frequency, power, or timing information related to a particular place in the processing system. The locations of these reference points are specified in the Information Class.

7. Packet Stream Association

The Packet Stream association for an Information Class specifies the relationship between Packet Streams within the Information Stream. This association reflects the relative purpose of each Packet Stream. For example, an IF Data Packet Stream and a corresponding IF Context Packet Stream should be associated because the IF Context Packet Stream provides metadata for the IF Data Packet Stream. The association specification may take the form of a diagram, or of an outline-like text description. In addition to being documented in the Information Class, this association is communicated in the Information Stream by fields in the packets that hold Stream Identifiers for other Packet Streams. The Stream Identifier mechanism for conveying the association between Packet Streams is fully described in Section 6.

The seven components of an Information class are demonstrated in the following example: Figure 4.1.3.2-1 shows a system that outputs a non-trivial Information Stream. This Information Stream consists of an IF Data Packet Stream and an IF Context Packet Stream from the adaptive combiner, plus a separate IF Context Packet Stream from each of the tuners and from each of the DDCs.

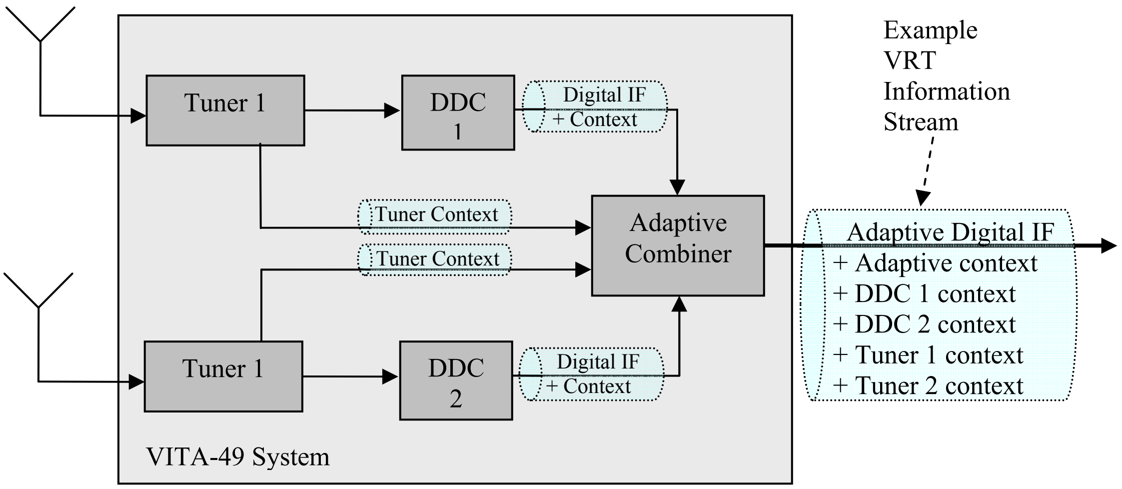


Figure 4.1.3.2-1: An example system outputting a non-trivial Information Class. The IF Data output of the adaptive combiner is transported over the link along with context from the adaptive combiner and from each of the four upstream modules.

The definition of this Information Stream, i.e. its Information Class, consists of the following:

1. The Name of the Information Class:
   "Adaptive Combiner Information Class"

2. The purpose of the Information Stream:
   "To convey IF Data and Context from this example system"

3. The names of the included Packet Streams:
   - Adaptive Combiner IF Data Packet Stream
   - Adaptive Combiner IF Context Packet Stream
   - DDC-1 IF Context Packet Stream
   - DDC-2 IF Context Packet Stream
   - Tuner-1 IF Context Packet Stream
   - Tuner-2 IF Context Packet Stream

4. The Purpose of each included Packet Stream:
   - To convey Adaptive Combiner IF Data

- To convey Adaptive Combiner Context
- To convey DDC-1 Context
- To convey DDC-2 Context
- To convey Tuner-1 Context
- To convey Tuner-2 Context

5. A Packet Classes from which the Packet Streams are made Packet Stream:
    - Example Adaptive Combiner IF Data Packet Class
    - Example Adaptive Combiner IF Context Packet Class
    - Example DDC IF Context Packet Class (Used twice, once for each DDC)
    - Example Tuner-1/Tuner-2 IF Context Packet Class (Used twice, once for each tuner)

6. Context Reference Points.
    - Tuner Context always references the tuner input.
    - DDC Context always references the tuner input.
    - Adaptive Combiner Context always references the tuner input.

7. A specification of the Packet Stream Association.
    - Figure 4.1.3.2-2 shows an association diagram for this example. The arrows connecting the Packet Streams indicate which ones are directly associated with each other. The collection of these pair-wise associations also serves to associate all of the Packet Streams with the Information Stream.

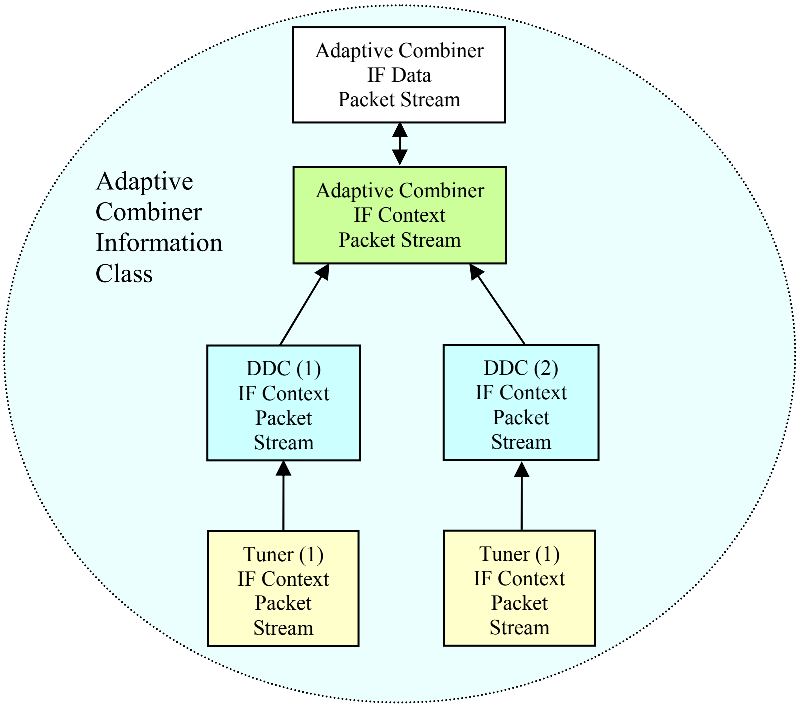


Figure 4.1.3.2-2: The Association of Packet Streams within the example Information Stream. The arrows between the streams indicate which streams are directly associated.

Note that the association diagram bears a strong resemblance to the architecture of the system. This is because most of the associations in this example are indications of signal flow within the system. The arrows indicate the direction of signal flow and the system. There is one exception to this rule. The arrow between the Adaptive combiner IF Data and Context Streams is bidirectional. This represents a special type of association that can only occur between a Data and a Context Packet Stream. It does not relate at all to signal flow. This will be explained in Section 6. For now we simply state that exactly one Context Packet Stream can have this special association to a Data Packet Stream. The purpose is to create a unique association between the Data Packet Stream and the Context Packet Stream that is from the device/process that actually produces the data.

Associations between Packet Streams may be based on criteria other than signal flow. Section 6 explains the other types supported. The provision for several types of association, and the conveyance of these associations in the transmitted packets, facilitates auto-discovery of the architecture of a VRT system by the receiver of the Information Stream.

5 Compliance

There are two requirements that must be met for an Information Stream to comply with this specification. The first requirement is that the structure and function of the Information Stream comply with all of the rules in this specification governing the structure and function of Information Streams. The second requirement is that the Information Stream be documented by an Information Class according to the rules in this specification governing Information Classes. The determination of Information Stream compliance is to be made based on its Information Class, which provides all the specifications for building the Information Stream.

Rule 5-1: A VRT Information Stream **shall** comply with all the rules in this specification governing the structure and function of Information Streams.

Rule 5-2: A VRT Information Class **shall** provide all of the defining details for the Information Stream, as required by this specification.

6 Data Packet Classes and Streams

This Section sets forth rules controlling the structure of Data Packet Classes and Data Packet Streams. A Data Packet Class is the specification of a structure and function for a packet type that conveys some particular type of data. As explained in Section 4, VRT defines two categories of Data Packet Classes, one for IF Data, one for Extension Data. Within each category, any number of Data Packet Classes may be created, each with its own unique choice of parameters, such as data type and data packing, Timestamp type, etc. The following sections specify requirements and restrictions for these Packet Classes, and for their documentation.

The conveyance of IF Data is the primary focus of the VRT specification. Therefore, IF Data Packet Streams are considered the backbone of this specification. These Packet Streams are considered to be the ones most affecting interoperability, so these Packet Streams are the ones most tightly controlled. Nevertheless, the rules for IF Data Packet Classes and Streams allow for a wide range of variability from one IF Data Packet Class to another. This feature allows a diversity of applications to be brought together under the umbrella of VRT.

Many applications involving IF Data also involve one or more forms of “derived data.” Derived data is data that is somehow extracted from a signal. For example, a demodulated bit-stream from a PSK carrier would be considered derived data. This data takes many forms, so it requires more flexibility to convey it than is provided for in IF Data Packet Streams. Therefore this specification also sets forth rules for “Extension Data Packet Streams.” These Packet Streams are intended to convey all of the other types of data that cannot be conveyed in an IF Data Packet Stream. As a result, the rules for Extension Packet Classes and Streams are more relaxed.

The following two rules are set forth in order to promote the highest possible level of interoperability given the diverse nature of applications involving IF Data signals:

Rule 6-1: An IF Data Packet Class **shall** be used only to convey IF Data.

Rule 6-2: An Extension Data Packet Class **shall** be used only to convey data that cannot be conveyed in an IF Data Packet Class.

Section 6.1 presents rules for IF Data Packet Classes and Streams, and Section 6.2 presents rules for Extension Data Packet Classes and Streams.

6.1 *IF Data Packet Classes and Streams*

This section sets forth rules controlling the structure and function of all IF Data Packets specified by an IF Data Class and used in an IF Data Packet Stream. Although these rules allow for substantial variability, all IF Data Packet Classes and Streams have certain structures and functions in common. The common top-level format to be used for every IF Data Packet is shown in Figure 6.1. The options and rules for the fields shown are explained in detail in the following sections.

Rule 6.1-1: The order of the fields in an IF Data Packet **shall** be as shown in Figure 6.1.

Rule 6.1-2: If an optional field is not present in an IF Data Packet Class, the remaining words in the packet **shall** “move up” toward the header, with no padding.

Rule 6.1-3: If an optional field is present in an IF Data Packet Class, but is omitted in a transmitted packet, the remaining words in that packet **shall** “move up” toward the header, with no padding.

Observation 6.1-1: Rule 6.1-3 pertains to fields specified by the Packet Class as optional in the transmitted packets. For example, the Manufacturer field might be specified as present in only every 10th packet.

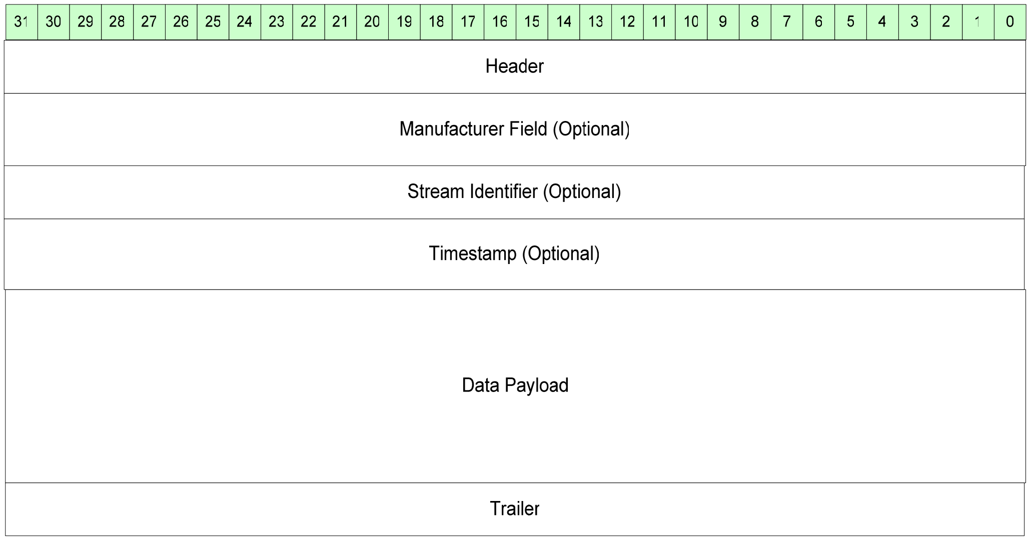


Figure 6.1: The IF Data Packet Class Format
(The first 32-bit word in the packet is the one at the top.)

6.1.1 The IF Data Packet Header

Rule 6.1.1-1: Every VRT Data Packet **shall** have a mandatory header having the format shown in Figure 6.1.1-1.

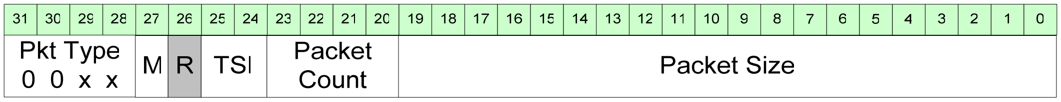


Figure 6.1.1-1: IF Data Packet Header

Rule 6.1.1-2: The IF Data Packet Header **shall** contain the following fields:

Packet Type – Defines the type of packet. Table 6.1.1-1 shows the types of VRT packets, and corresponding packet codes. As shown, valid IF Data Packet codes are “0001” and “0000” for packets with and without Stream Identifiers respectively.

Rule 6.1.1-3: The “Packet Type” field **shall** accurately indicate the type of packet, as specified in Table 6.1.1-1. Every packet in a Packet Stream **shall** be of the same type.

Packet Type	Meaning
0000 (0)	IF Data Stream Packet without Stream Identifier
0001 (1)	IF Data Stream Packet with Stream Identifier
0010 (2)	Extension Data Stream Packet without Stream Identifier
0011 (3)	Extension Data Stream Packet with Stream Identifier
0100 (4)	Standard Context Packet (see section 6)
0101 (5)	Extension Context Packet (see section 7)
Others	Reserved for future VRT Packet Types

Table 6.1.1-1: The assignment of Packet Type codes

“M” bit – This single-bit field indicates whether the Manufacturer field is included in a packet. Each packet in an IF Data Packet Stream may or may not include the Manufacturer field.

Rule 6.1.1-4: The M bit **shall** be set to one in IF Data Packets that include the Manufacturer field. It **shall** be set to zero in IF Data Packets that do not include the Manufacturer field.

Reserved – These bits are reserved for future use in VRT.

Rule 6.1.1-5: The reserved bit **shall** be set to 0.

Timestamp Indicator (TSI) – This is an encoded field, indicating which, if any, Timestamp is present in the packet. Table 6.1.1-2 shows the type of Timestamp corresponding to valid TSI codes.

Rule 6.1.1-6: The TS field in an IF Data Packet **shall** accurately indicate the type of Timestamp included the packet, according to the code assignments in Table 6.1.1-3.

Rule 6.1.1-7: All the packets in an IF Data Packet Stream **shall** have the same TSI code.

TSI code	Meaning
00	No Timestamp field included
01	UTC plus Higher Precision “Sample Count” Time Code
10	UTC plus Higher Precision “Picoseconds” Time Code
11	UTC plus Higher Precision “Free Running Count” Time Code

Table 6.1.1-2: The assignment of Timestamp codes

Packet Count – This field contains a modulo-16 count of IF Data Stream packets for a stream. The least-significant bit (lsb) of the count is the right-most in the field.

Rule 6.1.1-8: The Packet count **shall** increment in each consecutive IF Data Packet having the same Stream Identifier and the same packet type. The count **shall** roll over from “1111” to “0000.”

Packet Size – This 20-bit field indicates the total number of 32-bit words present in the IF Data Packet, including the header, trailer and any optional fields.

Rule 6.1.1-9: In each packet, the Packet Size field **shall** correctly indicate the total number of 32-bit words in the packet.

Observation 6.1.1-1: Every IF Data Packet consists of an integer number of 32-bit words less than 2^{20} (1,048,576). This size is the sum of the sizes of all the included fields in the packet. The packet size may vary from packet to packet.

6.1.2 Manufacturer Field

The manufacturer field provides information regarding the identity of the manufacturer and the Packet Class from which the packet was made. Figure 6.1.2-1 shows the contents of the Manufacturer field. This field contains two subfields. The first subfield contains the Organizationally Unique Identifier (OUI). This is an IEEE-assigned 24-bit number which indicates the identity of the manufacturer of the equipment generating the IF Data Packet Stream. The second field contains the Class Code. The Class Code indicates to which of that manufacturer’s Packet Classes the transmitted packet belongs. This allows the format of the packet to be determined from a table, based on the combination of the OUI and Class Code.

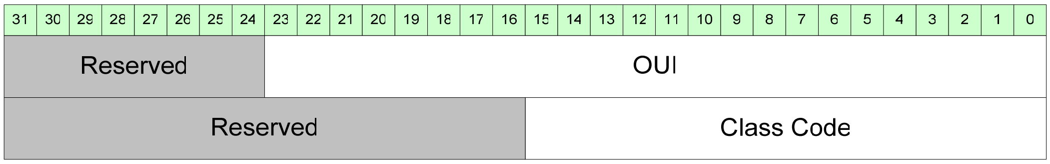


Figure 6.1.2-1: The Contents of the Manufacturer Fields.

Rule 6.1.2-1: When the manufacturer field is present in an IF Data Packet, the OUI field **shall** contain the OUI of the manufacturer of the equipment that emitted the packet.

Rule 6.1.2-2: When the manufacturer field is present in an IF Data Packet, the Class Code field **shall** contain a manufacturer-assigned code indicating the Packet Class from which the packet was created. A value of zero in this field **shall** indicate that the Packet Class is unspecified.

Rule 6.1.2-3: All of the reserved bits in the Manufacturer Field **shall** be set to zero.

6.1.3 Stream Identifier

The Stream Identifier, or simply Stream ID, is a unique number associated with a VRT Packet Stream. Every packet in the Packet Stream has the same Stream ID.

Rule 6.1.3-1: The Stream Identifier Field shall be either included in, or omitted from, every IF Data Packet in an IF Data Packet Stream, as described in Table 6.1.1-1.

Rule 6.1.3-2: When an IF Data Packet Stream includes a Stream Identifier Field, all IF Data Packets in that Packet Stream **shall** use the same Stream Identifier in that field.

Rule 6.1.3-3: Different IF Data Packet Streams **shall not** use the same Stream Identifier.

6.1.4 Timestamp

The Timestamp in an IF Data Packet indicates the time associated with the first data sample contained in the packet. This specification provides for three distinct types of Timestamps. Irrespective of the type of Timestamp used, the Timestamp field, when present, occupies three 32-bit words. The first word contains a Universal Time Code (UTC), which gives the time to one-second precision. The remaining two words contain a 64-bit higher-precision time code. The first two forms of the Higher Precision Time Code add resolution to the UTC, providing a combined precision down to either sample-period or picosecond precision respectively. The third form provides an incrementing sample count from an arbitrary point in time.

Rule 6.1.4-1: When present, the Timestamp in an IF Data packet **shall** represent the time that the information represented by the first data sample in the Data Payload was present at the reference point. This rule **shall** hold unless a “Timestamp Adjustment” has been sent in an associated Context packet, in which case the indicated time **shall** be adjusted as described in Section 7.2.5.11.

Observation 6.1.4-1: In a typical application, the Timestamp would give the ADC conversion time, and the Timestamp Adjustment field in the Context Packet would adjust this time so that it reflects when the information represented by the sample arrived at some upstream point, such as an antenna. (Move this to Section 7? –Dick)

Rule 6.1.4-2: The Timestamp **shall** consist of one 32-bit word for the UTC, plus a 64-bit field (two consecutive 32-bit words) containing only one of the three possible Higher Precision Time Codes.

The three types of Timestamps are described in the following sections. All have a subfield for the UTC.

6.1.4.1 The Universal Time Code (UTC)

Rule 6.1.4.1-1: When used, the UTC shall be a 32-bit unsigned integer representing time in seconds since 00:00:00Z, January 1, 1970.

6.1.4.2 Sample Count Timestamp

The Sample Count Timestamp consists of a mandatory UTC followed by a mandatory 64-bit "Sample Counts" field. The Sample Counts field contains the sample number of the first data sample in the IF Data Stream Packet relative to the time of the last UTC increment. The Sample Counts field is reset to zero on each increment of the UTC.

Figure 6.1.4.2-1 shows the components of the Sample Count Timestamp, in their relative positions in the field.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Universal Time Code</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Sample Counts (MSW)</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Sample Counts (LSW)</td></tr>
</table>

Figure 6.1.4.2-1: The Format of the Sample Count Timestamp
(The top word shown occurs first in the field.)

Rule 6.1.4.2-1: The Sample Count Timestamp **shall** contain a UTC in the first 32-bit word of the field. The lsb of the word **shall** be on the right.

Rule 6.1.4.2-2: The Sample Count Timestamp **shall** contain a Higher Precision "Sample Count" Time Code in the last two 32-bit words in the field. The most significant 32 bits **shall** be in the first of these two words. The lsb of each word **shall** be on the right.

Rule 6.1.4.2-3: The "Sample Count" Time Code **shall** be a 64-bit unsigned binary integer indicating the time of the first data sample in the data packet, in sample counts, relative to the time of the most recent UTC increment.

Rule 6.1.4.2-4: The Higher Precision Sample Count Time Code **shall** be reset to zero on each increment of the UTC.

6.1.4.3 Real-Time Timestamp

The Real-Time Timestamp consists of a mandatory UTC followed by a mandatory 64-bit "picoseconds" field. The picoseconds field contains the time, in picoseconds, of the first data sample in the IF Data Stream packet relative to the time of the last UTC increment. The picoseconds field is reset to zero on each increment of the UTC.

Figure **6.1.4.3**-1 shows the components of the Real-Time Timestamp, in their relative positions in the field.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Universal Time Code</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Picoseconds (MSW)</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Picoseconds (LSW)</td></tr>
</table>

Figure 6.1.4.3-1: The Format of the Real-Time Timestamp
(The top word shown occurs first in the field.)

Rule 6.1.4.3-1: The Real-Time Timestamp **shall** contain a UTC in the first 32-bit word of the field. The lsb of the word **shall** be on the right.

Rule 6.1.4.3-2: The Real-Time Timestamp **shall** contain a Higher Precision "Picoseconds" Time Code in the last two 32-bit words in the field. The most significant 32 bits **shall** be in the first of these two words. The lsb of each word **shall** be on the right.

Rule 6.1.4.3-3: The Picoseconds Time Code **shall** be a 64-bit unsigned binary integer indicating the time of the first data sample in the data packet, in picoseconds, relative to the time of the most recent UTC increment.

Rule 6.1.4.3-4: The Higher Precision Picoseconds Time Code **shall** be reset to zero on each increment of the UTC.

6.1.4.4 Free Running Count Timestamp

The Free Running Count Timestamp consists of an optional UTC followed by a mandatory 64-bit "Free Running Count" field. The Free Running Count field contains a free running sample counter. The Free Running Counter rolls over modulo-*N*, i.e. from *N*-1 to zero, where *N* can be any positive number up to 2^{64}. The Free Running Count has no constant relationship to the UTC.

Figure 6.1.4.4-1 shows the components of the Free Running Count Timestamp, in their relative positions in the field.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Universal Time Code (Optional)</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Free Running Count (MSW)</td></tr>
<tr><td colspan="32">Higher Precision Time Code, Free Running Count (LSW)</td></tr>
</table>

Figure 6.1.4.4-1: The Format of the Free Running Count Timestamp
(The top word shown occurs first in the field.)

Rule 6.1.4.4-1: The first 32-bit word of the Free Running Count Timestamp **shall** contain either a UTC or all ones. The lsb of the word shall be on the right.

Rule 6.1.4.4-2: The Free Running Count Timestamp **shall** contain a Higher Precision "Free Running Count" Time Code in the last two 32-bit words in the field. The most significant 32 bits **shall** be in the first of these two words. The lsb of each word **shall** be on the right.

Rule 6.1.4.4-3: The Free Running Count Time Code **shall** be a 64-bit unsigned binary integer indicating the time of the first data sample in the data packet, in sample counts, modulo-*N*, relative to the counter starting time.

Rule 6.1.4.4-4: This Higher Precision Time Code is free running, and **shall not** reset to zero except as the free running sample count rolls over modulo-*N*, where *N* may be any positive number up to 2^{64}.

Rule 6.1.4.4-5: If the Free Running Count Time Code is used, the originating Packet Class **shall** specify the modulus, *N*, for the free running counter.

6.1.5 Data Payload

The data payload of an IF Data Packet contains a contiguous subsequence of the data samples in an IF Data sample stream. The number of words is flexible, and is derivable from the Packet Type, Packet Size, M bit, and Timestamp indicator in the Header of the IF Data Packet.

Rule 6.1.5-1: The Maximum number of data payload words **shall** be 2^{20}-1, minus the number of words required for the Header, Stream ID, Timestamp, and Trailer fields.

Rule 6.1.5-2: The data payload **shall** consist of an integer number of contiguous 32-bit words.

Rule 6.1.5-3: If the number of payload bits to pack is not an integer multiple of 32 bits, then the payload size, in bits, **shall** be rounded up to the nearest multiple of 32 bits.

Permission 6.1.5-1: The size of the data payload may vary from packet to packet as long as the resulting size of each IF Data Packet is accurately described by the Packet Size field in the Header of that Packet.

Rule 6.1.5-4: Payloads in IF Data packets **shall** be packed in accordance with the rules set forth in the following sections. The originating Packet Class shall specify how the data is packed.

6.1.5.1 Item Packing Fields and Their Contents

In order to facilitate a wide range of payload packing methods, and at the same time provide some commonality to these methods, VRT makes use of the concept of an Item Packing Field. An Item Packet Field is a virtual container that can contain multiple pieces of data. These pieces of data are defined below. When included in an Item Packing Field, they are always packed in a certain order, also defined below.

Definition 6.1.5.1-1: A Data Item is a binary number representing a portion of the information to be conveyed in the payload. This number may be a real-valued sample, or a real or imaginary component of a complex Cartesian sample, or an amplitude or phase component of a polar complex sample.

Definition 6.1.5.1-2: An Event Tag is a bit used to indicate that a signal-related or processing-related event has occurred coincident with a Data Item in the payload.

Definition 6.1.5.1-3: A Channel Tag is a label associating a Data Item with a particular channel conveyed by an IF Data Packet.

Definition 6.1.5.1-4: An Item Packing Field is a virtual container for a Data Item, Event Tags, and a Channel Tag.

Figure 6.1.5.1-1 shows the relative location of the Data Item, unused bits, if any, Event Tags, and Channel Tag in an Item Packing Field. Rules for placing these items in an Item Packing Field follow.

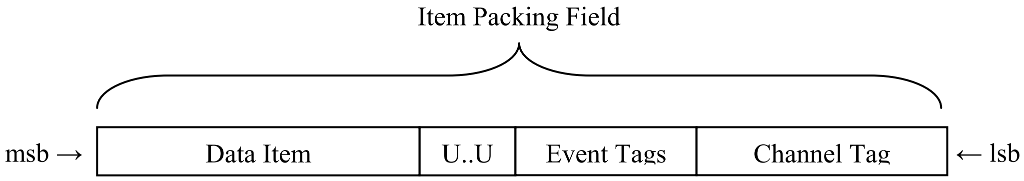


Figure 6.1.5.1-1: Relative locations of items within the Item Packing Field

Rule 6.1.5.1-1: The Data Payload of an IF Data Packet Class **shall** be populated by a sequence of "Item Packing Fields." The Item Packing Fields shall be from one to 64 bits in size. All Item Packing Fields specified by an IF Data Packet Class **shall** be the same size.

Rule 6.1.5.1-2: An Item Packing Field **shall** contain exactly one Data Item. The Data Item **may** be from one to 64 bits in size, and **shall** be left-justified in the Item Packing Field. All Data Items specified by an IF Data Packet Class **shall** be the same size.

Observation 6.1.5.1-1: Rule 6.1.4.1-2 applies even when Polar complex samples are used. That is, the Data Item sizes for the amplitude and phase components are required to be the same, though their meanings are very different.

Rule 6.1.5.1-3: An Item Packing Field **shall** contain either zero or one Channel Tag(s). The channel tag, when present, **may** be from one to 15 bits in size. The Channel Tag **shall** be right-justified in the Item Packing Field. All Channel Tags specified by an IF Data Packet Class **shall** be the same size.

Rule 6.1.5.1-4: An Item Packing Field **shall** contain either zero or one Event Tag field(s). The Event Tag field, when present, **may** be from one to seven bits in size. The Event Tag field **shall** be justified in the Item Packing Field just to the left of the Channel Tag. All Event Tags specified by an IF Data Packet Class **shall** be the same size.

Recommendation 6.1.5.1-1: The precise meaning of the Event Tag bits is not specified. However, the encodings **should** be chosen such that the Event Tag field contains all zeros when no events have occurred.

Rule 6.1.5.1-5: If the Event Tag field does contain application-specific metadata, then the originating Packet Class **shall** specify what this data is, and how it is formatted.

Rule 6.1.5.1-6: When an event occurs, and is indicated by some bit pattern in an Event Tag field, the Item Packing Field containing this non-zero bit pattern **shall** be the one containing the Data Item most closely associated with the time of the event.

Rule 6.1.5.1-7: The size of the Item Packing Field **shall** be at least equal to the sum of the sizes of the Data Item, the Event Tag field, and the Channel Tag. The Item Packing Field size **shall not** exceed 64 bits. Therefore, this sum also **shall not** exceed 64 bits.

Rule 6.1.5.1-8: The size of the Item Packing Field **may** be larger than required to contain the included items. In this case the unused (U) bits **shall** reside immediately to the right of the Data Item.

Observation 6.1.5.1-1: Although the maximum Data Item size is 64 bits, it can only be this large if there are no Event Tags and no Channel Tag included.

Rule 6.1.5.1-9: The originating Packet Class **shall** specify the Item Packing Field size, the Data Item size, the Event Tag field size, and the Channel Tag field size.

6.1.5.2 The Arrangement of Item Packing Fields in 32 Bit Words

This specification requires Item Packing Fields to be packed into 32-bit words in the payload section according to either of two methods. The first method is called “Processing-Efficient” packing, and the second is called “Link-Efficient” packing. Figure 6.1.5.2-1 shows an example of each for an Item Packing Field size of 15 bits.

Rule 6.1.5.2-1: Item Packing Fields **shall** be arranged in the 32-bit wide payload region either according to Processing-efficient rules, or according to Link-efficient rules. All payloads in an IF Data Packet Stream **shall** be packed identically.

Rule 6.1.5.2-2: For Processing-efficient packing, the number of Item Packing Fields per 32-bit word **shall** be the maximum number that will fit into a 32 bit word. In the case where less than 32 bits are required, Item Packing Fields **shall** be left-justified, so that all unused bits reside at the right (lsb) end of each 32-bit word. These unused bits **should** be set to zero. They **shall** be ignored by VRT compliant receivers. For Processing-efficient packing, every 32-bit word **shall** be packed identically.

Rule 6.1.5.2-3: For Link-efficient packing, all bits in each 32-bit word **shall** be used for the packing of Item Packing Fields, with the possible exception of the last 32-bit payload word, which **may** be only partially filled. Each consecutive Item Packing Field **shall** reside immediately to the right of the previous one. When the remaining bits in a 32-bit word are fewer than the size of an Item Packing Field, the left-most bits of the next Item Packing Field **shall** be loaded into the available bit positions. The remaining portion of such an Item Packing Field **shall** occupy the left-most positions of the next 32-bit word. In this way, Item Packing Fields are said to “wrap” from one 32-bit word to the next.

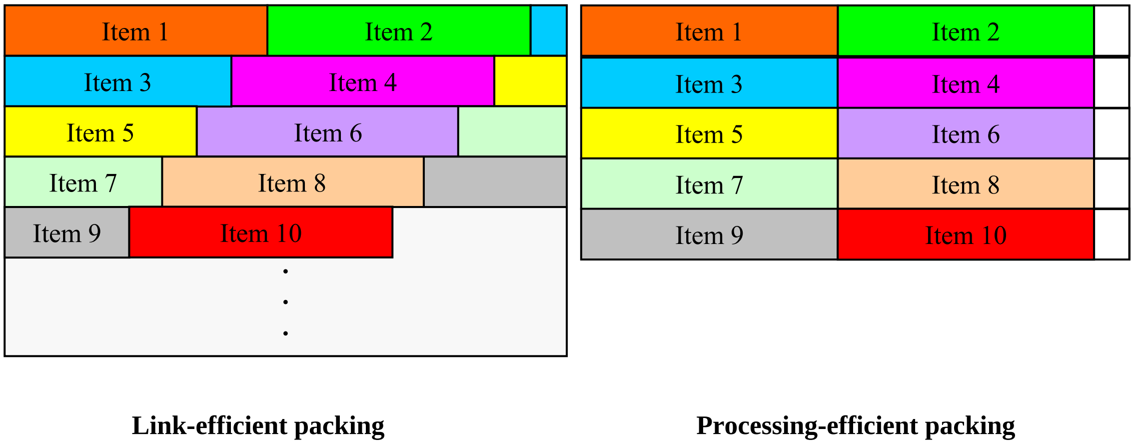


Figure 6.1.5.2.1: Fifteen-bit Item Packing Fields packed into 32-bit words.
For link-efficient packing the leftover bits at the right of each word hold the most-significant bits of an Item Packing Field, and the rest of the field is packed into the left-most bits of the next 32-bit word. For processing-efficient packing the rightmost bits are not used.

Rule 6.1.5.2-4: The originating Packet Class **shall** specify whether Processing-efficient or Link-efficient packing is used.

6.1.5.3 Ordering of Data Item Packing Fields in the Data Payload

Every Data Item in the payload must be the same size as every other data item in that payload. The meaning of the Data Items can vary however, e.g. “I” and “Q” Data Items that make up a sample. As another example, a sample vector contains multiple identical Data Items, with each relating to a different coordinate. This section presents rules for ordering these Data Items as they are packed into sequential Item Packing Fields, and thus into the Data Payload section of the IF Data Packet. Throughout this discussion, Item Packing Fields are numbered according to the following conventions:

1. The First Item Packing Field in the payload occupies the left-most bits in the first 32-bit word of the payload.
2. Within a 32-bit word, a lower numbered Item Packing Field is always to the left of a higher numbered one.
3. An Item Packing Field in an earlier 32-bit word is always numbered lower than one in a later 32-bit word.
4. In the numbering of Item Packing Fields, no number is skipped.

Based on these conventions, Item Packing Fields are understood to be arranged from left to right and from earlier 32-bit words to later ones as the Item Packing Field number increases.

Rule 6.1.5.3-1: This section gives rules for ordering Data Items within sequential Item Packing Fields. These rules **shall** be complied with independent of whether processing-efficient or link-efficient packing is used.

Definition 6.1.5.3-1: A Data Sample is either a single Data Item, in the case of real-valued samples, or a pair of Data Items, in the case of complex samples.

Definition 6.1.5.3-2: A Channel is an independent sample stream within an IF Data Packet Stream.

Definition 6.1.5.3-3: A Sample Vector is a collection of synchronous data samples. The intended use of sample vectors is for the transport of multidimensional data streams, or for the multiplexing of multiple synchronous channels into a single packet. In the case of multidimensional data streams, each sample in a sample vector represents a component of a vector in an N-dimensional signal space (given that N is the vector size). In the case of multiplexed channels, each sample in the vector represents a different signal. The sample vector “components” might be from a common process that generates a multidimensional output, or from separate but synchronous processes, each of which generates a 1-dimensional output. Whatever the source(s), the components are grouped together into a sample vector for processing, or for other application-dependent purposes.

Definition 6.1.5.3-4: Sample Component Repeating is a method of arranging the “I” and “Q” components, or amplitude and phase components, of a set of complex samples, such that all of the “I” components (or possibly amplitude components) are packed first, followed by the associated “Q” components (or possibly phase components). The number of sequential identical components is referred to as the “Repeat Count.”

Definition 6.1.5.3-5: Channel Repeating is a method of arranging Data Items such that some number of samples for each channel is packed sequentially prior to packing samples from the next channel. The number of consecutive Data Samples to be packed is referred to as the “Repeat Count.”

Rule 6.1.5.3-2: The originating Packet Class **shall** specify the following parameters affecting the ordering of Data Items in sequential Item Packing Fields:
1. Data sample type, real or complex type.
2. Whether Sample-Component repeating is in use, and the repeat count value.
3. Whether Channel repeating is in use, and what the repeat count is.
4. The width of the sample vectors (a value of one **shall** indicate vectors are not in use).

Rule 6.1.5.3-3: Earlier (i.e. corresponding to lower time values) Data Items **shall** always precede later Data Items in sequential Item Packing Fields.

Rule 6.1.5.3-4: The In-phase (I) component of a Cartesian complex sample **shall** always be packed into a lower numbered Item packing Field than its corresponding Quadrature (Q) component. The "Q" component **shall** immediately follow the corresponding "I" component when Sample-Component repeating is not in use.

Rule 6.1.5.3-5: The amplitude component of a Polar complex sample **shall** always be packet into a lower numbered Item packing Field than its corresponding phase component. The phase component **shall** immediately follow the amplitude component when Sample-Component repeating is not in use.

Rule 6.1.5.3-6: Sample Component Repeating **shall** only be used with complex samples.

Rule 6.1.5.3-7: When Sample Component Repeating is used with Cartesian samples, the number of "I" Data Items specified by the repeat count parameter **shall** be packed into sequential Item Packing Fields, with earlier Data Items going into lower numbered Item Packing Fields. This **shall** be followed by the corresponding "Q" Data Items, similarly arranged. The Repeat Count **shall** range from one to 65,536, with a value of one indicating that Sample Component Repeating is not in use.

Rule 6.1.5.3-8: When Sample Component Repeating is used with polar complex samples, the number of amplitude Data Items specified by the repeat count parameter **shall** be packed into sequential Item Packing Fields, with earlier Data Items going into lower numbered Item Packing Fields. This **shall** be followed by the corresponding phase Data Items, similarly arranged. The Repeat Count **shall** range from one to 65,536, with a value of one indicating that Sample Component Repeating is not in use.

Rule 6.1.5.3-9: When Channel Repeating is used, the number of samples specified for each channel by the repeat count **shall** be packed into sequential Item Packing Fields, with earlier Data samples going into lower numbered Item Packing Fields. The Repeat Count **shall** be in the range of one to 65,536, with a value of one indicating that Channel Repeating is not in use.

Rule 6.1.5.3-10: Sample Component Repeating and Channel Repeating are mutually exclusive features. They **shall not** be used together in the same IF Data Packet stream.

Rule 6.1.5.3-11: Sample vector width **may** range from one to 256. A value of one **shall** indicate that sample vectors are not in use.

6.1.5.4 The Interpretation of Data Items in the Data Payload

In addition to parameters affecting the packing and ordering of data in the payload, additional details must be specified to convey a basic representation of the IF Data from the source to the destination. The remaining details are:

1. For complex samples, which of three complex formats is used
2. Which of the supported Data Item formats is used
3. ~~Whether the sampled data is spectrally inverted~~

Rule 6.1.4.4-1: The originating Packet Class **shall** specify which of the following Data Item formats is used:

1. Real Data Items
2. Complex Cartesian (i.e. in-phase and quadrature components)
3. Complex Polar, signed phase, i.e. unsigned amplitude and signed phase (from –π to π) components
4. Complex Polar, unsigned phase, i.e. unsigned amplitude and signed phase (from 0 to 2π) components

~~**Rule 6.1.5.4-2:** The originating Packet Class **shall** specify whether data samples in the Data Stream Packet payload are of normal or spectrally inverted data.~~

Rule 6.1.5.4-3: The originating Packet Class **shall** specify which one of the compliant data formats is used for the non-phase Data Items in the Data Payload. The compliant formats are:

1. Fixed-point unsigned
2. Fixed-point signed
3. VITA 49 unsigned floating point
4. VITA 49 signed floating point
5. IEEE 32-bit floating-point
6. IEEE 64-bit floating-point

Rule 6.1.5.4-4: All the non-phase Data Items in all the packets in an IF Data Packet Stream **shall** use the same format.

Rule 6.1.5.4-5: All the phase Data Items in all the packets in an IF Data Packet Stream **shall** use the same format.

Rule 6.1.5.4-6: The fixed-point unsigned Data Item size **may** be any size, N, from 1 bit to 64 bits. This Data Item **shall** represent a fractional number in range of 0 to $1-2^{-N}$ inclusive. I.e. the radix point **shall** be assumed to be just left of the msb. The Originating Information Class **shall** specify the physical meaning of this range. (I think this part of the rule needs to move to the Information Class section. –Dick)

Rule 6.1.5.4-7: The fixed-point signed Data Item size **may** be any size, N, from 1 bit to 64 bits. The format **shall** be 2's compliment. The Data Item **shall** represent a fractional number in range of –½ to $½-2^{-N}$ inclusive. I.e. the radix point **shall** be assumed to be just left of the msb. The Originating Information Class **shall** specify the physical meaning of this range.

Rule 6.1.5.4-8: The VRT unsigned floating point Data Item size **may** be any size, N, from 2 bits to 64 bits. The Data Item **shall** contain two fields, a fixed-point unsigned mantissa and a fixed-point unsigned exponent, as shown in Figure 5.1.4.4-1. The mantissa **shall** be an unsigned number of any size, M, from 1 to 63 bits. The exponent **shall** be an unsigned number of any size, E, from 1 to 6 bits such that $M+E = N$. The mantissa **shall** precede the exponent, i.e. the mantissa **shall** occupy the most significant bits of the Data Item. The exponent **shall** indicate the number of bits to shift the mantissa left, with zero fill, in order to arrive at an equivalent unsigned fixed-point number. The range of the equivalent unsigned fixed-point number **shall** be 0 to $1-2^{-M}$ inclusive. I.e. the radix point **shall** be assumed to be just left of the msb when the exponent is set to its maximum value. The Originating Information Stream **shall** specify the physical meaning of this range.

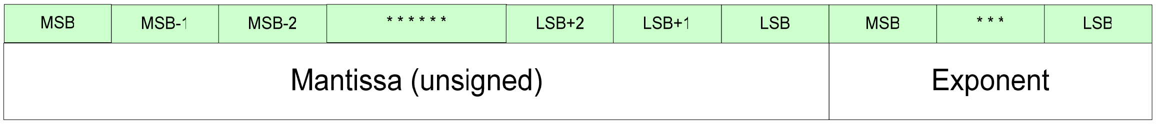


Figure 6.1.5.4-1: VRT Unsigned Floating Point Number

Rule 6.1.5.4-9: The VRT signed floating point Data Item size **may** be any size, N, from 2 bits to 64 bits. The Data Item **shall** contain two fields, a fixed-point signed mantissa and a fixed-point unsigned exponent, as shown in Figure 5.1.4.4-2. The mantissa **shall** be a signed 2's compliment number of any size, M, from 1 to 63 bits. The exponent **shall** be an unsigned number of **any** size, E, from 1 to 6 bits such that $M+E = N$. The mantissa **shall** precede the exponent, i.e. the mantissa **shall** occupy the most significant bits of the Data

Item. The exponent **shall** indicate the number of bits to shift the mantissa left, with zero fill, in order to arrive at an equivalent signed fixed-point number. The range of the equivalent signed fixed-point number **shall** be -½ to $½\text{-}2^{-M}$ inclusive. I.e. the radix point **shall** be assumed to be just left of the msb when the exponent is set to its maximum value. The originating Information Class **shall** specify the physical meaning of this range.

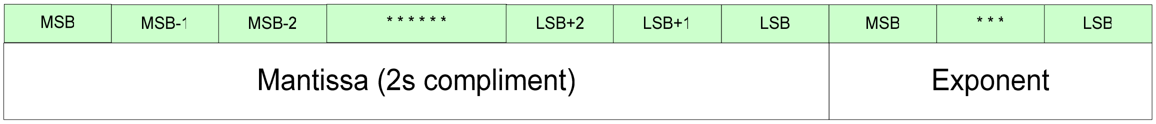


Figure 6.1.5.4-2: VRT Signed Floating Point Number

Observation 6.1.5.4-2: There is no requirement for the mantissa of VRT floating point Data Items to be normalized. This format typically results from a gain control with 6-dB step increments preceding an A/D converter. Several existing DDC ASICs accept this format as their input.

Observation 6.1.5.4-3: VITA 49 floating point numbers are appropriate for reducing link layer bandwidth when a reduction in instantaneous resolution can be tolerated.

Observation 6.1.5.4-4: The exponent is placed below the mantissa so that if the entire Data Item, including the exponent, is shifted left, the resulting error is less than one lsb of the mantissa. This is an acceptable way to reduce processing in some existing applications.

Rule 6.1.5.4-10: IEEE floating-point Data Items **shall** be either 32 bits or 64 bits in size, depending on which is used. Their format **shall** be as specified in IEEE 754. The numeric range of the IEEE floating point Data Items **shall** be as specified in IEEE 754. The originating Information Class shall specify the physical meaning of this range.

Rule 6.1.5.4-12: Signed N-bit phase components **shall** be 2's complement numbers with a range of –π to $[(1\text{-}2^{-N-}) \cdot \pi]$ radians. The N-bit value "1,0,…,0" shall correspond to a phase of –π radians, and the N-bit value "0,1,…,1" shall correspond to a phase of $[(1\text{-}2^{-N}) \cdot \pi]$ radians.

Rule 6.1.5.4-13: Unsigned N-bit phase components **shall** be 2's complement numbers with a range of 0 to $[(1\text{-}2^{-N}) \cdot 2\pi]$ radians. The N-bit value "0,…,0" shall correspond to a phase of 0 radians, and the N-bit value "1,…,1" shall correspond to a phase of $[(1\text{-}2^{-N}) \cdot 2\pi]$ radians.

6.1.5.5 Data Structure and Payload Size

Definition 6.1.5.5-1: A Basic Data Structure is a collection of contiguous Item Packing Fields in which the meaning of each Data Item is predetermined based on its offset from the start of the structure, and which is complete in the sense that the entire data payload can be described as a sequence of identical Basic Data Structures.

Observation 6.1.5.5-1: The combination of the various parameters affecting Data Item arrangement in the payload provides for a variety of different Basic Data Structures. In the simplest case, the payload contains a sequence of real data samples, and the resulting structure is a single Item Packing Field containing a single Data Item. When samples are complex, or when sample vectors are used, or when sample component repeating or channel repeating is used, the structures are larger. The largest structures are generated by the simultaneous use of all the compatible features in this set.

Rule 6.1.5.5-1: The number of 32-bit words in the Data Payload field **shall** be the smallest number needed to contain the desired (integer) number of Basic Data Structures. The size of the Basic Data Structure, i.e.

the number of Item Packing Fields required to contain it, **shall** be calculated as the number of Data Items in a sample multiplied by the vector size, multiplied by the repeat count.

6.1.6 Trailer

Rule 6.1.6-1: Every VRT IF Data Packet **shall** contain a Trailer.

The IF Data Packet Trailer contains four fields, as shown in Figure 6.1.5-1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Enables												State and Event Indicators												E	Associated Context Packet Count						

Figure 6.1.6-1: Data Stream Packet Trailer Word

Together, the Enables field and the State and Event Indicators field provide the capability to mark an IF Data Packet with one or more data events or state updates to be communicated from a Source to a Destination. An Event Indicator might indicate a system synchronization signal or other event that affects some portion, or all, of the IF Data Packet payload. A State update might be an Indication of tuner phase-lock. When these fields are used, no provision is made for indicating the precise time of the events or state changes. They are only understood to be associated with the data in that packet. The "Enables" field contains an enable bit for each Indicator bit in the State and Event Indicators field. Some of the Indicators (and their enable bits) are predefined, and some are user-defined. Table 6.1.5-1 shows the bit positions and the functions of each bit in these fields.

Rule 6.1.6-2: State and Event Indicators and Enable bits **shall** be positioned as indicated in Table 6.1.5-1.

Rule 6.1.6-3: For each Indicator bit, there is a corresponding enable bit. The associated enable bit for each Indicator bit **shall** be the one that is in the same position in the Enables field as the associated Indicator bit is in the State and Event Indicators field.

Rule 6.1.6-4: For the predefined Indicators, when an Enable bit is set to one, the corresponding Indicator **shall** function as indicated in Table 6.1.5-1. Otherwise the meaning of the corresponding Indicator is undefined.

Rule 6.1.6-5: When an Enable bit is set to zero in an IF Data Packet, the assumed state of the corresponding Indicator **shall** be the state most recently communicated in a previous IF Data Packet with that enable bit set to one.

Permission 6.1.6-1: The user-defined Indicators **may** be used for any purpose. They **may** be used together as well as individually.

Rule 6.1.6-6: The originating Packet Class **shall** specify all "user-defined" Indicators in the Trailer, along with the meaning of the binary states.

Rule 6.1.6-7: When user-defined Indicators are used individually, the corresponding enable bit for each Indicator **shall** operate in a manner identical to the enables for the predefined Indicators. When user-defined Indicator bits are used in together, the corresponding enable bits **shall** be identical, and together shall indicate whether theIndicator group is enabled. As with the individual enables, a one **shall** indicate that the Indicator group is enabled.

Enable Bit Position	Indicator Bit Position	Indicator Name
31	18	Calibrated Time Indicator
30	17	Valid Data Indicator
29	16	Reference Lock Indicator
28	15	AGC/MGC Indicator
27	14	Detected Signal Indicator
[26..25]	[13..12]	Reserved
[24..20]	[11..7]	User-Defined Indicator

Table 6.1.6-1: Indicator bits and Qualifier Enable bits
Each Indicator functions as indicated, but only when the corresponding
Enable bit is set otherwise, the Qualifier bit is undefined.

Rule 6.1.6-8: The Calibrated Time Indicator, when set to one, **shall** indicate that the timestamp in the IF Data Packet is calibrated to some external reference. When set to zero, **shall** indicate that the timestamp is free-running, and may be inaccurate.

Rule 6.1.6-9: The Valid Data Indicator, when set to one, **shall** indicate that the data in the packet is valid. When set to zero, **shall** indicate that some condition exists that may invalidate the data.

Observation 6.1.6-1: The use of Valid Data Indicator is application dependent. When used in the output of an RF Tuner, for example, it may indicate that the data in the packet is valid or invalid during frequency changes. The Valid Data Indicator could be set to zero when a frequency change is initiated and a phase-locked loop is not locked, and later set to one when frequency change is complete and the phase-locked loop is locked.

Rule 6.1.6-10: The Reference Lock Indicator, when set to one, **shall** indicate that any phase-locked loops (PLL) affecting the data are locked and stable. When set to zero, **shall** indicate that at least one PLL is not locked and stable.

Rule 6.1.6-11: The AGC/MGC Indicator, when set to one, **shall** indicate that Automatic Gain Control is active. When set to zero, this tag **shall** indicate manual gain control.

Rule 6.1.6-12: The Detected Signal Indicator, when set to one, **shall** indicate that the data contained in the packet contains some detected signal. The originating Packet Class **shall** specify what signal is being detected. (Information Class?)

The remaining two fields in the Trailer optionally provide a count of the number of Context Packets relating to each IF Data Packet. This provides information that events or other context changes associated with the data in the packet either have been or are yet to be communicated via Context Packets. These events or context changes may be of the same type as are communicable via the Event Tags in the Item Packing Fields, or as the State and Event Indicators in the Trailer, or may be of other types. *Note: Unlike the use of State and Event Indicators in the Trailer, the use of Context Packets to convey this information allows for more precise communication regarding the timing of events, and specific values of changing parameters. The capabilities of Context Packets are discussed in Section 7.*

The first, one-bit, field is the “Associated Context Packet Count Enable” bit. This bit is labeled “E” in Figure 6.1.5-1. The other field is the “Associated Context Packet Count” field.

Rule 6.1.6-17: When the “E” bit is set to one, the “Associated Context Packet Count” **shall** provide a count of all of transmitted Context Packets that are directly or indirectly associated with the IF Data Packet, OR a count of some special subset of these. If only a subset is counted, the originating Information Class **shall**

specify what that subset is. When the "E" bit is cleared, the "Associated Context Packet Count" is undefined.

Observation 6.1.6-2: In some applications only the subset of the Context Packets that affect data processing immediately downstream might be counted. This would assure that the downstream process knows to wait for that context prior to processing the data.

Observation 6.1.6-3: In some applications only those associated Context Packets generated by the same process that generated the data might be counted. Other Context Packets from other sources might be later associated somewhere downstream, where the count can then be modified to reflect the full number.

Rule 6.1.6-18: When used, the 7-bit "Associated Context Packet Count" field **shall** contain an unsigned number in the range of zero to 127 inclusive. The lsb of the number **shall** be the right-most bit in the field.

Permission 6.1.6-2: The "Associated Context Packet Count" **may** include associated Context Packets transmitted by a process different from the one that transmits the IF Data Packet containing the count.

Recommendation 6.1.6-1: All Trailer bits **should** be set to zero when the Trailer is unused.

Recommendation 6.1.6-2: In a VRT Information Stream, all Context Packets affecting a particular IF Data Packet, should be transmitted before transmitting the following IF Data Packet for that Packet Stream.

6.2 *Extension Data Packet Classes and Streams*

An Extension Data Packet Stream conveys a data payload that is unique to a particular application. Examples of data that could be placed in an Extension Data Packet include pulse descriptor words, or FFT data.

The originating Packet Class provides the specifications for the custom data payload format used.

The packet format for all Extension Data Packets is shown in Figure 6.2-1. It has a structure that is less constrained than that of IF Data Packets. It contains a mandatory Header, an optional Stream ID, and a mandatory Data Payload.

Rule 6.2-1: Every Extension Data Packet **shall** be organized as shown in Figure 6.2-1.

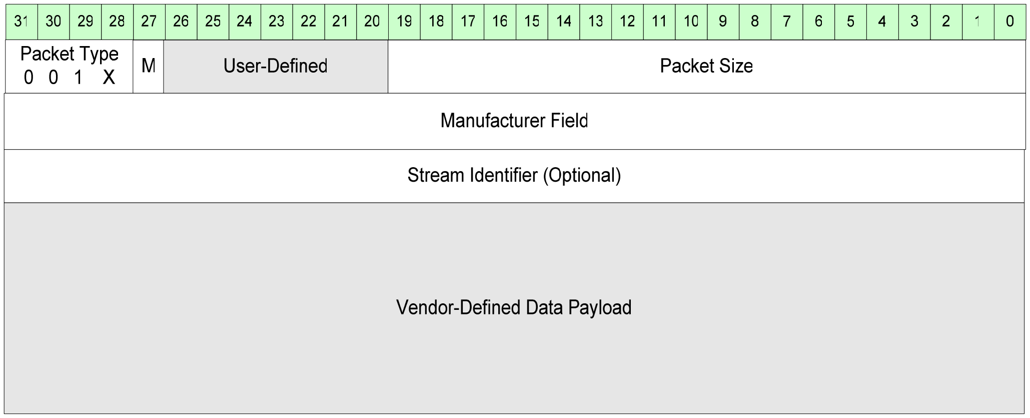


Figure 6.2-1: The Format of the Extension Data Packet
(The first 32-bit word in the packet is the one at the top.)

Rule 6.2-2: An Extension Data Packet **shall** be an integer number of 32-bit words in size.

Rule 6.2-3: The Extension Data Packet **shall** include the Packet Type, M bit, and Packet Size fields with the same functionality and in the same locations as in IF Data Packets.

Rule 6.2-4: In an Extension Data Packet Stream, the "Packet Type" field **shall** contain the binary value "0010," when a Stream ID is included, or "0011" when it is not, as shown in Table 6.1.1-2.

Observation 6.2-1: An Extension Data Packet Class may or may not specify the use of the Packet Count and TSI fields of the IF Data Stream Packet Class. In an Extension Data Packet Stream, these bits may be allocated to any function desired.

Rule 6.2-5: The Manufacturer field in Extension Data Packets **shall** function in the same way as in the IF Data Packets.

Rule 6.2-6: The Stream ID, when present in an Extension Data Packet Stream, **shall** function in the same way as in an IF Data Stream.

Rule 6.2-7: Extension Data Packets may be structured in any way that does not exceed the maximum packet size of 1048575 32-bit words. The originating Packet Class **shall** specify how the packet is structured.

Recommendation 6.2-1: Extension Data Packets **should** use the same Header, Timestamp, and Trailer definitions as required for IF Data Packets whenever possible, in order to maximize interoperability and minimize additional development time.

Rule 6.2-8: The originating Packet Class **shall** describe the function of the custom bits in the header of Extension Data Packets. These bits **may** be used for any purpose, or **may** be unused.

Rule 6.2-9: The originating Packet Class shall describe the format of the payload section of the Extension Data Packet.

7 Context Packet Classes and Streams

This section sets forth the rules controlling the structure and function of Context Packet Classes and Context Packet Streams. A Context Packet Class is the specification of a structure and function for a packet type that conveys metadata related to a signal. This metadata includes information required to give full meaning to the signal, as well as certain information about the equipment or processes involved in creating the signal. As explained in Section 4, VRT defines two templates for Context Packet Classes: the IF Context and Extension Context Classes. With each class template, any number of Context Packet classes may be created, each with its own unique choice of parameters. The following sections specify requirements for and restrictions upon these packet classes, and for their documentation.

IF Data Packet Streams are the backbone of the VRT specification and IF Context Packet Streams are the primary means of communicating metadata concerning these IF Data Packet Streams. However, IF Context Packet Streams can also communicate metadata about analog signals. For example, in the simple VRT system shown in Figure 1.3-1, assume that both the Downconverter and the ADC processes generate IF Context Packets. The IF Context packets for the Downconverter and ADC refer to the analog IF signal and IF Data Packet Streams, respectively. In both cases, the signal to which the Context Packet refers to is called the referred signal.

The IF Context Packet Class contains the most common RF-related metadata, including frequency, level, and timing information related to the referred signal. They also contain information concerning the location and orientation of the RF equipment necessary for geolocation applications. The IF Context Packet Class contains a timestamp field that indicates when the information in the contained packet takes effect because equipment settings and location can change at any time, and knowledge of the timing of these changes is often useful.

In many cases it will be necessary to communicate metadata that can’t be communicated in the IF Context Packet Class. For these cases, an Extension Context Packet Class (See Section 7.2) is provided. This type of packet class can be custom tailored to the information that is required for the particular application. It is not intended to be used to convey the “standard” information the IF Context Packet Class conveys. It is intended to augment the capabilities of the IF Context Packet Class for applications requiring additional metadata.

The use of Context Packets in a VRT system is optional. When Context Packets are used, they will typically be transmitted as a result of a change in one of the states or parameters communicable by that Context Packet Class. Context Packets will also typically be transmitted at some periodic interval in order to ensure that a VRT receiver can recover from any miscommunication or loss of communication. Such periodic updates would typically involve retransmitting all fields required by the application.

Permission 7-1: There are no restrictions on the rate or time at which Context Packets are emitted. When a Context Packet is emitted, it **may** contain as many or as few of the optional fields as required.

Observation 7-1: The state of the sensor might be conveyed entirely by a single Context Packet or by the combination of multiple Context Packets.

Observation 7-2: To minimize link utilization, each Context Packet might contain only fields conveying information that has changed. At the other extreme, each packet might contain a full update, including fields that never change. In between the extremes there is a continuum of possibilities. System requirements will dictate when Context Packets are emitted, and what they will contain.

7.1 *IF Context Packet Classes and Streams*

This section sets forth rules controlling the structure and function of the IF Context Packet Class template, which is the superset of all possible IF Context Packet Classes. Individual IF Context Packet classes are formed from this template by selecting a subset of supported fields and possibly restricting the range of valid contents for those fields. Individual IF Context Packet Streams are created as a sequence of packets from a class. Multiple streams can be formed from the same class, each with its own unique value in the Stream Identifier field.

Rule 7.1-1: The order of the fields in an IF Context Packet **shall** be as shown in Figure 6.1.5.5-1.

Rule 7.1-2: If an optional field is not present in an IF Context Packet Class, the remaining words in the packet **shall** "move up" toward the header, with no padding.

Rule 7.1-3: If an optional field is present in an IF Context Packet Class, but is omitted in a transmitted packet, the remaining words in that packet **shall** "move up" toward the header, with no padding.

Observation 7.1-1: Rule 7.1-3 pertains to fields specified by the packet class as optional in the transmitted packets. For example, the Reference Level field, though included in the packet class, may not be present in every packet.

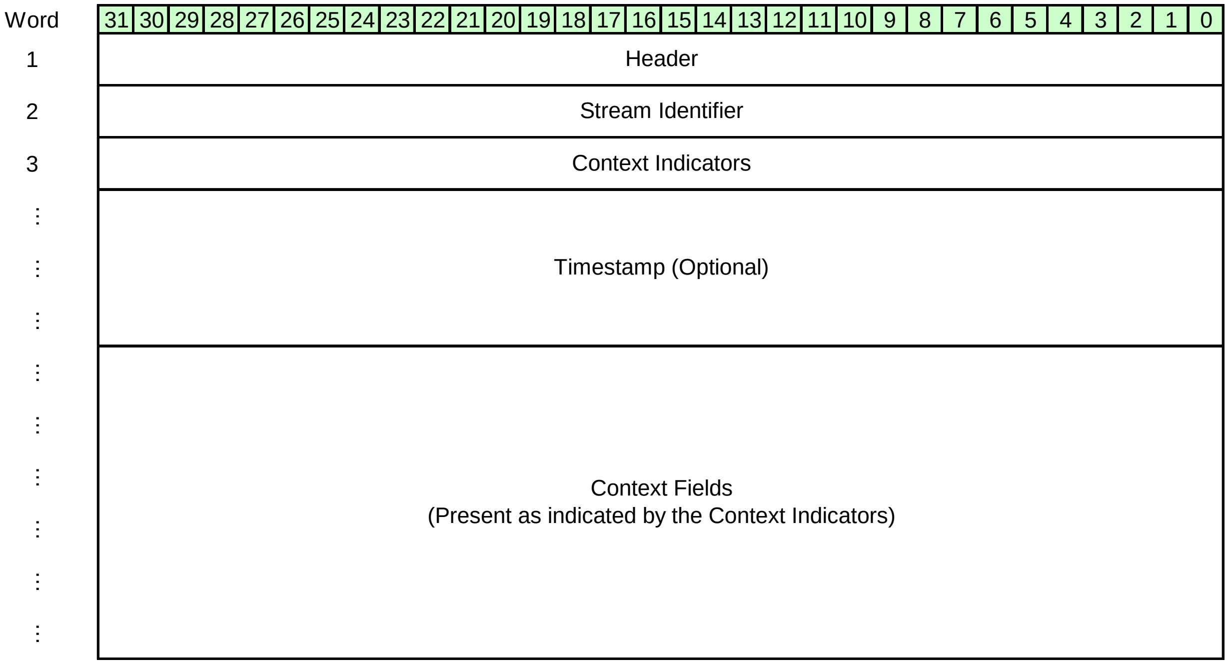


Figure 6.1.5.5-1: The template for the IF Context Packet Class

The IF Context Packet is divided into five sections, described below.

The Header is the first 32-bit word in the IF Context Packet. The Header is mandatory. The Header contains the Packet Type field that identifies the VRT packet as being an IF Context Packet and a Packet Count field facilitating lost packet detection. It also includes a Timestamp Indicator field, that specifies the format of the optional Timestamp field and the Timestamp Resolution field that indicates whether the Timestamp indicates the context change time with sample/picosecond resolution, or with only packet resolution. Lastly, the Packet Size field specifies the size of the IF Context Packet in 32-bit words.

The Stream Identifier, or Stream ID, field is the second 32-bit word in the IF Context Packet. It is also mandatory. It is used to indicate to which packet stream this packet belongs. Stream IDs are used to associate different packet streams with an Information Stream, as described in Section 8.

The Context Indicator field is the third 32-bit word in the Context Packet. It is the last mandatory field in the IF Context Packet. The bits in the Context Indicator field indicate which optional Context Fields are present in this Context Packet.

The Timestamp field indicates the time of the context update. It typically serves to correlate context changes conveyed in the IF Context Packet with specific samples in the associated Data Packet Stream. The Timestamp field is optional, and its inclusion in the IF Context Packet is indicated by the value in the “TSI” field in the first word of the IF Context Packet Header.

The Context Fields contain the metadata, i.e. context, updates. Each field is optional. The inclusion of each field is determined by the value of the Context Indicator Field.

7.1.1 Header

The Header of the IF Context Packet matches the Header of the IF Data Packet with the exception of the omission of the M bit and the addition of the TSM bit. The format of the IF Context Packet Header is shown in Figure 7.1.1-1.

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Packet Type				R	TSM	TSI		Packet Count				Packet Size																			
	0	1	0	0																												

Figure 7.1.1-1: Context Packet Header

Rule 7.1.1-1: Every IF Context Packet **shall** contain the Header shown in Figure 7.1.1-1.

Rule 7.1.1-2: The IF Context Packet Header **shall** contain the following information: Packet Type, Packet Count, TSM, TSI, and Packet Size.

> **Rule 7.1.1-3:** In an IF Context Packet, the “Packet Type” field **shall** contain the binary value “0110,” as shown in Table 5.1.1-2.
>
> **Rule 7.1.1-4:** The Reserved bit in the Header, bit 27, **shall** be set to zero.
>
> The Timestamp Mode (TSM) bit is used to indicate whether the Timestamp in the IF Context Packet is being used to convey the timing of events or context changes with fine or coarse resolution.
>
> **Rule 7.1.1-5:** The TSM bit **shall** be set to one when the Timestamp in the IF Context Packet is being used to convey the timing of events or context changes with fine resolution, which is the highest resolution supported by the Timestamp with its current TSI setting (either sample or picosecond resolution).
>
> **Rule 7.1.1-6:** The TSM bit **shall** be set to zero when the Timestamp in the IF Context Packet is being used to convey the timing of events or context changes with coarse resolution, which is the period of the directly associated Data Packet whose Timestamp matches that of this Context Packet. The period of the data packet is the time between the last sample of the previous Data Packet and the last sample of the current Data Packet.
>
> **Rule 7.1.1-7:** The Timestamp Indicator (TSI) field **shall** function as defined for the IF Data Packet Class.
>
> **Rule 7.1.1-8:** If an IF Context Packet Stream is directly associated to a Data Packet Stream, the Timestamp Indicator (TSI) bits for that IF Context Packet Class **shall** match the TSI bits for the directly associated Data Packet Stream (i.e. they will use the same type of Timestamp as described in Section 5.2.3).
>
> **Rule 7.1.1-9:** The Context Packet Class documentation **shall** specify the accuracy of the value indicated in the Context Packet Timestamp when sample or picosecond resolution is indicated by the TSM bit.

Rule 7.1.1-10: The Packet Count field **shall** function in the IF Context Packet Class as defined for the IF Data Packet Class.

Rule 7.1.1-11: The Packet Size field **shall** function in the IF Context Packet Class as defined in the IF Data Packet Class.

7.1.2 Stream Identifier

Stream Identifiers in the IF Context Packet Class are used in the same fashion as in the IF Data Packet Class (see Section 6.1.3) to identify particular packets as belonging to certain packet streams. In addition, the Stream Identifier in Context Packets is used to associate different packet streams into an Information Stream.

There are two types of association using Stream Identifiers. The first is Direct Association, where a Context Packet Stream is associated to a Data Packet Stream. The second is Context Association, where a Context Packet Stream is associated with another Context Packet Stream. The differences between these types of association are shown in Table 7.1.2-1 below:

Direct Association	**Context Association**
A Context Packet Stream is directly associated to a Data Packet Stream when it shares the same Stream ID.	A Context Packet Stream is associated to another Context Packet Stream by including the Stream ID of one into the other's Stream Association List (see Section 7.1.5.22).
A Context Packet Stream may be directly associated with either zero or one Data Packet Streams. Directly associated streams will typically originate from the same source.	A Context Packet Stream may be associated with any number of other Context Packet Streams. Associated Context Packet Streams can originate from many difference sources.

Table 7.1.2-1: Direct and Context Association Comparison

The rules for the Stream Identifier in the IF Context Packet Class are as specified in the IF Data Packet Class (Section 6.1.3) with the following differences.

Rule 7.1.2-1: Every IF Context Packet **shall** contain a Stream Identifier.

Observation 7.1.2-1: Stream IDs are optional for Data Packets because they are unnecessary for the simplest data-only single packet stream systems. Systems using Context Packets are inherently more complicated, and require Stream IDs to determine membership to Information Streams.

Rule 7.1.2-2: If an IF Context Packet Stream is directly associated with a Data Packet Stream, the Stream Identifier of the IF Context Packet Stream **shall** match that of the Data Packet Stream.

7.1.3 Context Indicator Field

Rule 7.1.3-1: Every IF Context Packet **shall** contain a Context Indicator field.

Definition 7.1.3-1: The Context Indicator field contains bit fields, one for each Context Field, that indicate whether the corresponding optional Context Field is present in the packet. The Context Fields are described in detail in Section 7.1.5.

Rule 7.1.3-2: The bit fields within the Context Indicator field shall be arranged as shown in Table 7.1.3-1.

Rule 7.1.3-3: If a bit is set in the Context Indicator field, the corresponding Context Field entry **shall** be included in the IF Context Packet in the sequence shown in Table 7.1.3-1.

Bit Position	Context Indicator Field	Number of Words in Context Field
31	Reference Point Identifier	1
30	Bandwidth	2
29	IF Reference Frequency	2
28	RF Reference Frequency	2
27	RF Reference Frequency Offset	2
26	Bandwidth Frequency Offset	2
25	Gain	1
24	Reference Level	1
23	Overrange Count	1
22	Sample Rate	2
21	Timestamp Adjustment	2
20	Temperature	1
19	Source Type	2
18	State and Event Indicator	1
17	IF Data Packet Payload Format	2
16	Formatted GPS (Global Positioning System) Geolocation	11
15	Formatted INS (Inertial Navigation System) Geolocation	11
14	ECEF (Earth-Centered, Earth-Fixed) Ephemeris	9
13	Relative Ephemeris	9
12	Ephemeris Reference Identifier	1
11	GPS ASCII	Variable
10	Stream Association List	Variable
9..0	Reserved	n/a

Table 7.1.3-1: Context Indicator Field Definition

7.1.4 Timestamp Field

The definition and expression of the Timestamp field in the IF Context Packet is identical to the Timestamp field in the IF Data Packet. However, the interpretation of the Timestamp field is dependent on the "TSM" bit in the Header, as described in Section 7.1.1.

7.1.5 Context Fields

The following subsections describe each of the context fields that may be present in the Context Fields section of the Context Packet. Their presence or absence is indicated by the values in the bit fields of the Context Indicator field described above in Section 7.1.3.

7.1.5.1 Reference Point Identifier

Definition 7.1.5.1-1: The Reference Point is the point in a system where the context fields listed below, which are conveyed in an IF Context Packet, are understood to properly explain the referred signal. The Reference Point is always located at the input of a process in the system. The context fields whose meaning is tied to the Reference Point are the:

- RF Reference Frequency
- RF Reference Frequency Offset
- Gain
- Reference Level
- Timestamp Adjustment

The Reference Point Identifier (or Reference Point ID) contains the Stream ID of the process at the Reference Point. See Appendix **B.1** for a simple example using the Reference Point Identifier field.

Rule 7.1.5.1-1: The Reference Point ID **shall** use the format shown in Figure 7.1.5.1-1.

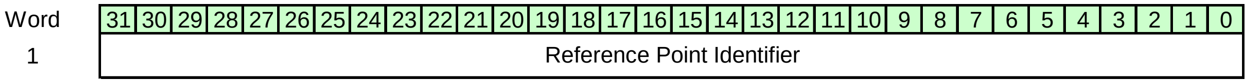


Figure 7.1.5.1-1: Reference Point ID Format

Rule 7.1.5.1-2: The Reference Point ID, when used, **shall** contain the Stream Identifier of the process whose input is the Reference Point.

Observation 7.1.5.1-1: There may be multiple reference points in a system. Different components within the same system may or may not declare the same Reference Point.

Observation 7.1.5.1-2: The **VRT equipment provider** may choose to provide the ability for the Reference Point to be modified by a system designer. This would require the ability for the frequency, level, and timestamp adjustment context fields to be re-calibrated to the new Reference Point.

7.1.5.2 Bandwidth Field

The Bandwidth field is used to describe the amount of usable spectrum at the output of a process. It is not used to describe the Nyquist bandwidth for sampled data streams, which is typically larger than the amount of usable spectrum, and it is not typically used to describe the bandwidths of signals of interest, which are usually smaller than the amount of usable spectrum. See Appendix **B.2** for a simple example showing the meaning of the Bandwidth.

Rule 7.1.5.2-1: The Bandwidth field **shall** convey the amount of usable spectrum at the output of a process.

Rule 7.1.5.2-2: The Context Packet Class documentation **shall** specify the band-edge definition(s) used in calculating the Bandwidth field.

Observation 7.1.5.2-1: The amount of usable spectrum is usually set by band-limiting filters, but may also be affected by other phenomena such as aliasing or spurious signals. Possible bandwidth measurement criteria for filters are 1dB / 3dB / 6dB bandwidth, null-to-null, etc. It is up to the **VRT equipment provider** to determine and specify the meaning of "bandwidth."

Rule 7.1.5.2-3: The value of the Bandwidth field **shall** be expressed in units of Hertz. The Bandwidth field **shall** use the 64-bit two's complement format shown in Figure 7.1.5.2-1. This field has an integer and a fractional part, with the radix point to the right of bit 20.

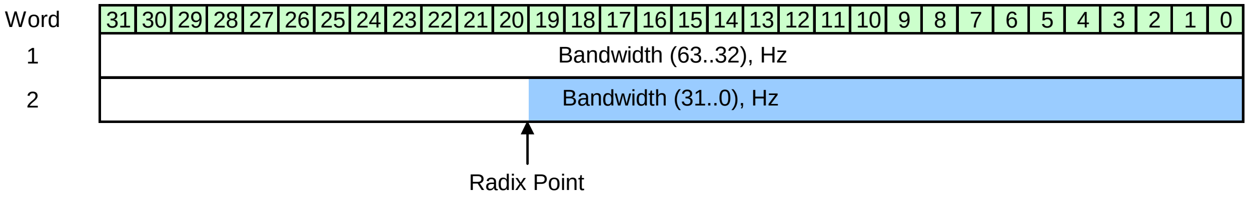


Figure 7.1.5.2-1: Bandwidth Field

Observation 7.1.5.2-2: The spectral band described by the Bandwidth field is typically oriented symmetrically about the IF Reference Frequency described in the next section. For cases where the band is not symmetric about the IF Reference Frequency it is necessary to also send the Bandwidth Frequency Offset field described in Section 7.1.5.6.

Observation 7.1.5.2-3: A Bandwidth field containing 0x0000 0000 0010 0000 indicates a bandwidth of 1 Hz.

Observation 7.1.5.2-4: The value of the Bandwidth field has a range of ±8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.2-1: The Context Packet Class documentation **should** specify the accuracy of the values reported in the Bandwidth field.

7.1.5.3 IF Reference Frequency Field

When a Data Packet Stream conveys a signal that has been translated from some higher frequency band, it is often necessary to communicate band of origin for the signal. In most cases, this is accomplished using the IF and RF Reference Frequency fields. The point in the signal path that corresponds to the original frequency is indicated with the Reference Point Identifier field described in 7.1.5.1. The original frequency is specified with the RF Reference Frequency field, and the subsequent translated frequency is specified with the IF Reference Frequency field. The IF Reference Frequency is a point within the energy band of the referred signal, typically its center frequency.

$$\text{RF Reference Frequency} \xleftrightarrow{\substack{\textit{Frequency}\\ \textit{Translation}}} \text{IF Reference Frequency}$$

The above relationship holds unless the RF Reference Frequency Offset field is also utilized by the IF Context Packet Class, in which case the relationship is described in Section 7.1.5.5.

The IF Reference Frequency also serves at the point of reference for the center of the band described by the Bandwidth field.

$$\textit{Band Center} = \text{IF Reference Frequency}$$

This relationship holds unless the Bandwidth Frequency Offset field is also utilized by the IF Context Packet Class, in which case the relationship is described in Section 7.1.5.6.

See Appendix **B.2** for an example using the IF Reference Frequency.

Rule 7.1.5.3-1: The IF Reference Frequency field, when used, **shall** indicate a frequency within the range of usable spectrum described by the Bandwidth and Bandwidth Frequency Offset fields.

Rule 7.1.5.3-2: When the associated data stream contains real samples, the IF Reference Frequency **shall** be within the range of 0 and $+f_S/2$. When the associated data stream contains complex samples, the IF Reference Frequency **shall** be within the range of $- f_S/2$ and $+ f_S/2$. If the referred signal to is an analog signal, there are no restrictions on the IF Reference Frequency.

Recommendation 7.1.5.3-1: When the signal referred to is an analog signal, the IF Reference Frequency should be the IF center frequency.

Rule 7.1.5.3-3: The value of the IF Reference Frequency **shall** be expressed in units of Hertz. The IF Reference Frequency subfield **shall** use the 64-bit two's complement format as shown in Figure 7.1.5.3-1.

This field has an integer and a fractional part, with the radix point to the right of bit 20 in the second 32-bit word.

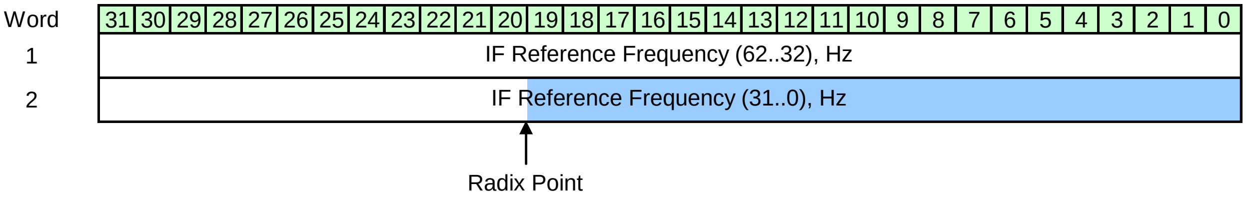


Figure 7.1.5.3-1: IF Reference Frequency Field

Observation 7.1.5.3-1: The spectrum of the referred signal may be inverted. This is indicated by the Spectral Inversion Indicator bit in the State and Event Indicators field, described in Section 7.1.5.14.

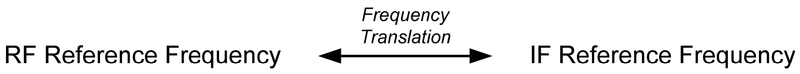
Observation 7.1.5.3-2: An IF Reference Frequency field value of 0x0000 0000 0010 0000 represents a frequency of 1 Hz. An IF Reference Frequency field containing 0x7FFF FFFF FFF0 0000 indicates a frequency of –1 Hz.

Observation 7.1.5.3-3: The value of the IF Reference Frequency field has a range of ±8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.3-2: The Context Packet Class documentation **should** specify the accuracy of the value reported in the IF Reference Frequency field.

7.1.5.4 RF Reference Frequency Field

When a Data Packet Stream conveys a signal that has been translated from some higher frequency band, it is often necessary to communicate band of origin for the signal. In most cases, this is accomplished using the IF and RF Reference Frequency fields. The point in the signal path that corresponds to the original frequency is indicated with the Reference Point Identifier field described in 7.1.5.1. The original frequency is specified with the RF Reference Frequency field, and the subsequent translated frequency is specified with the IF Reference Frequency field. The IF Reference Frequency is a point within the energy band of the referred signal, typically its center frequency.

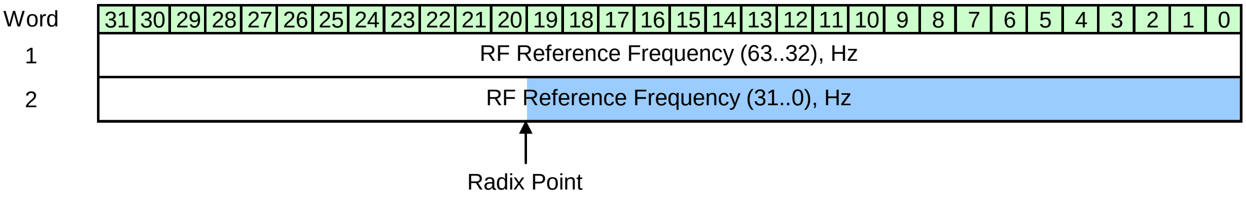


The above relationship holds unless the RF Reference Frequency Offset field is also utilized by the IF Context Packet Class, in which case the relationship is described in Section 7.1.5.5.

See Appendix B.2 for an example using the RF Reference Frequency.

Rule 7.1.5.4-1: When the RF Reference Frequency Offset field is not used, the value of the RF Reference Frequency **shall** be the frequency at the reference point that translates to the frequency specified in the IF Reference Frequency field.

Rule 7.1.5.4-4: The value of the RF Reference Frequency **shall** be expressed in units of Hertz. The RF Reference Frequency value field **shall** use the 64-bit two's complement format as shown in Figure 7.1.5.4-1. This subfield has an integer and a fractional part, with the radix point to the right of bit 20 in the second 32-bit word.



Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	RF Reference Frequency (63..32), Hz																															
2	RF Reference Frequency (31..0), Hz																															

Figure 7.1.5.4-1: RF Reference Frequency Field

Observation 7.1.5.4-6: A RF Reference Frequency field value of 0x0000 0000 0010 0000 represents a frequency of 1 Hz. An RF Reference Frequency field containing 0x7FFF FFFF FFF0 0000 indicates a frequency of –1 Hz.

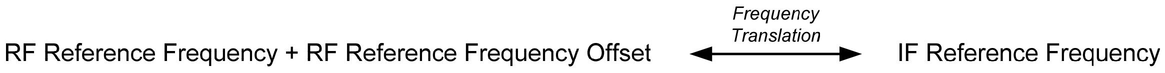
Observation 7.1.5.4-7: The value of the RF Reference Frequency field has a range of ±8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.4-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the RF Reference Frequency field.

7.1.5.5 RF Reference Frequency Offset Field

Some processes, such as channelizers, create a large number of narrowband signals from a wideband input signal. Each of the output signals originated from a frequency band at a constant offset from the band center of the channelizer input signal. In a system where a channelizer follows a tuner, the RF center frequencies corresponding to each channelizer output change whenever the tuner frequency changes. This could lead to the transmission of a large number of Context Packets, one for each channelizer output, conveying their new RF Reference Frequency fields. The transmission of such a large number of new packets could cause link congestion. The RF Reference Frequency Offset field provides a method to send only a single context packet in such instances, avoiding potential link congestion.

The presence of the RF Reference Frequency Offset in an IF Context Packet Class changes the relationship between the IF and RF Reference Frequencies described in Sections 7.1.5.3 and 7.1.5.4. When the RF Reference Frequency Offset field is present, the original frequency is the sum of the RF Reference Frequency and RF Reference Frequency Offset.



See Appendix B.2 for an example using the RF Reference Frequency Offset.

Rule 7.1.5.5-1: When the RF Reference Frequency Offset field is used, the sum of the RF Reference Frequency Offset and RF Reference Frequency conveyed in an associated Context Packet **shall** be the frequency at the Reference Point that is translated to the frequency specified in the IF Reference Frequency field. The Reference Point where this sum applies **shall** be the one specified by Reference Point Identifier in the Context Packet containing the RF Reference Frequency field.

Rule 7.1.5.5-2: The value of the RF Reference Frequency Offset field **shall** only be combined with the value of the RF Reference Frequency fields for associated upstream processes.

Rule 7.1.5.5-3: The value of the RF Reference Frequency Offset **shall** be expressed in units of Hertz. The RF Reference Frequency Offset field **shall** use the 64-bit two's complement format shown in Figure 7.1.5.5-1. This field has an integer and a fractional part, with the radix point to the right of bit 20.

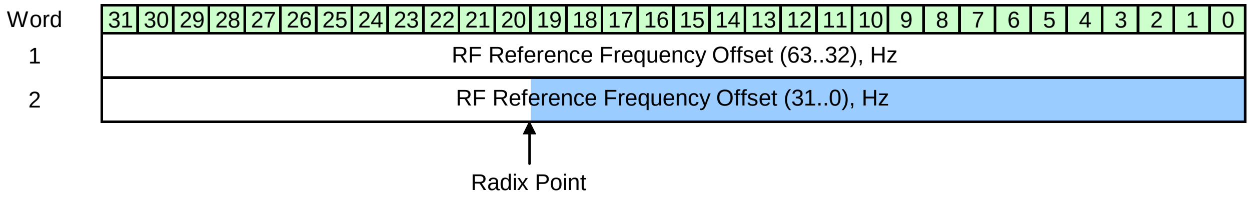


Figure 7.1.5.5-1: RF Reference Frequency Offset Field

Observation 7.1.5.5-1: The RF Reference Frequency Offset field is useful for describing the frequencies of a bank of filters, FFT channelizers, DDCs, etc.

Observation 7.1.5.5-2: A RF Reference Frequency Offset field containing 0x0000 0000 0010 0000 indicates a RF Reference Frequency Offset of +1 Hz.

Observation 7.1.5.5-3: The value of the RF Reference Frequency Offset field has a range of ±8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.5-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the RF Reference Frequency Offset field.

7.1.5.6 Bandwidth Frequency Offset Field

Typically the spectral band whose width is described by the Bandwidth field of Section 7.1.5.2 is symmetric about the IF Reference Frequency described in Section 7.1.5.3. For those cases where it is not symmetric about this frequency, the Bandwidth Frequency Offset field is used to specify the frequency offset from the IF Reference Frequency and the center of the band. When the Bandwidth Frequency Offset field is present in the IF Context Packet Class, the center of the band is located at the sum of the IF Reference Frequency and the Bandwidth Frequency Offset.

$$\textit{Band Center} = \text{IF Reference Frequency} + \text{Bandwidth Frequency Offset}$$

Rule 7.1.5.6-1: The value of the Bandwidth Frequency Offset field **shall** be the band center frequency minus IF Reference Frequency, where the band center frequency is the exact center of the band whose width is described by the Bandwidth field.

Observation 7.1.5.6-1: The Bandwidth Frequency Offset field will contain a positive value when the center of the band is higher than the IF Reference Frequency, and a negative value when the center of the band is lower than the IF Reference Frequency.

Rule 7.1.5.6-2: The value of the Bandwidth Frequency Offset **shall** be expressed in units of Hertz. The Bandwidth Frequency Offset field **shall** use the 64-bit two's complement format shown in Figure 7.1.5.6-1. This field has an integer and a fractional part, with the radix point to the right of bit 20.

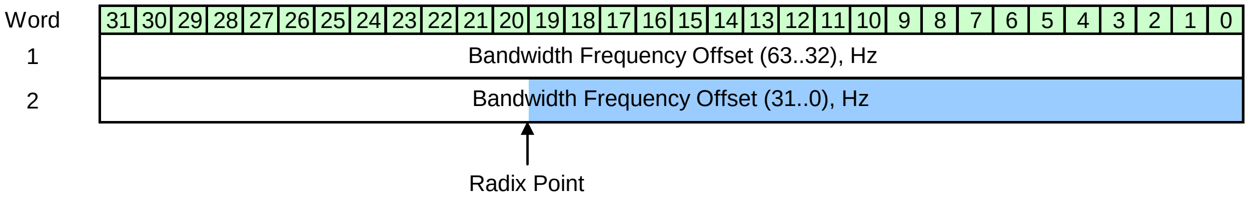


Figure 7.1.5.6-1: Bandwidth Frequency Offset Field

Observation 7.1.5.6-1: A Bandwidth Frequency Offset value of 0x0000 0000 0010 0000 indicates a band offset of 1 Hz.

Observation 7.1.5.6-2: The value of the Bandwidth Frequency Offset field has a range of ±8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.6-1: The Context Packet Class documentation **should** specify the accuracy of the values reported in the Bandwidth Frequency Offset field.

7.1.5.7 Gain Field

The Gain field is used to describe the amount of signal gain or attenuation from the reference point to the output of a process. It can be used in conjunction with the Reference Level field to infer the signal level at various locations.

Rule 7.1.5.7-1: The Gain field **shall** express the gain from the Reference Point to the output of the process described by the IF Context Packet.

Rule 7.1.5.7-2: The value of the Gain field **shall** be expressed in units of decibels (dB). The Gain field **shall** use the 32-bit format shown in Figure 7.1.5.7-1. The upper 16-bits of this field are reserved and **shall** be set to zero. The Gain value **shall** be expressed in two's complement format in the lower 16-bits of this field. This field has an integer and a fractional part, with the radix point to the right of bit 7.

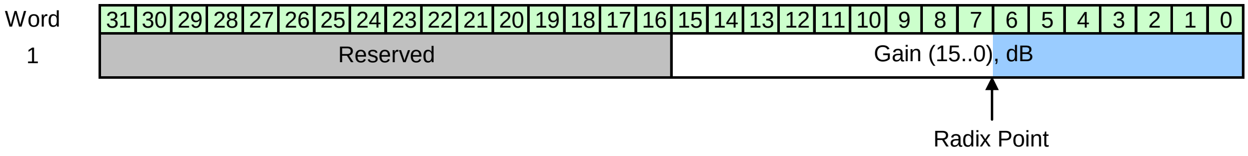


Figure 7.1.5.7-1: Gain Field

Observation 7.1.5.7-1: A Gain field value of 0x0000 0080 indicates a gain of +1 dB.

Observation 7.1.5.7-2: The value of the Gain field has a range of near ±256 dB with a resolution of 1/128 dB (0.0078 dB).

Observation 7.1.5.7-3: A process introducing attenuation will contain a negative Gain value.

Recommendation 7.1.5.7-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the Gain field.

7.1.5.8 Reference Level Field

The purpose of the Reference Level field is to relate the physical signal amplitude at the Reference Point (as indicated by the Reference Point ID) with the samples in IF Data Packet payload. The unit of measure for the Reference Level field is power, in dBm, since power is the preferred unit of measure when dealing with RF signals. The power value conveyed by the Reference Level field is the AC power of a sine wave at the Level Reference Point that results in a digitized sine wave with peak-to-peak amplitude of one, in the payload of the directly associated Data Packet Stream.

Observation 7.1.5.8-1: An N-bit fixed-point number has 2^N discrete values. For both signed and unsigned fixed point numbers, the difference between the highest and lowest value is 2^N-1 steps. This gives unsigned fixed point data items a range of zero to $1- 2^{-N}$ and signed fixed point data items a range of -½ to ½ - 2^{-N}. (See Section 6.1.5.4 for details).

Definition 7.1.5.8-1: A **unit-scale sinusoid** is a hypothetical sinusoid whose peak-to-peak amplitude is one in the data format used in the Data Items in the directly associated Data Packet Stream.

- For floating point Data Items, the unit-scale sinusoid is a sinusoid that has peak values of ± 0.5
- Although signed and unsigned fixed-point N-bit Data Item formats both have a range of $1-2^{-N}$, the unit-scale sinusoid for these data formats is assumed have peak-to-peak amplitude of one, just as in the case of floating point Data Items.
- Signed and unsigned N-bit Data Items are treated similarly to fixed-point N-bit Data Item formats. Data Items have a range of $1-2^{-M}$, where M is the mantissa size, but the unit-scale sinusoid for these data formats is assumed have a peak-to-peak amplitude of one, just as in the case for all other Data Item types.

Observation 7.1.5.8-1: As an example, a 4-bit 2s-complement fixed point data format has peak values of -½ and +7/16. A 4-bit unsigned data format has peak values of 0 and +15/16. Both have a range of 15/16 and both have a unit sinusoid peak-to-peak value of 1.0.

Rule 7.1.5.8-1: The Reference Level field **shall** express the AC power of a sine wave at the Reference Point that would result in a unit-scale sinusoid conveyed in the payload of the directly associated Data Packet Stream.

Permission 7.1.5.8-1: The **VRT equipment provider** may assume any line impedance for the purposes of converting Reference Level to signal voltage.

Rule 7.1.5.8-2: The Reference Point ID, when used, **shall** specify the location of Level Reference Point.

Rule 7.1.5.8-3: The value of the Reference Level field **shall** be expressed in units of decibels referenced to a milliwatt (dBm). The Reference Level field **shall** use the 32-bit format shown in Figure 7.1.5.8-1. The upper 16-bits of this field are reserved and **shall** be set to zero. The Reference Level value **shall** be expressed in two's complement format in the lower 16-bits of this field. This field has an integer and a fractional part, with the radix point to the right of bit 7.

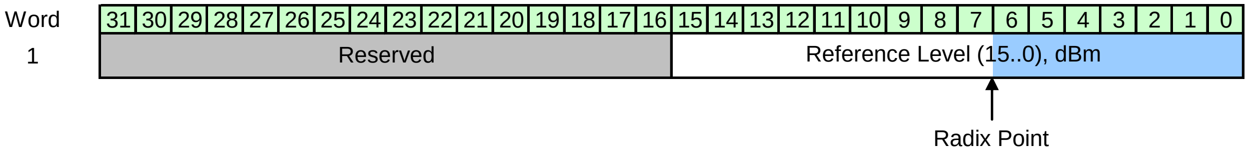


Figure 7.1.5.8-1: Reference Level Field

Observation 7.1.5.8-3: A Reference Level field value of 0x0000 0080 indicates a reference level of 1 dBm.

Observation 7.1.5.8-4: The value of the Reference Level field has a range of nearly ±256 dBm with a resolution of 1/128 dBm (0.0078 dBm).

Recommendation 7.1.5.8-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the Reference Level field.

7.1.5.9 Overrange Count Field

The Overrange Count field is used to convey the number of overrange samples in a single directly associated Data Packet.

Rule 7.1.5.9-1: The Overrange Count field **shall** contain the number of samples in the directly associated Data Packet whose amplitudes were beyond of the range of the data payload format.

Rule 7.1.5.9-2: For complex Cartesian data, the Overrange Count field **shall** contain the total number of complex samples in the directly associated Data Packet for which either the real or imaginary component was beyond of the range of the Data Item format.

Rule 7.1.5.9-3: The Overrange Count field **shall** use the 32-bit unsigned integer format shown in Figure 7.1.5.9-1.

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Overrange Count (31..0)																															

Figure 7.1.5.9-1: Overrange Count Field

Observation 7.1.5.9-1: The value of the Overrange Count field applies only to the directly associated Data Packet with the corresponding Timestamp. The overrange count does not accumulate over multiple Data Packets.

Rule 7.1.5.9-4: When conveying the Overrange Count, the Timestamp Epoch (TSM) field of the Context Packet may indicate either sample/picosecond or packet-level resolution. When sample level resolution is specified, the Context Packet Timestamp **shall** correspond to a sample time when the reported Overrange count was valid. That is, the Context Packet Timestamp **shall** be equal or later than the last sample in the directly associated Data Packet that was out of range, and equal to or before the last sample in the Data Packet. When packet-level resolution is specified, the Timestamp **shall** exactly match the timestamp of the Data Packet containing the Overrange samples.

7.1.5.10 Sample Rate Field

Rule 7.1.5.10-1: The Sample Rate field **shall** express the sample rate of the payload of the directly associated Data Packet Stream.

Rule 7.1.5.10-2: The value of the Sample Rate field **shall** be expressed in units of Hertz. The Sample Rate field **shall** use the 64-bit two's complement format shown in Figure 7.1.5.10-1. This field has an integer and a fractional part, with the radix point to the right of bit 20.

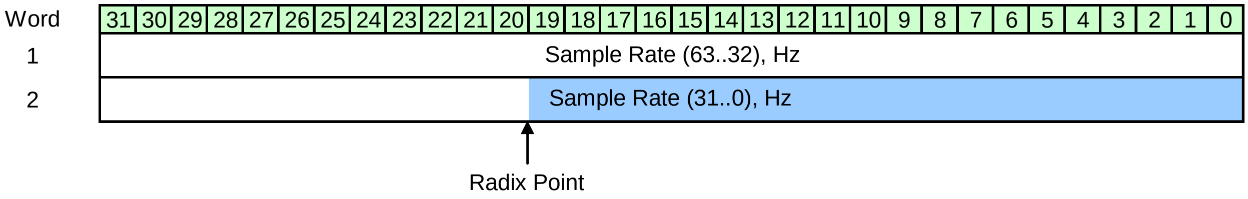


Figure 7.1.5.10-1: Sample Rate Field

Observation 7.1.5.10-1: A Sample Rate field value of 0x0000 0000 0010 0000 indicates a sample rate of 1 Hz.

Observation 7.1.5.10-2: The value of the Sample Rate field can range from 0.00 to 8.79 Terahertz with a resolution of 0.95 microhertz.

Recommendation 7.1.5.10-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the Sample Rate field.

7.1.5.11 Timestamp Adjustment Field

Typically, the purpose of a VRT processing chain is to extract some information from a signal detected by one or more sensors. Various consecutive forms of processing may occur upstream of the place where this information is finally extracted. Once it is extracted however, it may be important to calculate exactly when this information arrived at the sensor(s). Since processing usually adds delay, it is necessary to account for this delay in order to calculate the arrival time once the information has been extracted. The Timestamp Adjustment field is intended to express the cumulative delay through the various processes, from some reference point, e.g. an antenna, to the place where the extracted information resides.

Rule 7.1.5.11-1: The sum of the Timestamp field in a Data Packet and the most recently sent Timestamp Adjustment field in a directly associated Context Packet **shall** be equal to the time at which the information represented by the first data sample in that Data Packet was present at the Reference Point.

Observation 7.1.5.11-1: The Reference Point is indicated by the Reference Point Identifier field.

Rule 7.1.5.11-2: When the TSM bit is set to one, indicating that the Context Packet Timestamp has sample/picosecond accuracy, the time of the context changes indicated in a context packet is equal to the sum of the Timestamp and the most recently sent Timestamp Adjustment.

Permission 7.1.5.11-1: The Timestamp Adjustment field **may** be set to zero, as long as Rule 7.1.5.11-1 is not violated.

Permission 7.1.5.11-2: The Timestamp Adjustment field **may** be used to account for only a select subset of delays, as long as Rule 6.2.5.11-1 is not violated.

Observation 7.1.5.11-2: The Timestamp Adjustment field might only represent the cumulative analog delays upstream of an ADC. In this case the Timestamps in the Data Packets would be altered to reflect the inherent delays in the digital processing downstream of the ADC.

Observation 7.1.5.11-3: In general, the use of the Timestamp Adjustment field does not negate the need to recalculate Timestamps. For example, VRT packet size will typically vary from the input to the output of a process, necessitating the synthesizing of new timestamps. As another example, when re-sampling occurs, new Timestamps will usually be needed.

Rule 7.1.5.11-4: The Timestamp Adjustment field **shall** be expressed in units of picoseconds. The Timestamp Adjustment field **shall** use the 64-bit two's complement format shown in Figure 6.2.5.11-1.

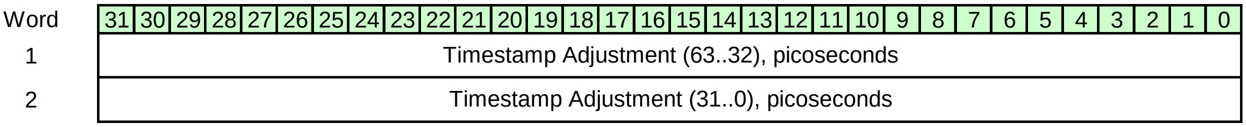


Figure 7.1.5.11-1: Timestamp Adjustment Field

Observation 7.1.5.11-3: The range of the Real-Time Timestamp Adjustment field is ±9.2 million seconds. The resolution of the Real-Time Timestamp Adjustment field is 1 picosecond.

Recommendation 7.1.5.11-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the Timestamp Adjustment field.

7.1.5.12 Temperature Field

The purpose of this field is to convey the temperature of some process, or process component, that may affect some aspect of the signal data.

Rule 7.1.5.12-1: The value of the Temperature field **shall** be expressed in units of degrees Celsius (°C). The Temperature field **shall** use the 32-bit format shown in Figure 7.1.5.12-1. The upper 16-bits of this field are reserved and **shall** be set to zero. The Temperature value **shall** be expressed in two's complement format in the lower 16-bits of this field. This field has an integer and a fractional part, with the radix point to the right of bit 6.

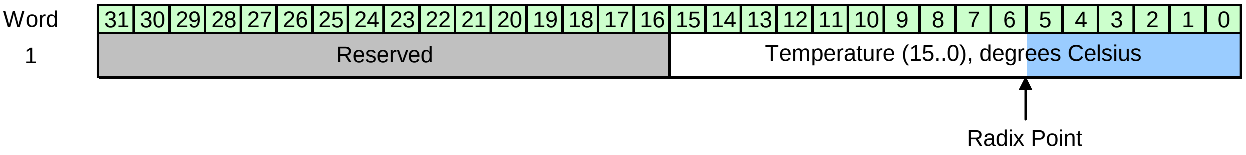


Figure 7.1.5.12-1: Temperature Field

Observation 7.1.5.12-1: The range of the Temperature field is ±511.984375 °C. The resolution of the Temperature field is 0.015625 °C (1/64 °C).

Observation 7.1.5.12-2: Temperature values below absolute zero are allowed but are not valid.

Recommendation 7.1.5.12-1: The Context Packet Class documentation **should** specify the accuracy of the value reported in the Temperature field.

7.1.5.13 Source Type Field

The Source Type field is used to identify the source of the signal referred to by this Context Packet. It contains a field to identify the **VRT equipment provider**, and a field to identify a specific device supplied by the **provider**.

Rule 7.1.5.13-1: The Source Type field **shall** use the format shown in Figure 7.1.5.13-1.

<table>
<tr><td>Word</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">Reserved</td><td colspan="24">OUI (23..0)</td></tr>
<tr><td>2</td><td colspan="16">Reserved</td><td colspan="16">Device Code (15..0)</td></tr>
</table>

Figure 7.1.5.13-1: Source Type Field

Note: Manufacturer Field specifies a ‘Class Code’, not ‘Device Code’. Do we want these to be the same or different?

Rule 7.1.5.13-2: The OUI subfield **shall** contain the 24-bit field used to store the IEEE-registered Organizationally Unique Identifier (company identifier) of the provider of the product which generated the Context Packet. The msb of the OUI **shall** reside in the left-most position in the field. If the OUI is unknown, this field **shall** be set to zero.

Rule 7.1.5.13-3: The Device Code subfield **shall** contain a 16-bit number to identify the device generating the Context Packet. The msb of the Device Code **shall** reside in the left-most position in the field. For a given manufacturer, the Device Code shall be unique for each VRT compliant device.

Recommendation 7.1.5.13-1: The **VRT equipment provider should** maintain a complete list of all VRT Source Type Device Codes.

7.1.5.14 State and Event Indicator Field

The State and Event Indicator field is used for the communication of binary indications. It can also be used to communicate a limited number of non-binary state indications. It contains five predefined Indicator bits, each with a corresponding enable bit, which controls whether or not the indicator bit is valid. There are also a number of bits reserved for future specification, and eight user-defined bits.

Rule 7.1.5.14-1: The form of the State and Event Indicator field shall follow that shown in Figure 7.1.5.14-1, shown in greater detail in Table 7.1.5.14-1.

<table>
<tr><td>Word</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="12">Enables</td><td colspan="12">Indicators</td><td colspan="8">User-Defined</td></tr>
</table>

Figure 7.1.5.14-1: State and Event Indicator Fields

Bit Position	Bit Function
31	Calibrated Time Enable
30	Valid Data Enable
29	Reference Lock Enable
28	AGC/MGC Enable
27	Detected Signal Enable
26	Spectral Inversion Enable
[25..20]	Reserved Enables
19	Calibrated Time Indicator
18	Valid Data Indicator
17	Reference Lock Indicator
16	AGC/MGC Indicator
15	Detected Signal Indicator
14	Spectral Inversion Indicator
[13..8]	Reserved Indicators
[7..0]	User-defined

Table 7.1.5.14-1: State and Event Indicator Bit Definitions

Rule 7.1.5.14-2: For the predefined Indicators, when an Indicator Enable bit is set to one, the corresponding Indicator bit **shall** function as indicated in Table 7.1.5.14-1. Otherwise the meaning of the corresponding Indicator bit is undefined.

Rule 7.1.5.14-3: When an Enable bit is set to zero in an IF Context Packet, the assumed state of the corresponding Indicator bit **shall** be the state most recently communicated by some previous Context Packet.

Permission 7.1.5.14-1: The User-Defined bits **may** be used for any purpose. They **may** be used individually as well as in groups that convey more than two states.

Rule 7.1.5.14-4: The Context Packet Class documentation **shall** provide a complete specification of all “user-defined” Indicators in use, along with the binary codes used.

Observation 7.1.5.14-1: The eight bits provided as user-defined Indicators **may** be used to indicate eight independent binary states, or up to 256 mutually-exclusive states, or any combination of independent and mutually-exclusive states that can be represented in the 8-bit field.

Rule 7.1.5.14-5: The Indicators used in an IF Context Packet **shall** apply to the Data Packet Stream that is directly associated with the Context Packet sending the Indicators.

Rule 7.1.5.14-6: The Calibrated Time Indicator **shall** function just as it does in the Data Packet trailer.

Rule 7.1.5.14-7: The Valid Data Indicator **shall** function just as it does in the Data Packet trailer.

Rule 7.1.5.14-8: The Reference Lock Indicator **shall** function just as it does in the Data Packet trailer.

Rule 7.1.5.14-9: The AGC/MGC Indicator **shall** function just as it does in the Data Packet trailer.

Rule 7.1.5.14-10: The Detected Signal Indicator **shall** function just as it does in the Data Packet trailer.

Rule 7.1.5.14-11: The Spectral Inversion Indicator **shall** be set to one when the spectrum of the referred signal is inverted with respect to the spectrum at the Reference Point, and set to zero otherwise.

Obersvation 7.1.5.14-1: The Reference Point for spectral inversion is specified by the Reference Point ID field.

Rule 7.1.5.14-12: The Indicator and Enable bits that are reserved **shall** be set to zero.

7.1.5.15 Data Packet Payload Format Field

The Data Packet Payload Format field shall consist of two 32-bit words containing parameters that indicate both the packing and content of the data. Table 6.2.5.15-1 shows the organization of the parameter fields in the first payload format word, and Table 6.2.5.15-2 shows the placement of parameters in the second payload format word.

Rule 6.2.5.15-1: When the corresponding context indicator bit is set, the Data Packet Payload Format field **shall** contain the parameters shown in Table 6.2.5.15-1 and Table 6.2.5.15-2, in the locations indicated.

Bit Position	Field	Field Width (bits)
[31]	Packing Method	1
[30..25]	Item Packing Field Size	6
[24..19]	Data Item Size	6
[18..16]	Event-tag Size	3
[15..12]	Channel-tag Size	4
[11..8]	Reserved (Set to zero)	4
[7..0]	Vector Size	8

Table 6.2.5.15-1: Location of parameter fields in the first payload format word

Bit Position	Field	Field Width (bits)
[31..16]	Repeat Count	16
[15]	Sample-repeat Indicator	1
[14..7]	Reserved (Set to zero)	8
[6-5]	Real/Complex Type	2
[4..0]	Data Item Format	5

Table 6.2.5.15-2: Location of parameter fields in the second payload format word

Rule 6.2.5.15-2: The “Packing Method” field **shall** be set to one when Link-Efficient packing is used in the directly associated Data Packet Stream. It **shall** be set to zero when processing-efficient packing is used.

Rule 6.2.5.15-3: The “Item Packing Field Size” field **shall** contain an unsigned number that is one less than the actual Item Packing Field size used in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-4: The “Data Item Size” field **shall** contain an unsigned number that is one less than the actual Data Item size in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-5: The “Event-Tag Size” field **shall** contain an unsigned number equal to the Event-Tag size used in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-6: The “Channel-Tag Size” field **shall** contain an unsigned number equal to the Channel-Tag size used in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-7: The “Repeat Count” field **shall** contain an unsigned number that is one less than the actual Repeat Count used in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-8: The “Vector Size” field **shall** contain an unsigned number that is one less than the actual Vector size in the diretly associated Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 6.2.5.15-9: The Data Item Format field **shall** contain the appropriate 5-bit code to indicate the type of Data Items used in the diretly associated Data Packet Stream. The code **shall** be chosen according to Table 6.2.5.15-3.

Code	Data Item Type	Code	Data Item Type
00000	Unsigned Fixed-point	10000	Signed Fixed-point
00001	Unsigned VRT, 1-bit exponent	10001	Signed VRT, 1-bit exponent
00010	Unsigned VRT, 2-bit exponent	10010	Signed VRT, 2-bit exponent
00011	Unsigned VRT, 3-bit exponent	10011	Signed VRT, 3-bit exponent
00100	Unsigned VRT, 4-bit exponent	10100	Signed VRT, 4-bit exponent
00101	Unsigned VRT, 5-bit exponent	10101	Signed VRT, 5-bit exponent
00110	Unsigned VRT, 6-bit exponent	10110	Signed VRT, 6-bit exponent
00111	Reserved	10111	Reserved
01000	Reserved	11000	Reserved
01001	Reserved	11001	Reserved
01010	Reserved	11010	Reserved
01011	Reserved	11011	Reserved
01100	Reserved	11100	Reserved
01101	Reserved	11101	IEEE-754 Single-Precision Floating-Point
01110	Reserved	11110	IEEE-754 Double-Precision Floating-Point
01111	Reserved	11111	Other

Table 7.1.5.15-3: Data Item Format Codes

Rule 6.2.5.15-10: The Sample-repeat Indicator **shall** be set to one when sample repeating is in use in the diretly associated Data Packet Stream. Otherwise it **shall** be set to zero.

Rule 6.2.5.15-11: The Real/Complex Type Indicator **shall** indicate whether the Data Samples are real or one of the complex types using the appropriate code indicated in Table 7.1.5.15-4.

Code	Data Sample Type
00	Real
01	Complex, Cartesian
10	Complex, Polar, Signed Phase
11	Complex, Polar, Unsigned Phase

Table 7.1.5.15-4: Data Sample Format Codes

7.1.5.16 Formatted GPS (Global Positioning System) Geolocation Field

The GPS (Global Positioning System) and INS (Inertial Navigation System) Geolocation fields share the same format, shown below in Figure 7.1.5.16-1.

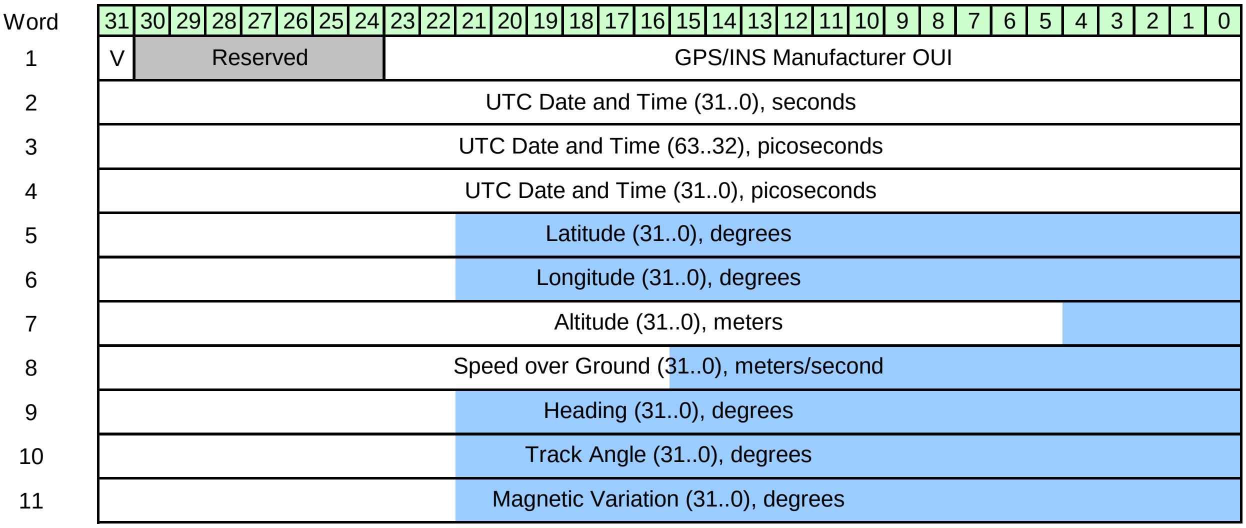


Figure 7.1.5.16-1: Formatted GPS and INS Geolocation Fields

Rule 7.1.5.16-1: The Formatted GPS Geolocation field **shall** be formatted as shown in Figure 6.2.5.16-1.

Rule 7.1.5.16-2: The Valid ("V") subfield **shall** be set to one to indicate that the position information within the GPS/INS Geolocation field is valid and **shall** be set to zero otherwise.

Rule 7.1.5.16-3: The Manufacturer OUI subfield **shall** contain the 24-bit field for the IEEE-assigned Organizationally Unique Identifier (company identifier) of the GPS/INS manufacturer. If the Manufacturer OUI is unknown, this field **shall** be set to zero.

Rule 7.1.5.16-4: The UTC Date and Time subfield **shall** express the date and time of the location fix. The UTC Date and Time subfield **shall** be presented in the format of the Real-Time Timestamp described in Section 6.1.4.3.

Observation 7.1.5.16-1: The timestamp in the Geolocation UTC Date and Time subfield will generally differ from the timestamp in the header of the Context Packet.

Definition 7.1.5.16-1: The **Geolocation Angle Format** describes angles in units of degrees. The Geolocation Angle Format uses the 32-bit two's complement format shown in Figure 7.1.5.16-2. This field has an integer and a fractional part, with the radix point to the right of bit 22.

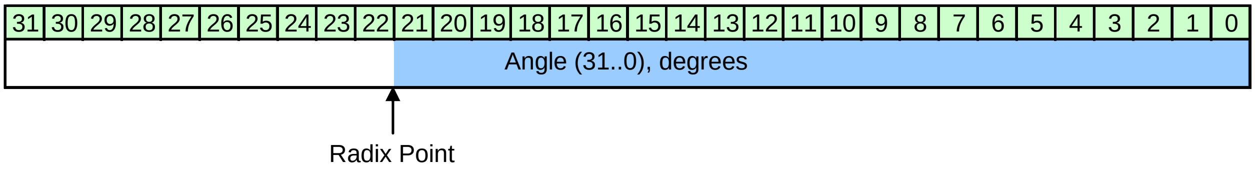


Figure 7.1.5.16-2: Geolocation Angle Format

Observation 7.1.5.16-2: The Geolocation Angle Format has a possible range of ±512 degrees and a resolution of 2.38e-7 degrees. Particular angular measurements will allow various ranges such as 0 to 360 degrees, ±180 degrees, or ±90 degrees. This format is used for several other fields within the GPS Geolocation field, including latitude, longitude, heading, and track angle.

Observation 7.1.5.16-3: On the surface of the Earth, the latitude and longitude angular resolution provides a Cartesian resolution on the order of a few centimeters.

Rule 7.1.5.16-5: The Latitude and Longitude subfields **shall** use the Geolocation Angle Format shown in Figure 7.1.5.16-2.

Rule 7.1.5.16-6: The Latitude subfield value **shall** range from –90.0 (South) to +90.0 (North) degrees.

Rule 7.1.5.16-7: The Longitude subfield value **shall** range from –180.0 (West) to +180.0 (East) degrees.

Rule 7.1.5.16-8: The Altitude subfield **shall** use the 32-bit two's complement format shown in Figure 7.1.5.16-3. The value of the Altitude subfield **shall** be expressed in units of meters. This field has an integer and a fractional part, with the radix point to the right of bit 5.

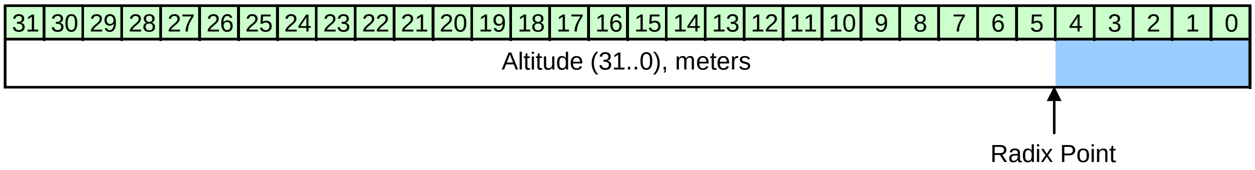


Figure 7.1.5.16-3: Altitude Subfield Format

Observation 7.1.5.16-4: The Altitude subfield has a range of ±67108 kilometers and a resolution of 3.1 centimeters.

Rule 7.1.5.16-9: The Context Packet Class documentation **shall** specify the Altitude reference. For GPS hardware, this will typically be height above the WGS-84 ellipsoid. For INS hardware, this will typically be height above mean sea level.

Rule 7.1.5.16-10: The Speed Over Ground subfield **shall** use the 32-bit two's complement format shown in Figure 7.1.5.16-4. The value of the Speed Over Ground subfield **shall** be expressed in units of meters per second. This field has an integer and a fractional part, with the radix point to the right of bit 16.

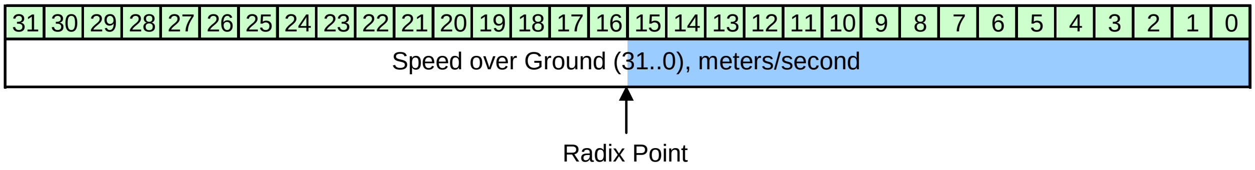


Figure 7.1.5.16-4: Speed Over Ground Subfield Format

Observation 7.1.5.16-5: The Speed Over Ground subfield has a range of 0 to 65636 m/s and a resolution of 1.5e-5 m/s.

Rule 7.1.5.16-11: The Heading Angle subfield **shall** use the Geolocation Angle Format shown in Figure 7.1.5.16-2. The Heading Angle **shall** express the platform's orientation with respect to true North in decimal degrees. The Heading angle value **shall** range from 0.0 to +359.999999761582 degrees.

Observation 7.1.5.16-6: The Heading Angle may not be provided by GPS hardware. If this information is not available, this field may be filled with the default value 0xFFFFFF**.**

Rule 7.1.5.16-12: The Track Angle subfield **shall** use the Geolocation Angle Format shown in Figure 7.1.5.16-2. The Heading Angle **shall** express the platform's direction of travel with respect to true North in decimal degrees. The Bearing angle value **shall** range from 0.0 to +359.999999761582 degrees.

Observation 7.1.5.16-7: The Heading Angle reflects the direction the platform is pointed at, whereas the Track Angle reflects the direction the platform is moving. These two angles may not be the same. For example, external forces on the platform such as wind or water currents may cause the Track Angle to differ from the Heading Angle.

Rule 7.1.5.16-13: The Magnetic Variation subfield **shall** use the Geolocation Angle Format shown in Figure 6.2.5.15-2. The Magnetic Variation subfield **shall** express magnetic variation from true North in decimal degrees. The Magnetic Variation value **shall** range from -180.0 (West) to +180.0 (East) degrees.

Rule 7.1.5.16-14: The GPS/INS Manufacturer OUI subfield **shall** take the value 0xFFFFFF when unspecified.

Rule 7.1.5.16-15: The UTC Date and Time subfields **shall** take the value 0xFFFFFFFF when unspecified.

Rule 7.1.5.16-16: The remaining Geolocation subfields **shall** take the value 0x7FFFFFFF when unspecified. This includes the Latitude, Longitude, Altitude, Speed Over Ground, Heading, Track Angle and Magnetic Variation words.

Observation 7.1.5.16-8: These default subfield values are invalid or practically impossible for all GPS and INS Geolocation subfields.

7.1.5.17 Formatted INS (Inertial Navigation System) Geolocation Field

The Formatted INS (Inertial Navigation System) Geolocation field shares the same format as the GPS Geolocation field described above. Both GPS and INS data may be included in the same Context Packet.

Rule 7.1.5.17-1: The Formatted INS Geolocation field **shall** follow the same rules as the Formatted GPS Geolocation field.

7.1.5.18 ECEF (Earth-Centered, Earth-Fixed) Ephemeris Field

The ECEF Ephemeris field provides a format to convey location in Earth-Centered, Earth-Fixed Cartesian coordinates. It also contains a Cartesian decomposition of velocity and attitude. Figure 7.1.5.18-1 shows the interpretation of the various components of ECEF location.

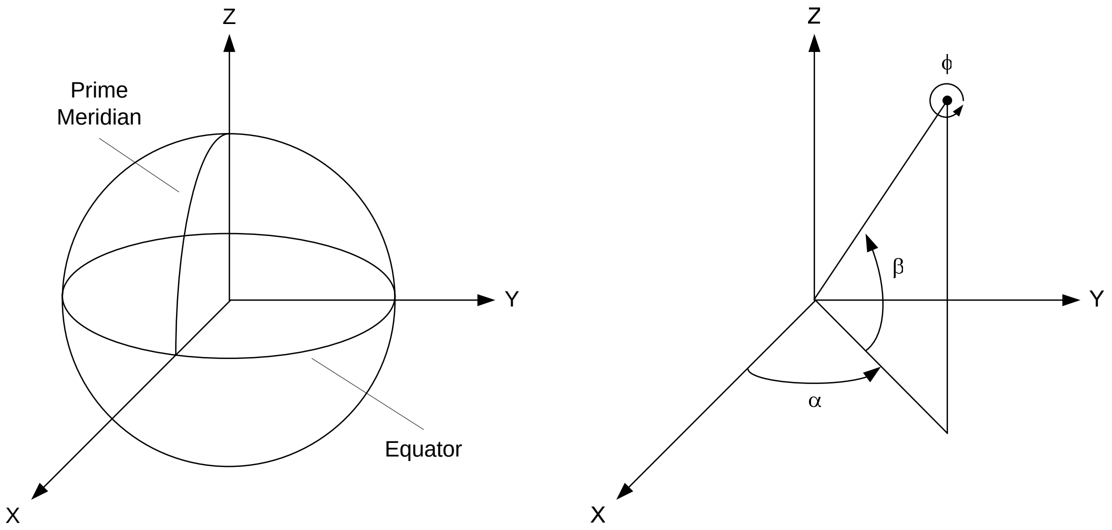


Figure 7.1.5.18-1: ECEF Ephemeris Coordinates

Rule 7.1.5.18-1: The position and velocity coordinates of the ECEF Ephemeris fields **shall** be specified in the Earth Centered, Earth Fixed Coordinate System, as shown in Figure 7.1.5.18-1, interpreted as follows:

- The XY-plane is the Earth's equator.
- The Z-axis is directed along the Earth's rotational axis and is positive north.
- The positive X-axis is the intersection of the planes defined by the equator and prime meridian.
- The Y-axis completes a right-handed orthogonal system, 90 degrees east of the X-axis.
- The X, Y, and Z coordinates are referenced to the center of mass of the WGS-84 ellipsoid.

Rule 7.1.5.18-2: The attitude coordinates of the ECEF Ephemeris fields **shall** be specified in the Earth Centered, Earth Fixed Coordinate System, as shown in Figure 7.1.5.18-1, interpreted as follows:

- The angle alpha is about the Z-axis. Positive rotation is X to Y, with alpha equal to zero at the X-axis.
- The angle beta is about the Y-axis. Positive rotation is X to Z, with beta equal to zero at the XY-plane.
- The angle phi is about the X-axis. Positive rotation is Y to Z with phi equal to zero at the Y-axis.

The ECEF (Earth-Centered, Earth-Fixed) field and Relative Geolocation field (See Section 6.2.5.19) share the same format, shown below.

Rule 7.1.5.18-3: The ECEF Ephemeris field **shall** be expressed using the format shown in Figure 7.1.5.18-2.

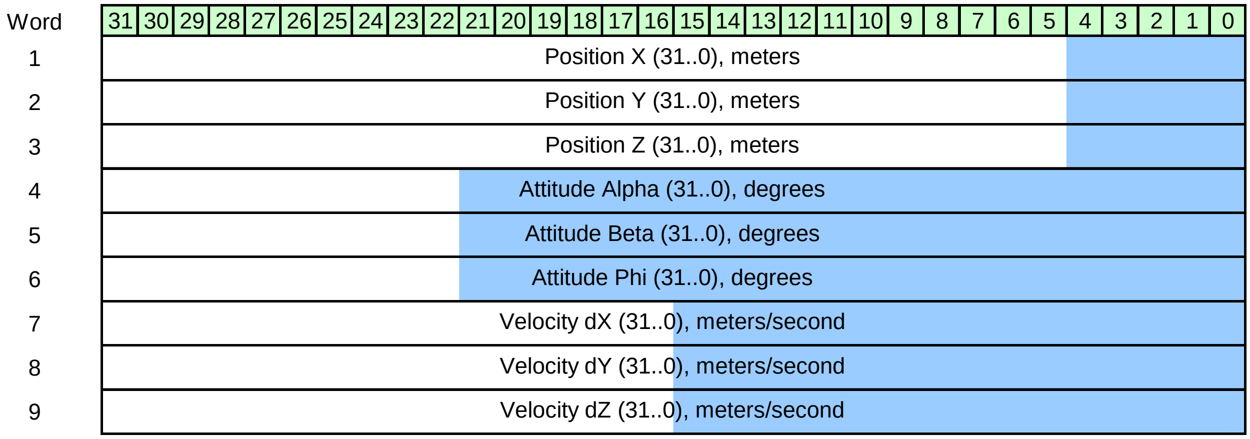


Figure 7.1.5.18-2: ECEF and Relative Ephemeris Fields

Figure 7.1.5.18-3 shows the position, velocity, and attitude coordinate formats in greater detail.

Rule 7.1.5.18-4: The position coordinates of the ECEF Ephemeris field **shall** use the 32-bit two’s complement “Position” format shown in Figure 7.1.5.18-3. The position values **shall** be expressed in units of meters. This field has an integer and a fractional part, with the radix point to the right of bit 5.

Rule 7.1.5.18-5: The attitude coordinates of the ECEF Ephemeris field **shall** use the 32-bit two’s complement “Attitude” format shown in Figure 7.1.5.18-3. The attitude values **shall** be expressed in units of decimal degrees. This field has an integer and a fractional part, with the radix point to the right of bit 22.

Rule 7.1.5.18-6: The velocity coordinates of the ECEF Ephemeris field **shall** use the 32-bit two’s complement “Velocity” format shown in Figure 7.1.5.18-3. The velocity values **shall** be expressed in units of meters per second. This field has an integer and a fractional part, with the radix point to the right of bit 16.

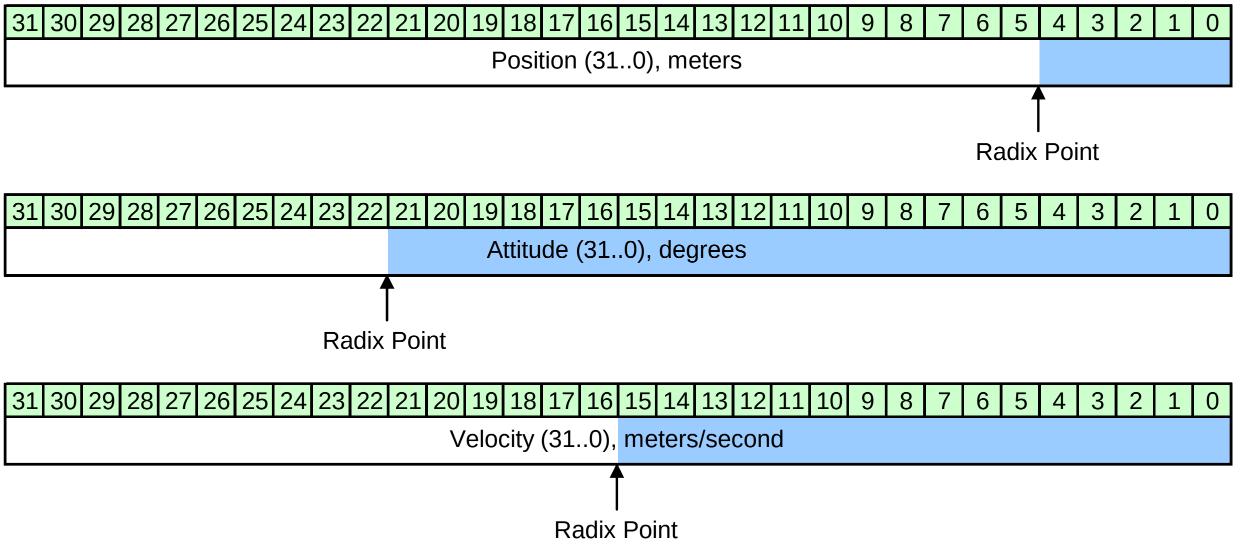


Figure 7.1.5.18-3: Position, Attitude and Velocity Coordinate Formats

Rule 7.1.5.18-7**:** Each word of the ECEF Ephemeris field **shall** take the value 0x7FFF FFFF when unknown.

7.1.5.19 Relative Ephemeris Fields

The Relative Ephemeris Geolocation field shares the same format as the ECEF Ephemeris field, described above. However, the ECEF Ephemeris coordinate system is always the Earth-Centered Earth-Fixed coordinate system whereas the center of the Relative Ephemeris coordinate system is user-defined. The ECEF and Relative Ephemeris coordinate systems may be used together if the translation between coordinate systems is specified.

The Relative Ephemeris field provides a format to convey relative location, velocity, and attitude in Cartesian coordinates. This field may be useful, for example, in applications where it is important to know the locations and attitudes of multiple antennas on a platform. The reference point and axis orientation for the Relative Ephemeris coordinate system are **equipment provider** specified. If desired, the Relative Ephemeris reference point may be given in the ECEF Ephemeris field of the process pointed to by the Ephemeris Reference Identifier described in Section 7.1.5.20.

Rule 6.2.5.19-1: The Relative Ephemeris field **shall** be expressed using the formats shown in Figure 7.1.5.18-2 and Figure 7.1.5.18-3.

Rule 6.2.5.19-2: When the Relative Ephemeris field is used in conjunction with the ECEF Ephemeris field. The Context Packet Class documentation **shall** provide the translation between the two coordinate systems.

Observation 6.2.5.19-1: The Relative Ephemeris field can be used on its own. It is not necessary to use the ECEF Ephemeris field in conjunction with the Relative Ephemeris field. As an example, the **VRT equipment provider** may designate the Relative Ephemeris x-axis to be along the heading of an aircraft and the Relative Ephemeris y-axis to be in the plane created by the aircraft's wings. The locations of two antennas on the aircraft are specified in the Relative Ephemeris coordinates. Figure 7.1.5.19-1 shows this example in two dimensions.

Observation 6.2.5.19-2: The Relative Ephemeris values may be calculated in ECEF coordinates by specifying the origin and orientation of the Relative Ephemeris reference with respect to the ECEF coordinate system. For example shown in Figure 7.1.5.19-1, the Relative Ephemeris origin could be at the GPS antenna, which may be near the tail of the aircraft, and the x-axis may be oriented along the heading of the aircraft. Knowing this, the locations of the two antennas in ECEF coordinates may be calculated.

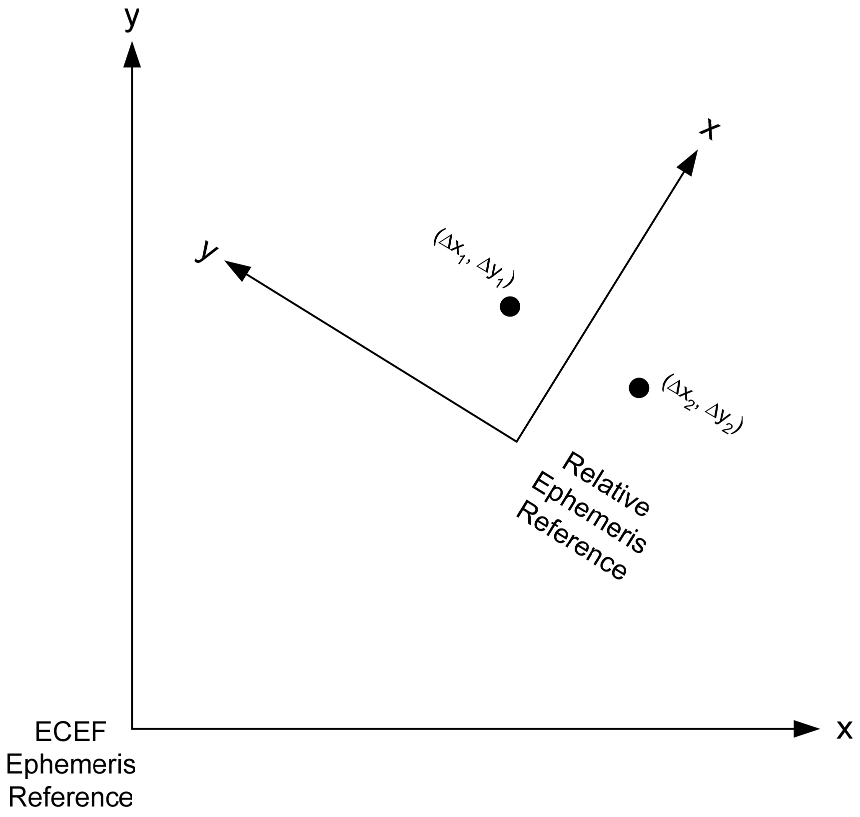


Figure 7.1.5.19-1: ECEF and Relative Ephemeris Translation Example
The Relative Ephemeris coordinate system has its origin at the GPS antenna near the rear of the aircraft. The locations of the two antennas, one on each wing, are reported in VRT packets as Relative Ephemeris, i.e. relative to the GPS antenna location.

Observation 6.2.5.19-2: The attitude coordinates can take on more than one meaning, as demonstrated in Table 7.1.5.19-1. The aircraft orientation is referenced to the aircraft direction of travel along the X-axis. The antenna orientation is referenced to a default antenna boresight along the X-axis, which places the antenna aperture in the YZ-plane.

	Aircraft Orientation	**Antenna Orientation**
Alpha	Yaw	Azimuth
Beta	Pitch	Elevation
Phi	Roll	Polarization

Table 7.1.5.19-1: Two Common Attitude Coordinate Interpretations

7.1.5.20 Ephemeris Reference Identifier

When Relative Ephemeris is reported, the Ephemeris Reference Identifier serves to identify the process whose location serves as the origin for the Relative Ephemeris. The Ephemeris Reference ID uses the format shown in Figure 7.1.5.20-1.

<table>
<tr><td>Word</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="32">Ephemeris Reference Identifier</td></tr>
</table>

Figure 7.1.5.20-1: Ephemeris Reference Identifier Format

Rule 7.1.5.20-1: The Ephemeris Reference Identifier, when used, **shall** contain the Stream ID of the process whose ECEF Ephemeris is necessary to translate the Relative Ephemeris given in this Context Packet Stream to ECEF coordinates.

7.1.5.21 GPS ASCII Field

Some GPS devices output their information in the form of formatted ASCII strings, known as GPS "sentences." The sentences from a GPS device that emits ASCII strings (such as an NMEA-0183 compliant GPS device) may be conveyed in their original ASCII format using this field.

Rule 7.1.5.21-1: The GPS ASCII field **shall** follow the format shown in Figure 7.1.5.21-1.

<table>
<tr><td>Word</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">Reserved</td><td colspan="24">GPS/INS Manufacturer OUI</td></tr>
<tr><td>2</td><td colspan="32">Number of Words</td></tr>
<tr><td>3</td><td colspan="8">Byte 1</td><td colspan="8">Byte 2</td><td colspan="8">Byte 3</td><td colspan="8">Byte 4</td></tr>
<tr><td>⋮</td><td colspan="8">⋮</td><td colspan="8">⋮</td><td colspan="8">⋮</td><td colspan="8">⋮</td></tr>
<tr><td>N+2</td><td colspan="8">Byte 4N-3</td><td colspan="8">Byte 4N-2</td><td colspan="8">Byte 4N-1</td><td colspan="8">Byte 4N</td></tr>
</table>

Figure 7.1.5.21-1: GPS ASCII Field

Rule 7.1.5.21-2: The GPS Manufacturer OUI subfield **shall** contain the 24-bit field for the IEEE-assigned Organizationally Unique Identifier (company identifier) of the GPS manufacturer. If the Manufacturer OUI is unspecified, this field **shall** be set to 0xFFFFFF.

Rule 7.1.5.21-3: The Number of Words subfield **shall** be represented as a 32-bit unsigned integer. It **shall** express the total number of 32-bit words required to convey the ASCII sentences.

Observation 7.1.5.21-1: The total number of words in the GPS ASCII field will be the number contained in the "Number of Words" subfield plus two.

Rule 7.1.5.21-4: The GPS ASCII Sentence subfield (words 3…N+2 in the above figure) **shall** only contain complete ASCII sentences, such as defined in NMEA-0183 or other valid GPS output format**.**

Permission 7.1.5.21-1: Multiple ASCII sentences **may** be concatenated and sent in a single Context Packet.

Rule 7.1.5.21-5: The GPS ASCII field **shall** be padded with null characters so that the total number of ASCII sentence characters plus null characters equals four (4) times the value in the Number of Words subfield.

7.1.5.22 Stream Association Lists Section

The Stream Association Lists section in the IF Context Packet Class contains three Stream Association Lists. Each Stream Association List supports associating in slightly different kinds of metadata. All three

Stream Association Lists are identical in the way in which they provide associations between context packets. A Context Packet Stream is “associated” to another Context Packet Stream by including its Stream ID in one of the Stream Association Lists of the other Context Packet Stream.

Figure 7.1.5.22-1 shows the organization of the Stream Association Lists section of the Context Packet. The first two 32-bit words contain the sizes of the three Stream Association Lists. The lists themselves follow the words containing the list sizes.

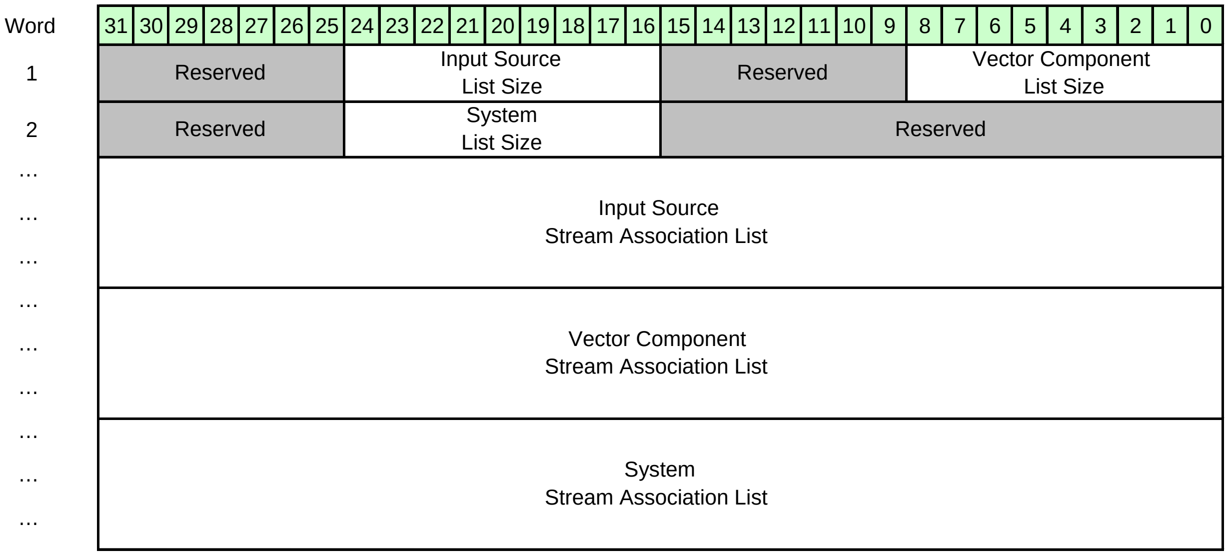


Figure 7.1.5.22-1: The format of the Stream Association Lists Section of the Context Packet.

For each Stream Association List there is a “List Size” which indicates the number of included Stream IDs.

Rule 7.1.5.22-1: The organization of the Stream Association Lists section **shall** be as shown in Figure 7.1.5.22-1.

Rule 7.1.5.22-2: Each List Size field **shall** be a binary number ranging from zero to 511 containing the number of Stream IDs in the corresponding Stream Association List. The lsb **shall** be the rightmost bit in the field.

Rule 7.1.5.22-3: If the value in the List Size field is zero, then the corresponding Stream Association List **shall** be omitted.

Rule 7.1.5.22-4: The “Reserved” bits shown in Figure 7.1.5.22-1 **shall** be set to zero.

Rule 7.1.5.22-5: Stream Identifiers entered in the Stream Association Lists **shall** be 32-bits long.

Rule 7.1.5.22-6: Any Context Packet Stream to be contextually associated to a Data Packet Stream **shall** be associated by listing their Stream IDs in one of the Stream Association Lists of an IF Context Packet Stream that is either directly or contextually associated to the Data Packet Stream.

Observation 7.1.5.22-1: The Stream ID list method does not inherently limit the length of the chain of context associations.

Observation 7.1.5.22-2: All of the associations between Context Packet Streams will not be fully communicated to the receiver until at least one packet from each stream, with the required Stream Association Lists present, has been received. The time required for full communication of the associations is therefore dependent on the frequency with which the Stream Association Lists are included in the emitted Context Packets.

Permission 7.1.5.22-1: Stream Association Lists **may** be used to associate together multiple Context Packet Streams without any of them being associated to a Data Packet Stream.

7.1.5.22.1 Input Source Stream Association List

It is often desirable to communicate metadata related to the sequence of processing steps that produced a data stream. For example, for a system where a tuner is followed by a DDC, which is followed by a demodulator, it may be desirable to associate metadata such as the center frequency of the tuner and DDC bandwidth to the demodulated data stream. The Input Source Stream Association List provides this capability. See Appendix **B.5** for an example using this list.

Definition 7.1.5.22.1-1: An upstream process is a process whose output data affects the output data of other (downstream) processes.

Definition 7.1.5.22.1-2: An Input Source ID is the Stream Identifier for an IF Context Packet Stream that provides information related to the immediately upstream process.

Rule 7.1.5.22.1-1: The Input Source Stream Association List in an IF Context Packet Stream associated directly to a Data Packet Stream **shall** associate in only Context Packet Streams related to the process that is immediately upstream of the process generating the corresponding data stream.

Rule 7.1.5.22.1-2: Whenever an IF Context Packet Stream contains an Input Source Stream Association List, those Context Packet Streams associated by that list **shall** relate to processes that are immediately upstream of the process associated with the IF Context Packet Stream containing the list.

Observation 7.1.5.22.1-1: When an IF Context Packet Stream is directly associated to a Data Packet Stream it is typically the case that the resulting Context Packets contain metadata relating to the process generating the data stream, as well as metadata explaining the data stream itself. In such cases, chains of associated Context Packet Streams, as shown in Figure 7.1.5.22-1, may simultaneously pertain both to a chain of data streams and to a chain of processes generating those streams.

Rule 7.1.5.22.1-3: Whenever metadata regarding an immediate upstream process affecting a data stream is to be conveyed via a Context Packet Stream, the Context Packet Stream **shall** be contextually associated to the Data Packet Stream using the Input Source Stream Association List. This **shall** hold except when the metadata is related to vector components, in which case the Vector Components Stream Association List **shall** be used.

Observation 7.1.5.22.1-2: An entire processing architecture may be represented by a collection of IF Context Packet Streams that are associated via Input Source Stream Association Lists in a way that mirrors the signal flow.

7.1.5.22.2 Vector Component Stream Association List

When a Data Packet Stream contains sample vectors, each component of these vectors may have metadata associated with it. The Vector Component Stream Association List provides a method of communicating the metadata for each vector component. This is accomplished by associating an additional Context Packet Stream for each vector component. The Vector Component Stream Association List is an ordered list of N Stream IDs, where N is the dimension of the sample vectors in the associated Data Packet Stream. The first Stream ID in the list associates a Context Packet Stream containing metadata for the first component of the vector. The second Stream ID similarly associates metadata for the second component, and so on to the last Stream ID in the list.

Rule 7.1.5.22.1-1: If the Vector Component Stream Association List is used in an IF Context Packet Stream directly associated to that Data Packet Stream, then this Vector Component Stream Association List, when present, **shall** have exactly *N* entries, where *N* is the dimension of the vectors contains in the Data Packet Stream.

Rule 7.1.5.22.11-2: The Vector Component Stream Association List **shall not** contain any Stream IDs other than the Stream IDs required to association in metadata for each vector component.

Rule 7.1.5.22.1-3: The ordering of the Vector Component Stream IDs in the Vector Component Stream Association List, **shall** exactly match the ordering of vector components in the directly associated Data Packet Stream.

7.1.5.22.3 System Stream Association List

The System Stream Association List facilitates contextual association to a data stream of additional metadata that is not related to vector components or the process immediately upstream in the processing chain. This additional metadata may be any type of metadata, and may be sent in an IF Context Packet or in an Extension Context Packet. For example, the directly associated IF Context Packet may contain a Stream ID in the System Stream Association List that associates in a Context Packet Stream whose only purpose is to report the temperature of the chassis power supply. As another example, an Extension Context Packet might contain the name of the current operator of the system.

Rule 7.1.5.22.3-1: The System Stream Association List **shall not** be used to associate in Context Packet Streams that can be correctly associated and interpreted via the Input Source Stream Association List or the Vector Component Stream Association List.

7.2 *Extension Context Packet Class Specifications*

Extension Context Packet Streams are intended to be used to communicate metadata for which no provision has been made in the IF Context Packet. Figure 7.1.5.22-1 shows the organization of the Extension Context Packet Class.

Rule 7.2-1: The order of the fields in an Extension Context Packet Class **shall** be as shown in Figure 7.1.5.22-1.

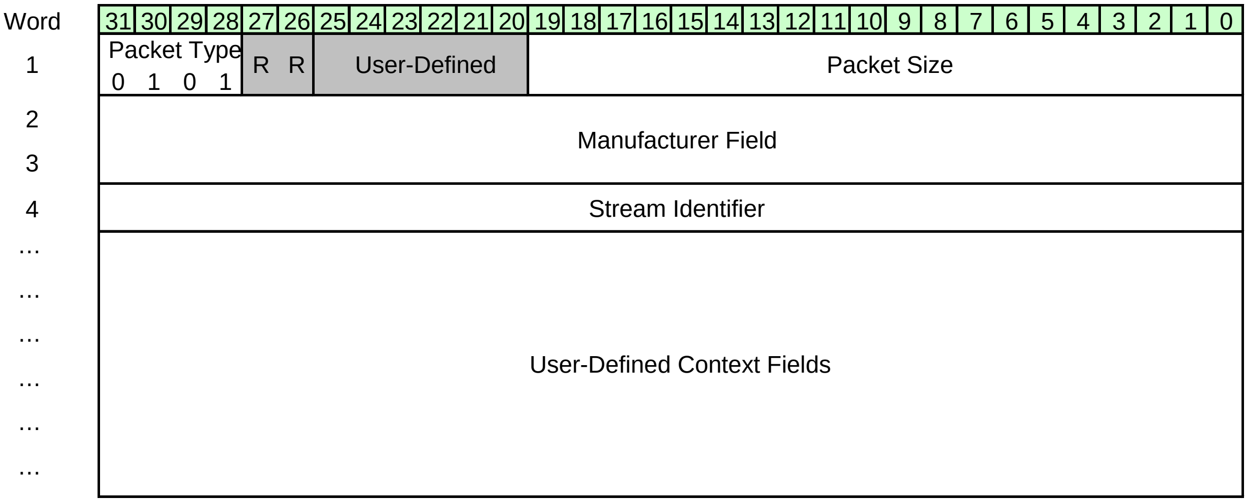


Figure 7.1.5.22-1: The format of the Extension Context Packet Class

Rule 7.2-2: An Extension Context Packet **shall** be an integer number of 32-bit words in size.

Rule 7.2-3: The Extension Context Packet Class **shall** include the Packet Type and Packet Size fields with the same functionality and in the same locations as the IF Context Packet.

Observation 7.2-1: The Extension Context Packet Class may or may not have the Packet Count and TSI fields of the IF Context Packet Class.

Rule 7.2-4: The Extension Context Packet Class **shall** contain the binary value “0101” in the Packet Type field.

Rule 7.2-5: The Manufacturer field in Extension Data Packets **shall** function in the same way as in the IF Data Packets.

Rule 7.2-6: The Stream Identifier **shall** function in the same way as in the IF Context Packet.

Rule 7.2-7: The Context Packet Class documentation **shall** specify the format of the Extension Context Fields.

Recommendation 7.2-1: The Extension Context Packet Class **should** use the same packet header and timestamp definition as the IF Context Packet Class whenever possible to maximize interoperability and minimize additional development time.

8 Information Classes and Streams

This section sets forth rules related to Information Classes and Information Streams. An Information Class is a specification for the creation of an Information Stream. It describes the structure and purpose of each of the Packet Streams involved, as well as the association between the Packet Streams. It also specifies the purpose of the resulting Information Stream.

Definition 8-1: A VRT Information Class is the specification for a VRT Information Stream.

Rule 8-1: A VRT Information Class **shall** provide all of the defining details for the resulting Information Stream, as required by this specification.

The following sections provide rules and recommendations for the documentation of Information Classes for compliant Information Streams. Conformance to the rules is required. Conformance to the recommendations is not required, but will facilitate the quick assessment of the capabilities of an Information Stream by system designers. Section 8.1.1 presents rules and recommendations Information Classes. Section 8.1.2 presents rules for specifying the accuracy of various frequency and time fields, which is determined by the equipment emitting the Information Stream.

8.1 *Information Class Specification*

Definition 5-1: An Information Class is a specification for the creation of a VRT compliant Information Stream. This specification has seven components. They are:

1. Information Class Name
2. Information Stream Purpose
3. Packet Stream Names
4. Packet Stream Purposes
5. Packet Classes
6. Context Reference Points
7. Packet Stream Associations

The requirements for each of these components are given in the following subsections.

8.1.1 Information Class Name

The originating Information Class for a VRT compliant Information Stream must have a name. This name does not appear in the resulting Information Stream. It is simply provided as a way for system designers to refer to the structure of the resulting Information Stream(s). This name should be related to the purpose of the resulting Information Stream(s).

Rule 8.1.1-1: Every VRT Information Class **shall** have a name.

Recommendation 8.1.1-1: The name given to an Information Class **should** reflect the purpose of the class.

8.1.2 Information Stream Purpose

The Information Stream purpose is the reason for which the Information a Stream is to be created. It may be very general, such as, “To convey any 70 MHz IF Data,” or may be more specific, such as, “To convey FFT data from product X to product Y in system Z.” Information Streams with more constituent Packet Streams are likely to have more specific statements of purpose. The purpose statement is intended to facilitate both interoperability and Information Stream reuse.

Rule 8.1.2-1: A VRT Information Class **shall** specify the purpose of the resulting VRT Information Stream.

Recommendation 8.1.2-1: Whenever multiple Information Streams are to be created from one Information Class, The purpose of the Information Class **should** state this fact, and **should** be worded broadly enough to encompass the purpose of each resulting Information Stream.

Observation 8.1.2-1: A device that outputs N channels of data of type X might use the same Information Class N times, once for each channel. The purpose of the Information Class in such a case might be stated as "To convey data and context for one of N channels of data of type X.

8.1.3 Packet Stream Names

Packet Streams called out in an Information Class are given names to be used as identifiers. These names do not appear in the Information Stream, but are used by system designers to refer to the Packet Streams within an Information Stream. It is often convenient to choose a name that indicates the purpose of the Packet Stream. These names have meaning only within the context of an Information Stream. They are reused from one Information Stream to another when multiple Information Streams are created from the same Information Class.

Rule 8.1.3-1: An Information Class **shall** specify a name for every Packet Stream within the resulting Information Stream.

Recommendation 8.1.3-1: The name given to a Packet Stream **should** reflect its purpose.

8.1.4 Packet Stream Purposes

An Information Class must include a statement of the purpose of each Packet Stream it specifies for the Information Stream. The purpose of a VRT Packet Stream in an Information Stream is to convey some particular information that is part of that Information Stream. This purpose is more specific than the purpose stated for the corresponding Packet Class, which may support multiple Packet Streams, and which generally only specifies the type of information conveyed rather than the particular information conveyed. For example, if the purpose of a Packet Class is "to convey IF Data in real 16-bit samples," the purpose of a resulting Packet Stream might be, "to convey the IF Data in real 16-bit samples from tuner model 5 made by company XYZ." This purpose is stated as an aid to understanding the structure and uses of the resulting Information Streams.

Rule 8.1.4-1: An Information Class **shall** state the purpose of each Packet Stream included in the resulting Information Stream.

Observation 8.1.4-1: A device that outputs N channels of data of type X might output them as N Packet Streams within one Information Stream. The purpose of each of these Packet Streams in such a case might be stated as "To convey data of type X for channel i out of N channels.

8.1.5 Packet Classes

Most of the detailed information about for an Information Stream is provided by the Packet Classes that are included in the Information Class. These Packet Classes specify the structure and function of the Packet Streams that make up the Information Stream. For each type of Packet Stream in the Information Stream, a Packet Class is given for use in creating that Packet Stream. The same Packet Class may be used for the creation of multiple Packet Streams.

There are four categories of Packet Classes. They are:

- IF Data Packet Classes
- Extension Data Packet Classes
- IF Context Packet Classes
- Extension Context Packet Classes

Each category has its own rules for the allowed structure and function of packets in that category.

Rule 8.1.5-1: Every Packet Stream in a VRT Information Stream **shall** be generated from either an IF Data Packet Class, or an Extension Data Packet Class, or an IF Context Packet Class, or an Extension Context Packet Class.

Rule 8.1.5-2: A VRT Information Class **shall** include all the Packet Classes used to make Packet Streams within the resulting Information Stream.

Permission Error! Reference source not found.**-1:** The same Packet Class **may** be used for the creation of multiple Packet Streams within an Information Stream. It **may** also be used for the creation of Packet Streams in multiple Information Streams.

8.1.5.1 IF Data Packet Classes

An IF Data Packet Class consists of four things:

1. A class name.
2. The specification of the purpose of the class.
3. The specification of the choice of allowed options for the class.
4. The specification of the meaning of channel numbers and/or Event Tags included in the Item Packing Fields, and of the meaning of any Data Qualifiers used in the Trailer.

Rule 8.1.5.1-1: Every VRT IF Data Packet Class **shall** have a name.

Recommendation 8.1.5.1-1: The name given to an IF Data Packet Class **should** reflect the purpose of the class.

Rule 8.1.5.1-2: A VRT IF Data Packet Class **shall** specify its own purpose.

Rule 8.1.5.1-3: The specification of an IF Data Packet Class for a VRT compliant Information Stream **shall** include the specification of the option chosen for each item shown in Table 8.1.5.1-1.

Recommendation 8.4.1.1-2: The specification of an IF Data Packet Class **should** be formatted as a table with entries in the same order as in Table 8.1.5.1-1.

Rule 8.1.5.1-4: A VRT IF Data Packet Class **shall** specify the meaning of any Channel-tags included in the Item Packing Fields. IF a table such as Table 8.1.5.1-1 is used, this information may reside in the comment box next to the Channel-tag size.

Rule 8.1.5.1-5: A VRT IF Data Packet Class **shall** specify the meaning of any Event Tags included in the Item Packing Fields. IF a table such as Table 8.1.5.1-1 is used, this information may reside in the comment box next to the Event-tag size.

Rule 8.1.5.1-6: A VRT IF Data Packet Class **shall** specify the meaning of any Data Qualifiers included in the Trailer. IF a table such as Table 8.1.5.1-1 is used, this information may reside in the comment boxes in the Trailer section of the table.

IF Data Packet Class Options Table		
Class Name:		
Packet Header		
Parameter	**Allowed Options**	**Comments**
Packet Type	- IF Data Stream Packet with Context Link Key - IF Data Stream Packet w/o Context Link Key	
OUI	- Not used - Vendor OUI	
Class Code	- Not used - Vendor Packet Class Code	
Packet Size	Up to 1048575 32-bit words	
Stream Identifier	Any 32-bit value, when used	
Timestamp Field	- Sample Count Time-code - Real-time Time-code - Free-running Counter Time-code - Not used	
Free-Running Counter Modulus	- Modulus, 2 to 2^{64} - Not used	
Packet Payload		
Parameter	**Allowed Options**	**Comments**
Packing Method	- Link Efficient - Processing Efficient	
Item Packing Field Size	1 to 64 bits	
Data Item Size	1 to 64 bits	
Event-tag Size	0 to 7 bits	
Channel-tag Size	0 to 15 bits	
Vector Size	1 to 8 bits	
Data Item Format	- Unsigned fixed-point - 2's Complement Fixed Point - 32-bit IEEE 754 Floating-point - 64-bit IEEE 754 Floating-point - Unsigned VRT Floating-point (Exponent 1-6 bits) - Signed VRT Floating-point (Exponent 1-6 bits)	
Sample-repeating /Channel Repeating	- Sample Component repeating - Channel Repeating	
Real/Complex Type	- Real - Complex Cartesian - Complex Polar, Signed Phase - Complex Polar, Unsigned Phase	
Spectral Polarity	- Normal spectrum - Inverted Spectrum	
Repeat Count	1 to 1024	
Packet Trailer		
Parameter	**Allowed Options**	**Comments**
Calibrated Time Indicator	Used/Not used	
Valid Data Indicator	Used/Not used	
Reference Lock Indicator	Used/Not used	
AGC/MGC Indicator	Used/Not used	
Detected Signal Indicator	Used/Not used	
User-defined bit 4 to 0 (Manufacturer Shall specify function in comment field)	Each of the five bits may be: - Used (indicate function) - Not used	
Associated Context Packet Count	0 to 127	
Maximum Count	0 to 127	

Table 8.1.5.1-1: Options for an IF Data Packet Class.

8.1.5.2 IF Context Packet Classes

An IF Context Packet Class consists of four things:

1. A class name.
2. The specification of the purpose of the class.
3. The specification of the choice of allowed fields and options for the class.
4. The specification of the meaning of the enabled fields.

Rule 8.1.5.2-1: Every VRT IF Context Packet Class **shall** have a name.

Recommendation 8.1.5.2-1: The name given to an IF Context Packet Class **should** reflect the purpose of the class.

Rule 8.1.5.2-2: A VRT IF Context Packet Class **shall** specify its own purpose.

Rule 8.1.5.2-3: The specification of an IF Context Packet Class for a VRT compliant Information Stream **shall** include specification of the option chosen for each item shown in Table 8.1.5.2-1.

Recommendation 8.1.5.2-2: The specification of an IF Context Packet Class **should** be formatted as a table with entries in the same order as in Table 8.1.5.2-1.

Rule 8.1.5.2-4: A VRT IF Context Packet Class **shall** specify the meaning of any fields that are used. If a table such as Table 8.1.5.2-1 is used, this information may reside in the comment box next to the option to which it relates.

Recommendation 8.1.5.2-3: The IF Context Packet Class documentation **should** provide, in addition to the information required in Table 8.1.5.2-1, any comments required to clarify the use of fields. For example, when a field is dynamically enabled, the comment **should** say so, and **should** say under what conditions the field will be present. (This should probably be in the Information Stream Class.)

IF Context Packet Class Specification Table		
Class Name:		
Packet Header		
Parameter	Options	Comments
Packet Type	Standard Context Packet	
Timestamp Precision	- Sample-precision - Packet-precision	
Stream Identifier	- Used (Value documentation optional) - Not used	
Timestamp	- Sample Count Time-code - Real-time Time-code - Free-running Counter Time-code - Not Used	
Free-Running Counter Modulus	- Modulus - Not used	
Context Fields		
Parameter	Options	Comments
Reference Frequency	Used/Not used	
Reference Frequency Origin	Used/Not used	
Reference Frequency Origin Adjustment	Used/Not used	
Lower Band Edge	Used/Not used	
Upper Band Edge	Used/Not used	
Band Edge Interpretation	- The meaning the above two fields	
Sample Rate	Used/Not used	
Timestamp Adjustment	Used/Not used	
Reference Level	Used/Not used	
Process Gain	Used/Not used	
Overflow count	Used/Not used	
Temperature	Used/Not used	

Source Type	- Manufacturer source-code, when used. - Not used	
Logical Events		
Calibrated Time Indicator	Used/Not used	
Valid Data Indicator	Used/Not used	
Reference Lock Indicator	Used/Not used	
AGC/MGC Indicator	Used/Not used	
Detected Signal Indicator	Used/Not used	
User-Defined Bits [7..0]	For each of the eight bits: Used/Not used	
Data Packet Payload Format	Used/Not used	
Reference Point Link Key	Used/Not used	
GPS	Used/Not used	
Formatted GPS	Used/Not used	
Formatted INS	Used/Not used	
ECEF Ephemeris	Used/Not used	
Relative Ephemeris	Used/Not used	
GPS ASCII	Used/Not used	
Ephemeris Reference Link Key	Used/Not used	
Context Association Lists	Used/Not used	
Vector Component Stream IDs	Number of Keys in the list (0 to 511)	
Input Source Stream IDs	Number of Keys in the list (0 to 511)	
System Stream IDs	Number of Keys in the list (0 to 511)	

Table 8.1.5.2-1: Options for a Standard Context Packet Class.

8.1.5.3 Extension Data Packet Classes

Extension Data Packet Classes have more specification requirements than IF Data Packet Classes, due to the unrestricted payload contents and formats. There is no prescribed list of options for Extension Data packets. In fact, they may contain information of a type never considered as of this writing. Therefore, these packets require a complete specification of the meaning and format of the data items in the payload, and of the arrangement of these data items in the payload. An Extension Data Packet Class consists of five things:

1. The class name.
2. The specification of the purpose of the class.
3. The specification of the meaning of any of the user-defined bits that are utilized in the header.
4. The specification of the organization of the payload of the packet. This includes the location and format of each field in the payload.
5. The meaning of each field in the payload section of the packet.

Rule 8.1.5.3-1: Every VRT Extension Data Packet Class **shall** have a name.

Recommendation 8.1.5.3-1: The name given to an Extension Data Packet Class **should** reflect the purpose of the class.

Rule 8.1.5.3-2: An Extension Data Packet Class **shall** specify its own purpose.

Rule 8.1.5.3-3: The specification of an Extension Data Packet Class for a VRT compliant Information Stream **shall** include the specification meaning of any user-defined bits that are utilized in the header.

Rule 8.1.5.3-4: The specification of an Extension Data Packet Class for a VRT compliant Information Stream **shall** include the specification the organization of the payload section of the packet. This **shall** consist of the specification of the location and format of each field in the payload.

Permission 8.1.5.3-1: The specification of the organization of an Extension Data Packet Class **may** be given in any format that clearly conveys the organization.

Permission 8.1.5.3-2: The specification of the field formats in an Extension Data Packet Class **may** be given in any format that clearly conveys these formats. Different methods **may** be used for each field.

Rule 8.1.5.3-5: A VRT Extension Data Packet Class **shall** specify the meaning each field in the payload section of the packet.

Recommendation 8.1.5.3-2: An Extension Data Packet Class **should** be made as much like an IF Data Packet Class as possible, to facilitate the reuse of hardware and/or software.

Recommendation 8.1.5.3-3: As with other Packet Classes, the specification of Extension Packet Classes **should** include comments explaining subtleties such as when the fields are included, if they are dynamic. (This should probably move to the Information Class section,)

8.1.5.4 Extension Context Packet Classes

Extension Context Packet Classes are more complicated to specify than IF Context Packet Classes, due to the unrestricted payload contents and formats. There is no prescribed list of options for Extension Context packets. In fact, they may contain information of a type never considered as of this writing. Therefore, these packets require a complete specification of the meaning and format of the data items in the payload, and of the arrangement of these data items in the payload. An Extension Context Packet Class consists of five things:

1. The class name.
2. The specification of the purpose of the class.
3. The specification of the meaning of any of the user-defined bits that are utilized in the header.
4. The specification of the organization of the payload of the packet. This includes the location and format of each field in the payload.
5. The meaning of each field in the payload section of the packet.

Rule 8.1.5.4-1: Every VRT Extension Context Packet Class **shall** have a name.

Recommendation 8.1.5.4-1: The name given to an Extension Context Packet Class **should** reflect the purpose of the class.

Rule 8.1.5.4-2: An Extension Context Packet Class **shall** specify its own purpose.

Rule 8.1.5.4-3: The specification of an Extension Context Packet Class for a VRT compliant Information Stream **shall** include the specification meaning of any user-defined bits that are utilized in the header.

Rule 8.1.5.4-4: The specification of an Extension Context Packet Class for a VRT Information Stream **shall** include the specification of the organization of the payload section of the packet. This **shall** consist of the specification of the location and format of each field in the payload.

Permission 8.1.5.4-1: The specification of the organization of an Extension Context Packet Class **may** be given in any format that clearly conveys the organization.

Permission 8.1.5.4-2: The specification of the field formats in an Extension Context Packet Class **may** be given in any format that clearly conveys these formats. Different methods **may** be used for each field.

Rule 8.1.5.4-5: A VRT Extension Context Packet Class **shall** specify the meaning each field in the payload section of the packet.

Recommendation 8.1.5.4-2: An Extension Context Packet Class **should** be made as much like an IF Context Packet Class as possible, to facilitate the reuse of hardware and/or software.

Recommendation 8.1.5.4-3: As with other Packet Classes, the specification of Extension Packet Classes **should** include comments explaining subtleties such as when the fields are included, if they are dynamic. (This should probably move to the Information Class section,)

8.1.6 Context Reference Points

There are four fields that might be included in an IF Context packet that convey information usually related to a signal at some upstream point in the system. They are:

- RF Reference Frequency
- RF Reference Frequency Offset
- Reference Level
- Timestamp Adjustment

In order for these fields to be properly understood, the exact point in the system to which they refer must be specified. It is also possible that Extension Context Packets might also include fields that convey similar information related to some upstream point in the system. In that case it would also be true that the exact point in the system to which they refer must be specified. Such a point is called a Context Reference Point.

Rule 8.1.6-1: For any Context Packet Stream conveying information related to one or more Context Reference Points, that location of those Context Reference Points in the system **shall** be specified in the Information Class.

8.1.7 Packet Stream Associations

Every VRT compliant Information Class must provide a specification of the Packet Stream associations in the resulting Information Stream. This association reflects the relative purpose of each Packet Stream. For example, an IF Context Packet Stream is related to the IF Data Packet Stream for which it provides Context. Therefore they should be associated. The association specification may take the form of a diagram, or of an outline-like text description. Appendix A presents an example of each of these methods. Also, any other method may be used, as long as it is complete and unambiguous.

In addition to being documented in the Information Class, Packet Stream association is communicated in the Information Stream by the Stream Association fields in the IF Context Packets. This mechanism for conveying the associations between Packet Streams is described in Section 6.

Rule 8.1.7-1: The Information Class for a VRT Information Stream **shall** specify the associations between the Packet Streams that make up that Information Stream.

Rule 8.1.7-2: Every Packet Stream specified by an Information Class as part of a VRT Information Stream, **shall** be associated with at least one other Packet Stream in that Information Stream.

Permission 8.1.7-1: A Packet Stream specified by an Information Class as part of a VRT Information Stream **may** be associated to more than one other Packet Stream within the Information Stream. (We need to discuss this. It relates to how many data streams can go in an Information Stream. -Dick)

Permission 8.1.7-2: A Packet Stream specified by an Information Class as part of a VRT Information Stream **may** be associated with Packet Streams in more than one Information Stream. This situation constitutes a shared Packet Stream.

Permission 8.1.7-3: An arbitrary number of Packet Streams of each type **may** be associated together into one Information Stream, as long as the association rules of Section 6 are not violated.

Permission 8.1.7-4: The specification of Packet Stream associations for an Information Stream **may** take any form that is complete and unambiguous.

Recommendation 8.1.7-1: The association between the Packet Streams, specified by the Information Class, **should** be reflected by the presence of the Stream Identifiers in the Stream Association Lists (see Section 6) whenever those lists are present in the packets making up the Packet Streams.

8.2 *Other Required Information Class Specifications*

Table 8.2-1 contains remaining specifications required for an Information Class. These specifications are related to the accuracy of certain fields conveyed in IF Context packets. This information need not be provided if the corresponding field is not used.

Rule 8.2-1: For each IF Context Packet Stream in a VRT Information Stream, the Information Class **shall** specify each item shown in Table 8.2-1 if the corresponding field is used.

Permission 8.2-1: In the event that a specification is unknown, or depends on some external device, or for any other reason cannot be given, the Information Class **may** provide as a specification any response that accurately communicates that fact.

Permission 8.2-2: When the specifications listed in Table 8.2-1 are common to multiple IF Context Packet Streams in an Information Stream, the originating Information Class **may** simple provide one table of specifications and a list of Packet Streams to which it applies.

Parameter	Explanation
Time Stamp Accuracy	RMS, or maximum peak-to-peak timestamp error, typically in nanoseconds.
Timestamp Adjustment Accuracy	
IF Reference Frequency Accuracy	RMS, or maximum peak-to-peak frequency error, in Hertz of the values in these fields.
Reference Frequency Origin Accuracy	
Reference Frequency Origin Adjustment Accuracy	
Bandwidth Edge Accuracy	
Sample Rate Accuracy	
Gain Accuracy	RMS, or maximum peak-to-peak gain error, in dB of the value in the “Gain” field.
Reference Level Accuracy	RMS, or maximum peak-to-peak gain error, in dB of the value in the “Reference Level” field.
Temperature Accuracy	RMS, or maximum peak-to-peak gain error, in degrees Celsius of the value in the “Temperature” field.

Table 8.2-1: Accuracy Specifications required for a full VRT Specification.
Use of the fields referenced is optional for the IF Context Packet Classes.

Appendix A Information Stream Specification - Example

Figure A-1 shows an example system using VRT transport. It consists of a tuner with a VRT digital output, followed by a 1024 QAM demodulator. There is also a system controller and an Ethernet interface that formats the data for output on an Ethernet link.

In this example we will focus on the VRT/Ethernet link emerging from the Ethernet interface. However, we will first make a few observations about the other VRT Information Streams.

The tuner outputs two copies of a very basic VRT Information stream consisting of only an IF Data Packet Stream. This Packet Stream conveys IF Data to the demodulator over one link, and simultaneously conveys the same IF Data to the Ethernet interface over a second link. In this case, the tuner-related Context is gathered by the system controller via the control bus. The system controller also keeps track of the power supply voltages of the modules and their temperatures, and GPS data. It formats the tuner Context and system health information into VRT packets and outputs them to the Ethernet interface.

The Demodulator receives the IF Data and demodulates a 1024 QAM signal contained within it. It outputs the demodulated bit stream in a new Information Stream consisting of only an Extension Data Packet Stream tailored to convey the demodulated bits. Context such as demodulator carrier-lock (or unlock) is gathered by the system controller via the control bus, formatted into VRT packets, and sent to the Ethernet Interface.

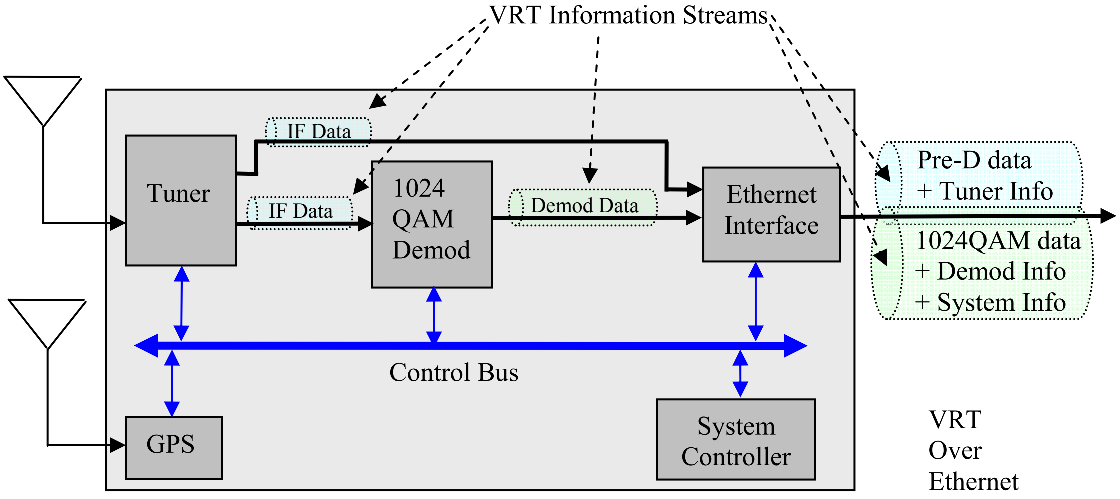


Figure A-1: An example system. The Tuner outputs an IF Data to the Demodulator and to the Ethernet Interface. The Demodulator outputs a demodulated bit stream (in Extension Data Packets) to the Ethernet interface. The Ethernet interface forwards both packet streams over the Ethernet link, along with board temperatures, GPS information, and other system health information.

The Ethernet interface receives both the IF Data Information Stream from the tuner, and the demodulated Information Stream from the demodulator. The Ethernet interface creates two Information Streams, one for the tuner data and one for the demodulated data. Each new Information Stream contains associated Context Packet Streams, provided by the system controller.

Figure A-2 graphically depicts the association between the Packet Classes in these two Information Streams. The Packet Stream names and types are indicated in the figure. Also, the type of association between each pair of Packet Streams is indicated by an arrow with a label next to it. A bidirectional arrow corresponds to a direct association, which corresponds to the Data Packet Stream and Context Packet Stream sharing a Stream ID. There are two of these arrows, one in each Information Stream. In each case it associates directly with the data the Context related to the module that produced the data.

In the figure, the "System Info" Packet Stream is associated with the "Demod Info" Packet Stream via a "System Association. This indirectly associates it with the "1024QAM Data" Packet Stream. The "System Association" arrow corresponds to the presence of the "System Info" Stream ID in the System Association List in the "Demod Info" Packet Stream. (See Section 6 for Association List details.)

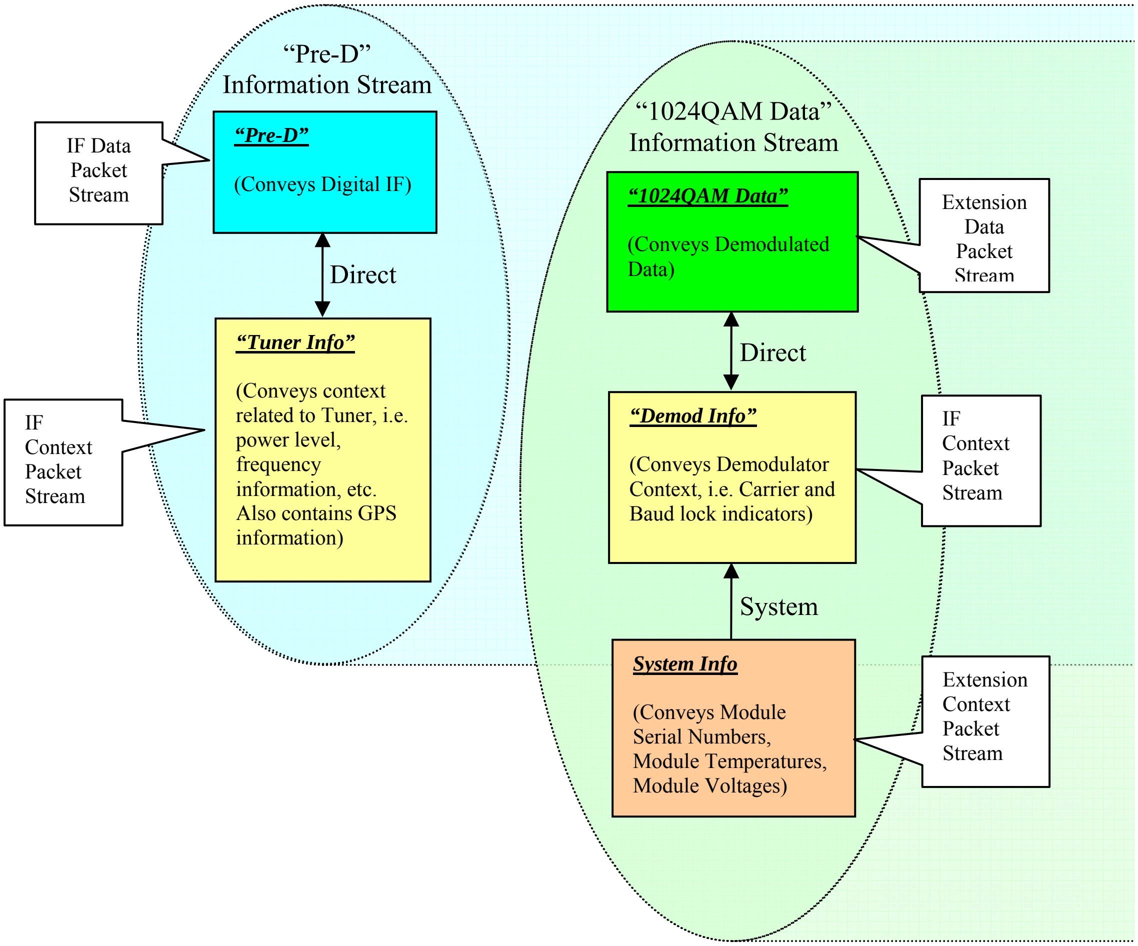


Figure A-2: Packet Stream associations in the example Information Stream. The "Pre-D" Information Stream contains the "Pre-D" IF Data Packet Stream, and the directly associated "Tuner Info" IF Context Packet Stream. The "1024QAM Data" Information Stream contains the "1024QAM Data" Extension Data Packet Stream, the directly associated "Demod Info" IF Context Packet Stream and the indirectly associated "System Info" Extension Context Packet Stream. The label next to each arrow indicates the type of association between the Packet Streams.

When the number of Packet Streams in an Information Stream is large, a graphical representation may not work well, since it may not fit on one page, or be easily segmented onto multiple pages. In such cases, an outline representation, such as the one below is an alternate way to specify Packet Class links for the Information Stream in the current example.

1. Information Stream: “***Pre-D***”
    1.1 IF Data Packet Stream: “***Pre-D***”
    1.2Directly Associated IF Context Packet Stream: “***Tuner Info***”

2. Information Stream: “1024QAM Data”
    2.1 Derived Data Packet Stream: “***1024QAMd Data***”
    2.2 Directly Associated IF Context Packet Stream: “***Demod Info***”
        2.2.2 System Stream Association List:
            2.2.2.1 Extension Context Packet Stream***: “System Info”***

The outline format is more compact than a graphical representation. Unfortunately in some cases the outline form can become confusing. For example, if some Context is shared between multiple data streams, then that Context Packet Stream would appear multiple times in the outline. In such cases, a simple list of associations might be provided, as shown below for this example.

Information Stream: “***Pre-D***”
1. (“***Tuner Info***” IF Context) ← (direct) → (“***Pre-D***” IF Data)

Information Stream: “1024QAM Data”
1. (“***Demod Info***” IF Context) ← (direct) → (“***1024QAMd Data***” Extension Data)
2. (***“System Info”*** Extension Context) → (system) → (“***Demod Info***” IF Context)

This method works for any set of associations, but does not lend itself to being easily pictured by the reader. As with the other methods, it has its weaknesses. Any method that completely and unambiguously indicates all the Packet Stream associations may be used. The most appropriate method is the one that conveys the associations most clearly for that application.

A.1 Specifying Packet Classes - Example

The current example uses Packet Classes from four categories. It uses one IF Data Packet Class, one Extension Data Packet Class, two IF Context Packet Classes, and one Extension Context Packet Class. The following sections show how to document each Packet Class.

A.1.1 *IF Data Packet Class Specification - Example*

IF Data Packet Classes are easily documented based on the list of options for an IF Data Packet Class. Table A.2.1-1 lists the selected options, and is filled in appropriately for the “Pre-D” IF Data Packet Class.

<table>
<tr><th colspan="3">IF Data Packet Class</th></tr>
<tr><th colspan="3">Class Name: “Pre-D”</th></tr>
<tr><th colspan="3">Packet Header</th></tr>
<tr><th>Parameter</th><th>Option</th><th>Comments</th></tr>
<tr><td>Packet Type</td><td>IF Data Packet with Stream ID</td><td>IF Data Packets</td></tr>
<tr><td>OUI</td><td>Not Present</td><td></td></tr>
<tr><td>Class Code</td><td>Not Present</td><td>Class Code is “19” but is not sent in packets.</td></tr>
<tr><td>Packet Size</td><td>2054 Words</td><td>Fixed Packet Size.</td></tr>
<tr><td>Context Link Key</td><td>Yes</td><td>Actual code is user selectable</td></tr>
<tr><td>Timestamp Field</td><td>Real Time</td><td>Present in every Packet</td></tr>
<tr><td>Free Running Counter Modulus</td><td>N/A</td><td></td></tr>
</table>

Packet Payload		
Parameter	**Option**	**Comments**
Packing Method	Link Efficient	2048 32-bit payload words containing 4681 14-bit words. Two fill bits at the end of the payload.
Item Packing Field Size	14	
Data Item Size	14	
Event-tag Size	0	No Event Tags
Channel-tag Size	0	No Channel Tags
Vector Size	0	No Vectors
Data Item Format	2’s Complement	
Sample-repeating/Channel Repeating	N/A	No repeating of any kind
Real/Complex Type	Real Data	
Spectral Polarity	Non-inverted	Same as Tuner input
Repeat Count	0	No repeating of any kind
Packet Trailer		
Parameter	**Option**	**Comments**
Calibrated Time Indicator	No	These details are sent in the trailer of the “Tuner Info” context packets, as indicated in those specifications.
Valid Data Indicator	No	
Reference Lock Indicator	No	
AGC/MGC Indicator	No	
Detected Signal Indicator	N/A	
User-defined bit 4 to 0 (Manufacturer Shall specify function in comment field)	No	Not used and undefined
Associated Context Packet Count	Yes	If too many context changes occur during one Data Packet interval, they will be combined, resulting in a loss of timing accuracy for some context change times.
Maximum Count	16	

Table A.2.1-1: Specification of the “Pre-D” Signal Data Packet Class.
Comments help clarify the application, and may be as long as required. The table specifies various reasonable options for this example.

A.1.2 *Extension Data Packet Class Specification - Example*

The format of Extension Data Packet Class in this case is nearly identical to that of an IF Data Packet Class. Therefore the same options table will be used to specify this Packet Class. The only difference between this and the IF Data Packet Class is that the demodulated bit stream does not conform to any of the numbering systems specifiable in the IF Data Packet Class, so the Data Item format is identified as “other.” The Table A.2.2-1 specifies the format of the “1024QAM Data” Packet Class.

Extension Data Packet Class		
Class Name: “1024QAM Data”		
Packet Header		
Parameter	**Option**	**Comments**
Packet Type	Extension Data Packet with Stream ID	Same format as IF Data Packets, but with custom Data Items.
OUI	0xE3199A	
Class Code	23	This Class was created for this application.
Packet Size	326 Words	Fixed Packet Size
Context Link Key	Yes	Actual code is user selectable
Timestamp Field	Real Time	Present in every Packet
Free Running Counter Modulus	N/A	

Packet Payload		
Parameter	**Option**	**Comments**
Packing Method	Link Efficient	320 32-bit words containing 1024 10-bit symbols from the 1024 QAM constellation.
Item Packing Field Size	10	
Data Item Size	10	
Event-tag Size	0	No Event Tags
Channel-tag Size	0	No Channel Tags
Vector Size	0	No Vectors
Data Item Format	"Other"	10-bit symbols from the 1024 QAM constellation.
Sample-repeating/Channel Repeating	N/A	No repeating of any kind
Real/Complex Type	N/A	10-bit symbols
Spectral Polarity	N/A	Doesn't apply to this demodulated data
Repeat Count	0	N/A
Packet Trailer		
Parameter	**Option**	**Comments**
Calibrated Time Indicator	No	This information is sent in the "Demod Info" context packets.
Valid Data Indicator	No	
Reference Lock Indicator	No	
AGC/MGC Indicator	No	
Detected Signal Indicator	No	
User-defined bit 4-0 (Manufacturer Shall specify function in comment field)		Unused and undefined
Associated Context Packet Count	Yes	If too many context changes occur during one Data Packet interval, they will be combined, resulting in a loss of timing accuracy for some context change times.
Maximum Count	16	

Table A.2.2-1: Specification of the "1024QAM Data" Extension Data Packet Class. This particular Extension Data Packet Class is made just like an IF Data Packet Class, except that the Data Items do not contain any of the predefined binary formats. The table specifies various reasonable options and comments for this example.

A.1.3 *IF Context Packet Class Specification - Examples*

In this example there are two Packet Classes falling into the category of an "IF Context Packet Class. The first is the "Demod Info" Packet Class, and the second is the "Tuner Info" Packet Class. The "Demod Info" IF Context Packet Class contains the Stream ID for another Context Packet Class in the "1024QAM Data" Information Stream. It also contains several context updates related to the Demodulator. Table A.2.3-1 specifies the "Demod Info" Packet Class.

IF Context Packet Class		
Class Name: "Demod Info"		
Packet Header		
Option	Supported/Parameter	Comments
Packet Type	IF Context Packet	
Rolling Counter	Yes	
TSE	"1"	Context changes are indicated with near sample accuracy.
Context Link Key	Yes	Actual code is user selectable
Timestamp	Real Time	Present in every packet
Context Fields		
Option	Supported/Parameter	Comments

IF Reference Frequency	No	
IF Reference Frequency Offset	No	
RF Reference Frequency	No	
RF Reference Frequency Offset	No	
Bandwidth	No	
Band Edge Interpretation	N/A	
Sample Rate	Yes	Always indicates the 10-bit symbol rate of 102.4 KHz
Timestamp Adjustment	No	
Reference Level	No	
Process Gain	No	
Overflow count	No	
Temperature	No	Temperatures are in Custom Packet
Source Type	Yes	Vendor OUI: xxyyzz. Vendor Device code: 0x19, which specifies the demodulator.
Logical Events	Yes	
Calibrated Time Indicator	Yes	Echoes “Pre-D” calibration indicator
Valid Data Indicator	Yes	Indicates valid Pre-D data AND carrier AND baud lock
Reference Lock Indicator	No	Not used
AGC/MGC Indicator	Yes	AGC mode always
Detected Signal Indicator	Yes	Indicates acceptable bit error rate from demodulator
User-Defined Bits [7..2]	No	Unused and undefined.
User-Defined Bits 1	Yes	Carrier Lock
User-Defined Bits 0	Yes	Baud Lock
Signal Data Packet Payload Format	Yes	This field is inserted about once per second.
Reference Point Key	No	
GPS	No	
Stream Association Lists	Yes	
Vector Component Keys	0	
Input Source Keys	0	
System Keys	Yes, 1	Links in “System Maintenance” Context

Figure A.2.3-1: Specification of the “Demod Info” IF Context Packet Class. This class is primarily for updating context related to the demodulator. It also links in the “System Info” Context Packet Class.

Table A.2.3-2 specifies the “Tuner Info” Packet Class. This class conveys information about tuner settings and state that are required for a complete explanation of the IF Data Stream coming from the tuner. It also contains GPS information.

IF Context Packet Class		
Class Name: **“Tuner Info”**		
Packet Header		
Option	Supported/Parameter	Comments
Packet Type	IF Context Packet	
Rolling Counter	Yes	
TSE	“1”	Context changes are indicated with near sample accuracy.
Context Link Key	Yes	Actual code is user selectable
Timestamp	Real Time	Present in every packet
Context Fields		
Option	Supported/Parameter	Comments
IF Reference Frequency	Yes	Always indicates the center of the digitized band, i.e. 17.5 MHz
IF Reference Frequency Offset	No	There is no offset
RF Reference Frequency	Yes	
RF Reference Frequency Offset	No	

Bandwidth	Yes	Always 30 MHz
Band Edge Interpretation	-3 dB point	
Sample Rate	Yes	Always 70 MHz IF Data sample rate
Timestamp Adjustment	No	
Reference Level	Yes	At the tuner input
Process Gain	No	
Overflow count	No	
Temperature	No	Temperatures are in a Custom Packet
Source Type	Yes	Vendor OUI: xxyyzz. Vendor Device code: 0x19, which specifies the entire system in Figure A-1.
Logical Events	Yes	
Calibrated Time Indicator	Yes	
Valid Data Indicator	Yes	“1” indicates tuner lock AND no overflow in the current packet.
Reference Lock Indicator	Yes	
AGC/MGC Indicator	Yes	AGC mode always
Detected Signal Indicator	No	N/A
User-Defined Bits [7..0]	No	
Signal Data Packet Payload Format	Yes	Indicates the format of the “Pre-D” packets.
Reference Point Link Key	No	
GPS	Yes	
Formatted GPS Geolocation	Yes	Updated once per second.
Formatted INS Geolocation	No	
ECEF Ephemeris	No	
Relative Ephemeris	No	
GPS ASCII	No	
Ephemeris Reference Link Key	No	
Stream Association Lists	No	

Figure A.2.3-2: Specification of the “Tuner Info” IF Context Packet Class. This class reports some static tuner context, and some changes in tuner context. It also reports GPS information about once per second.

A.1.4 *Extension Context Packet Class Specification - Example*

The “System Info” Extension Context Packet Class Is an entirely custom Packet Class. The fields in contains are not like those in the Standard Context Packet Class, so an entirely new packet format is required. Figure A.2.4-1 shows the format for this example Packet Class.

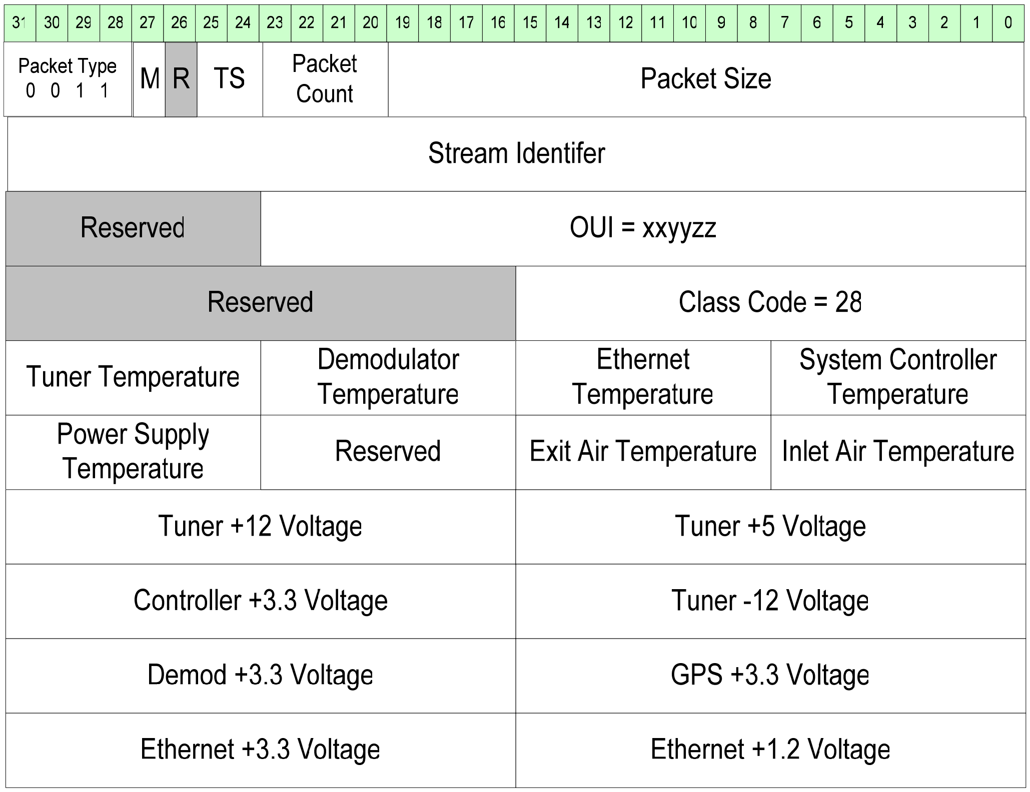


Figure A.2.4-1: Format of the Extension “System Info” Context Packet Class. The OUI corresponds to the equipment manufacturer. Together, the OUI and Application ID indicate this packet format.

Figure A.2.4-1 shows the format of the packet, but fails to specify the interpretation of the custom fields. This may be accomplished by simply adding a few notes to the specification, such as:

1. All temperatures are 2’s complement numbers ranging from -128 Celsius to +127 Celsius.
2. All voltages are unsigned numbers with ten bits to the right of the radix point and six to the left. This gives a range of zero to approximately 63.999 volts and about one millivolt resolution.

A.1.5 *Other Required Information Stream Specifications*

Table A.2.5-1 contains remaining VRT features/options to complete the specification of this example Information Stream.

Parameter	Specification
Time Stamp Accuracy	+/-100 ns
IF Reference Frequency Accuracy	GPS derived
RF Reference Frequency Accuracy	
IF Reference Frequency Offset Accuracy	N/A
RF Reference Frequency Offset Accuracy	N/A
Bandwidth Accuracy	+/-200 KHz
Gain Accuracy	N/A
Reference Level Accuracy	+/- 5%
Sample Rate Accuracy	GPS derived
Timestamp Adjustment Accuracy	NA
Temperature Accuracy	+/- 2 degrees Celsius

Table A.2.5-1: Other Required Information Stream Specifications.
An example specification is included for each item.

Appendix B Context Field Examples

B.1 Reference Point Identifier Example

Figure 8.1.5.4-1 shows an example of the usage of the Reference Point ID. In this example there are three processes, a Digital Receiver, a DDC, and a Demodulator, each of which outputs a Data Packet Stream and a Context Packet Stream. The Context Packet Streams coming from the Tuner, DDC, and Demodulator, have Stream Identifiers S1, S2, and S3 respectively, each forming a data association with their respective Data Packet Streams.

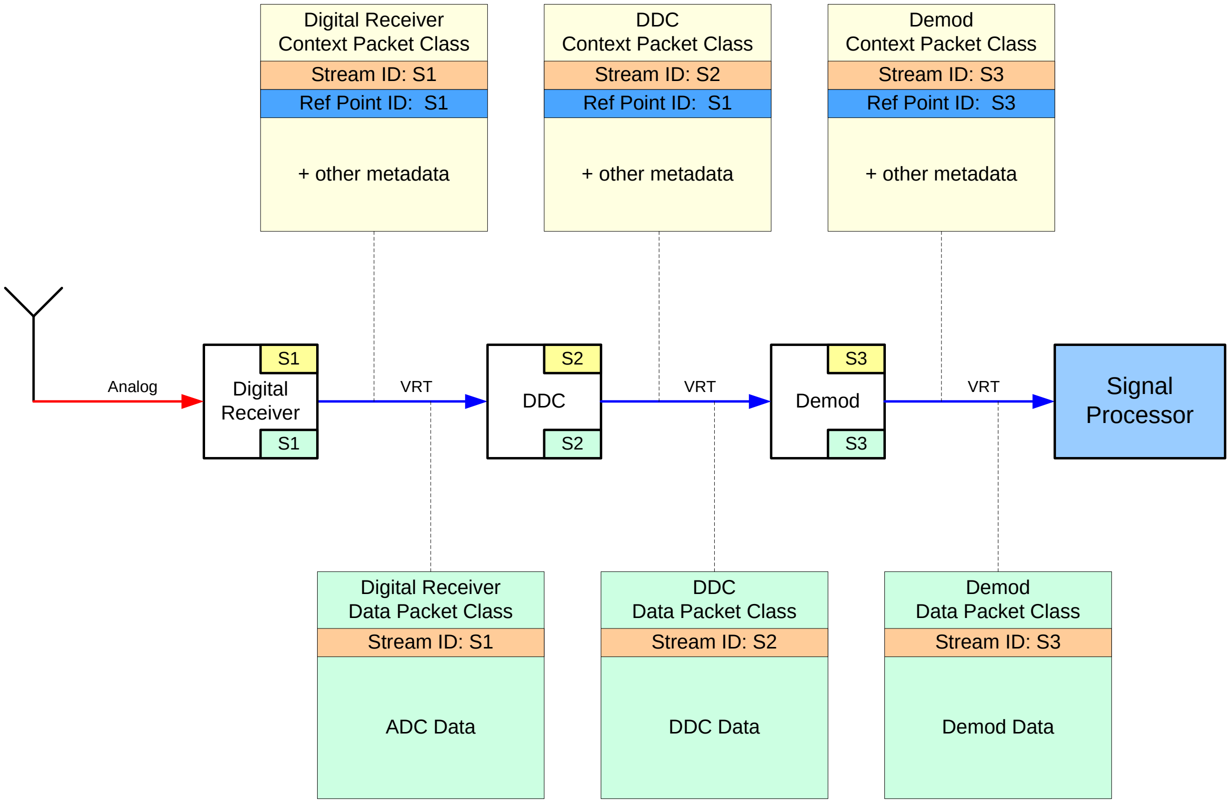
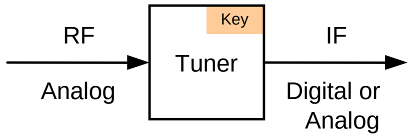


Figure 8.1.5.4-1: Reference Point ID Example
The reference point for the Digital Receiver and DDC Data is the Receiver input.
The reference point for the Demodulator is the Demodulator input.

Each of the Context Packet Streams contains a Reference Point ID field that specifies the reference point for each process. In this example we can see that the Receiver and DDC Reference Point IDs are both S1, which specifies the input to the Receiver as their reference point. The Demodulator, however, has S3 as its Reference Point ID, which specifies its own input as its reference point.

B.2 Spectral Fields Example

Figure 8.1.5.4-2 shows the meaning of the IF Reference Frequency, RF Reference Frequency, and Bandwidth fields.



(a): Basic VRT system

(b): Original and Translated Spectra.

Figure 8.1.5.4-2: Spectral Fields Usage Example

The tuner, shown in (a) results in the frequency translation shown in (b). The Reference point is the input to the tuner, and the associated data stream (or possibly associated analog signal) is the tuner's output. RF Reference Frequency refers to the Reference point signal, while IF Reference Frequency refers to the digital or analog IF signal. Bandwidth has the same meaning at both points.

Figure 8.1.5.4-2 (a) shows a block diagram containing only one process, a Tuner. The tuner translates spectral energy from a high frequency RF band to a lower frequency IF band. Figure 8.1.5.4-2 (b) shows the RF spectrum at the input of the tuner and the IF spectrum at the output of the tuner. In this example, the Reference Point is at the input to the tuner. Note that although VRT is intended primarily for IF Data signals, VRT context streams may be used to provide context for analog signals. In general, the choice of spectral field values will be impacted by whether the associated data stream is analog or digital. For example, with digitized IF signal the IF Reference Frequency might be 1/4th the sample rate. For analog signals however, sample rate is meaningless, so the IF filter center-point would be a more likely choice.

The Context Packet describing the above signal, be it a digital or analog signal, would use three of the spectral context fields, i.e. the Bandwidth field, the IF Reference Frequency field, and the RF Reference Frequency field. In this example we assume that this bandwidth is determined by the 3dB bandwidth at the tuner output. The Bandwidth field of the Context Packet would specify this value. The IF center frequency in this example corresponds to the nominal center of the output bandwidth the tuner. This frequency is marked with a wide bar in the Figure. The IF Reference Frequency field in the Context Packet would specify this value. The tuner input frequency that translates to the IF Reference Frequency would be specified by the RF Reference Frequency field. Assuming the spectrum is not inverted from the RF to the IF band, the Spectral Inversion bit ("I") in the RF Reference Frequency field would be zero. Finally, we note that since the output band in this example is symmetric about the IF Reference Frequency, the Bandwidth Frequency Offset field is not necessary.

If we assume in this example that the tuner is tuned to 2 GHz, and downconverts a 30 MHz band to a 70 MHz IF, without inverting its spectrum, then the spectral fields would contain the following values:

Bandwidth: 30 MHz
IF Reference Frequency: 70 MHz
RF Reference Frequency: 2 GHz
Spectral Inversion: 0 (not inverted)

See Appendix B.2 for more examples of higher-complexity use-cases for the spectral fields.

The translation frequency between the reference point and process output can be calculated from the RF and IF Reference Frequency fields. This translation frequency can be used to determine the corresponding location at the reference point of any point in the output spectrum. The translation frequency is also used in the calculations required when combining several contiguous processes or moving the reference point to a new location. Examples using the translation frequency are given in Appendix B.2. I think this paragraph should be in the appendix with the examples. -Dick

B.3 Reference Frequency Offset

In the case of a channelizer following a tuner, a Context Packet directly associated to a channelizer output data stream would hold the RF Reference Frequency Offset, and the Context Packet directly associated to the tuner output data would hold the RF Reference Frequency. The tuner Context Packet also would be associated to all the channelizer Context Packets using the Input Source Stream Association Lists (Described in Section 6.2.5.24). Whenever the tuner changed frequency, a single tuner Context Packet would be sent to convey the new RF Reference Frequency. Receivers of this Context Packet would add the RF Reference Frequency to the RF Reference Frequency Offset to get the new effective RF Frequency for the channelizer output channels.

Might need a picture to go with the above text. - Dick

B.4 *Bandwidth Offset*

Figure 8.1.5.4-1 shows an example similar to that described in the
. In this case the Tuner output filter is not precisely centered about the IF Reference Frequency. Therefore the Bandwidth Frequency Offset field is used to specify the difference between the true center of the band and the IF Reference Frequency.

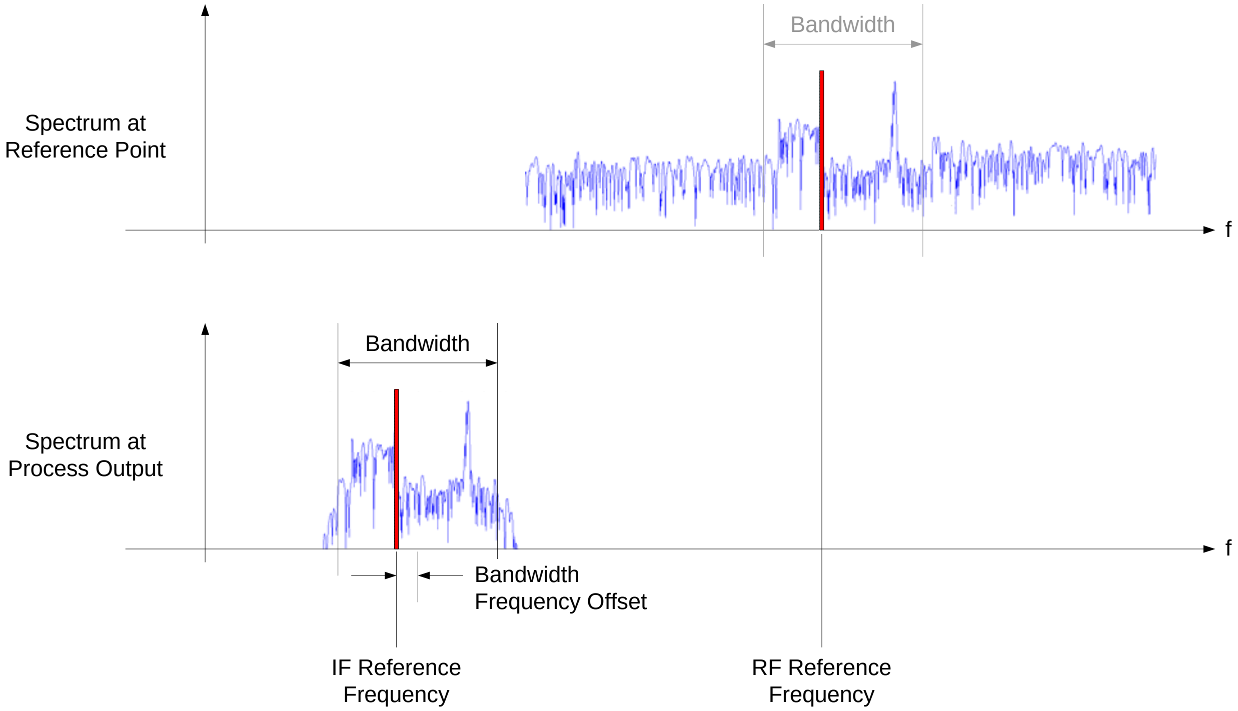


Figure 8.1.5.4-1: Bandwidth Frequency Offset Example

B.5 *Input Source Stream Association List*

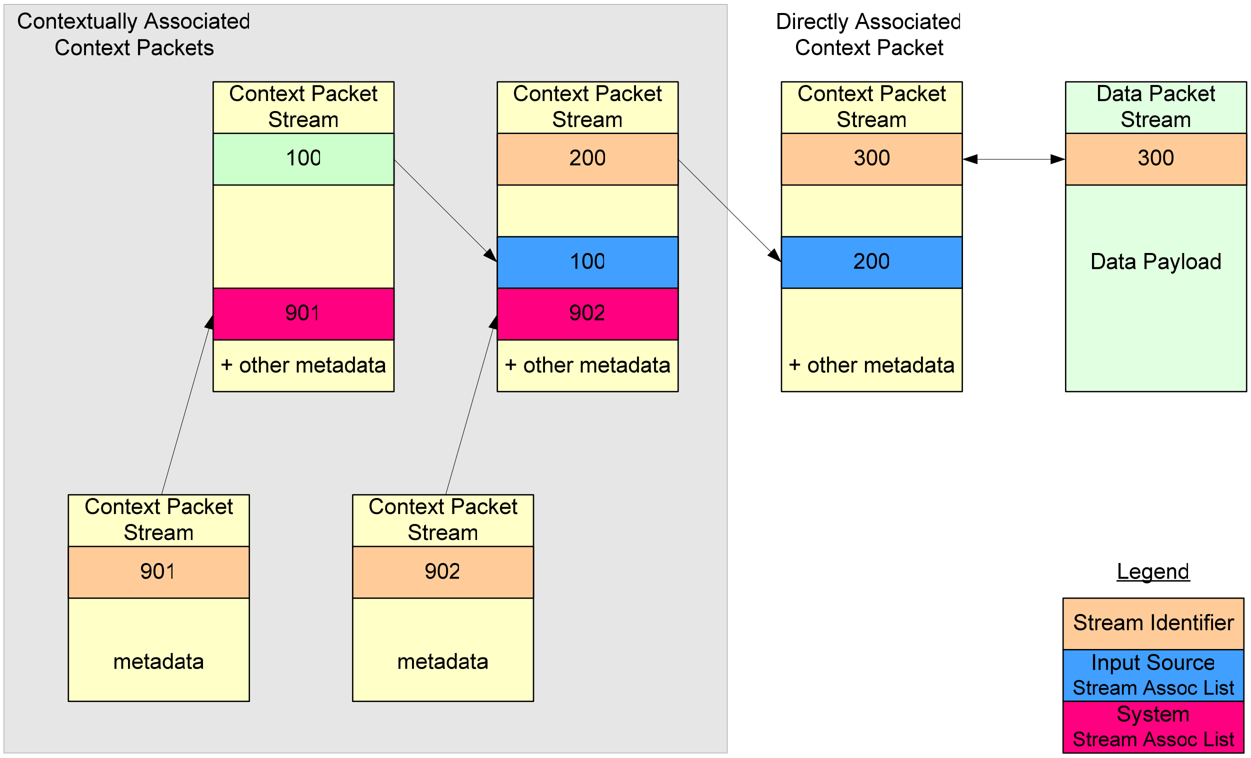


Figure 8.1.5.4-3: Direct and Contextual Associations of Context Packet Streams to a Data Packet Stream. The Data Packet Stream and its directly associated Context Packet Stream have a Stream ID of 300. The contextually associated Packet Streams have Stream IDs of 100, 200, 901, and 902.

Figure 8.1.5.4-1 illustrates the use of Input Source Stream IDs to associate the required Context Packet Streams in the case of the tuner, DDC and demodulator.

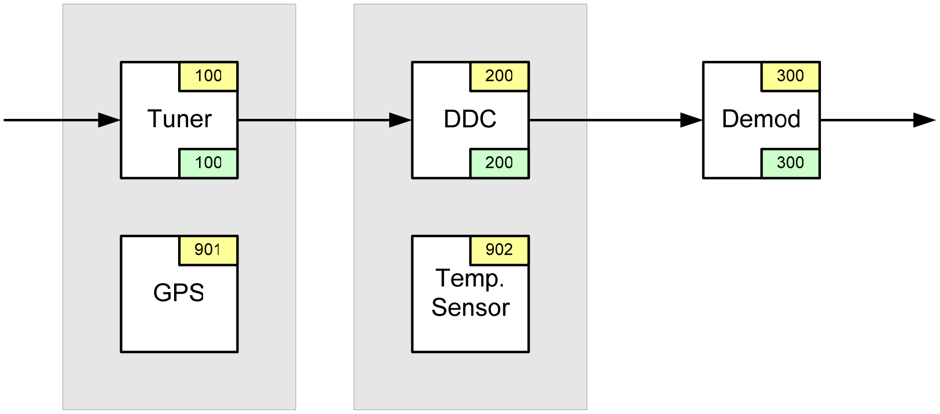


Figure 8.1.5.4-1 (a): System Block Diagram

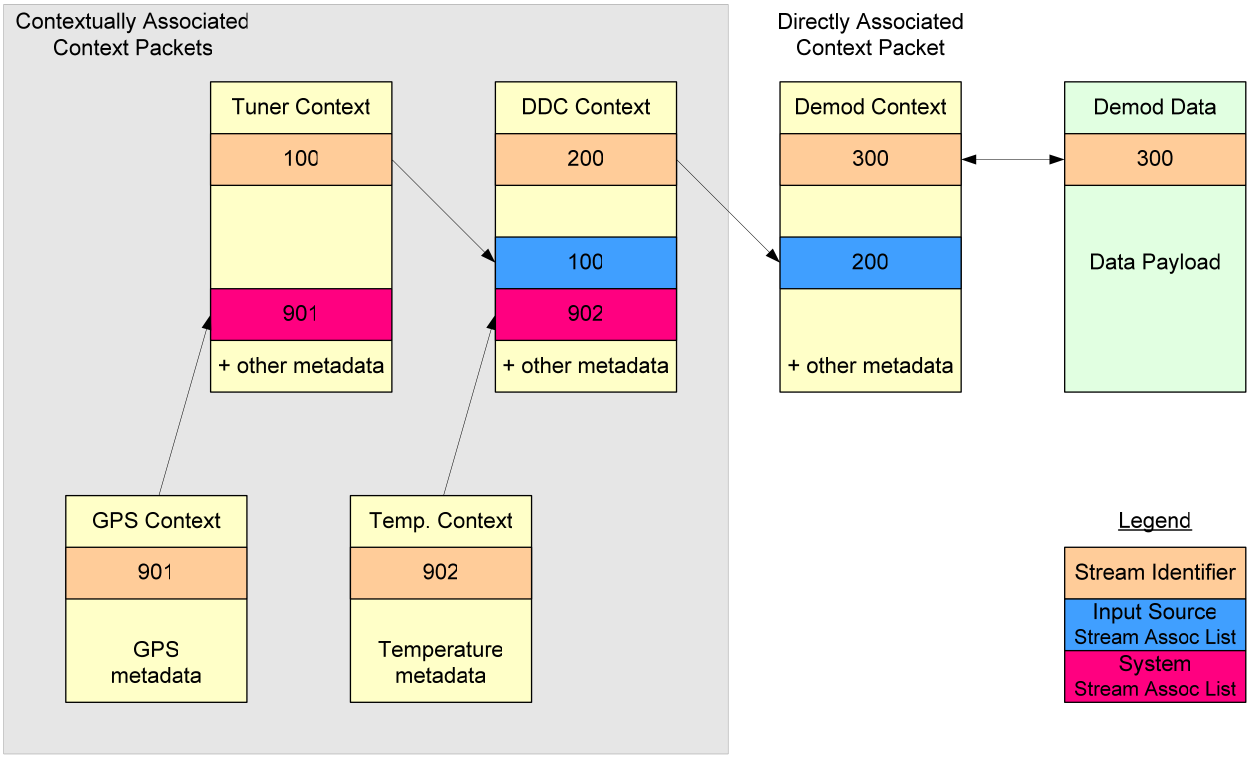


Figure 8.1.5.4-1 (b): Stream Associations

Figure 8.1.5.4-1: An example use of the Input Source Stream Association List

The Packet Streams for the system of Figure (a) are associated according to Figure (b). The Context Packet Stream directly associated to the demodulated data stream contains metadata related to the demodulator. Its Input Source Stream Association List contains a Stream ID (200) associating a Context Packet Stream for DDC metadata. This Packet Stream contains in its Input Source Stream Association List a Stream ID (100) associating in a Context Packet Stream for tuner metadata.

Appendix C Glossary

1 PPS

A 1 PPS (**One** **P**ulse **P**er **S**econd) signal is a pulse-train having one pulse per second. The pulse edge marks the time of the UTC increment.

ADC

Analog to **D**igital **C**onverter. A device that receives a band-limited continuous-time analog signal, and generates a sequence of binary numbers, at a sufficiently high sampling rate to allow full reconstruction of the analog signal from the samples. Also known as an “A/D,” and sometimes as an “AtoD” or “A2D.”

AGC

Automatic **G**ain **C**ontrol. A device that detects the peak-to-peak envelope, or average power of an incoming signal, and boosts or attenuates the signal so that the output envelope, or average power, is within some “small” range.

ASIC

Application **S**pecific **I**ntegrated **C**ircuit. An integrated circuit, such as a microprocessor or an Ethernet interface chip, designed for a specific purpose.

Channelizer

A process that extracts multiple channels, i.e. frequency sub-bands, from a wider band signal that contains them all.

Context Packet

A packet that communicates metadata, about a signal, about the processes generating a signal, or about the environment surrounding or related to the processes generating a signal.

Data Stream Packet

The Data Stream Packet is the basic data structure defined in this Standard to be used to send digitized IF data from a Source to a Destination.

DDC

Digital **D**own-**C**onverter. A process that band-limits a digitized signal and translates it down to a lower IF. Typically this also involves lowering the sample rate (down-sampling).

Demodulator

A process that extracts the information from a modulated carrier.

DF

Direction **F**inding. Any method that derives direction of arrival information from a received signal or signals.

IF Data

A band of spectrum, typically translated to an intermediate frequency (IF) range from a radio frequency (RF) range, and converted to a sequence of binary samples.

Discrete Fourier Transform (DFT)

A calculation that creates a frequency-domain representation of the information represented by a sequence of time-domain samples. Also often referred to simply as the “Fourier Transform” in contexts where it is understood to be discrete samples rather than continuous time.

Epoch (of a Data Packet)

The time period from the first to the last sample in the linked Data Stream Packet, inclusive.

FFT

Fast **F**ourier **T**ransform. A method for calculating a Discrete Fourier Transform in $N\log(N)$ operations rather than the N^2 required by a straightforward implementation of the Fourier Transform.

GPS

Global **P**ositioning **S**ystem. A system of satellites whose transmissions are used to calculate the position of the receiver. This term is also used to refer to the position-calculating receivers.

Higher Precision Time Code

A time code with better than the one-second precision provided by the Universal Time Code (UTC). This Standard defines three possible higher precision time references that can be used.

IF

Intermediate **F**requency - A frequency band, typically above the original base-band range of a signal, but below the intended transmit or receive band.

Meta-data

Data associated with a data sample but not part of the data sample, e.g. a channel number, or other encoded information, as specified by the equipment manufacturer.

MGC

Manual **G**ain **C**ontrol. A device or process that boosts or attenuates a signal under some external control.

Modulator

A device that modulates a carrier in order to impose information on it.

OUI

Organizationally **U**nique **I**dentifier. This is a 24-bit, IEEE-assigned, code that identifies the organization producing the product.

PDW

Pulse **D**escriptor **W**ord. A data structure describing temporal features of interest in a signal.

PLL

Phase-**L**ocked **L**oop. A process that locks a generated sinusoid or pulse train to a reference sinusoid or pulse train by comparing the phase of the generated signal with the phase of the reference, and adjusting the generated signal to hold the phase difference constant, typically zero.

RF

Radio **F**requency. A frequency used by a radio to transmit a signal. When referring to actual radio transmissions, this term can apply to any frequency above about 100 KHz. Sometimes used to imply very high frequency, in which context it usually implies a frequency above 100 MHz.

TBD

To **B**e **D**etermined.

TDOA

Time **D**ifference **O**f **A**rrival. A method for calculating position based on the fact that signals emitted or received from different locations are received at different times.

UTC

Universal **T**ime **C**ode. This is a 32-bit number containing the number of seconds since 00:00:00Z, January 1st 1970.

VITA-49
The term VITA-49 encompasses the entire family of VITA-49.x Standards.

VITA-49.0
This Standard.

VITA-49.x
Some future Standard in the VITA-49 family.

VRL
VITA Radio Link protocol (VITA-49.1). This protocol is an optional data link layer encapsulation for VRT Packet Streams.

VRP
VITA Radio Protocol. An umbrella term meaning the entire collection of VITA-49.X standards.

VRT
VITA Radio Transport (VITA-49.0) protocol. The protocol described in this standard.