VITA Radio Transport (VRT) Draft Standard

VITA-49.0 – 2007

Draft 0.21
31 October 2007

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

Approved by the VITA49 working group

VITA Standards Organization
PO Box 19658
Fountain Hills, AZ 85269

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

REVISION HISTORY

Version	Date	Comments & Major Changes/Updates
D0.21	31 Oct 2007	- Final Draft

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The VITA Radio Transport (VRT) standard defines a transport-layer protocol [12] 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. In support of this variety of applications, the VRT protocol provides a variety of formatting options that allow the transport layer to be optimized for each application. VRT also enables high-precision timestamping to provide time synchronization between multiple receiver channels. VRT is a useful building block to support data streaming for Software Defined Radio (SDR) applications.

The benefits of the VRT protocol include:

- Enables transport-layer **interoperability** between diverse equipment providers, reducing integration time and effort. This allows system infrastructures in which equipment from multiple vendors can be combined, so that technology insertion is simplified. Interoperability is accomplished by:
  - **Standardization of signal data transport** between receivers and signal processors with a wide variety of data types supported.
  - **Standardization of metadata transport** between receivers and signal processors, and **standardization of metadata types**. Standard metadata that may be conveyed include a receiver's geolocation and a variety of 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 one 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. This makes VRT ideal for distributed sensor applications.
- **Coherency** between multiple receiver channels for both real-time and recorded-data applications.
- **High-precision timestamping** 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 1-1shows a stove-pipe architecture using proprietary analog IF (intermediate frequency) interfaces transformed into an architecture using open VRT interfaces.

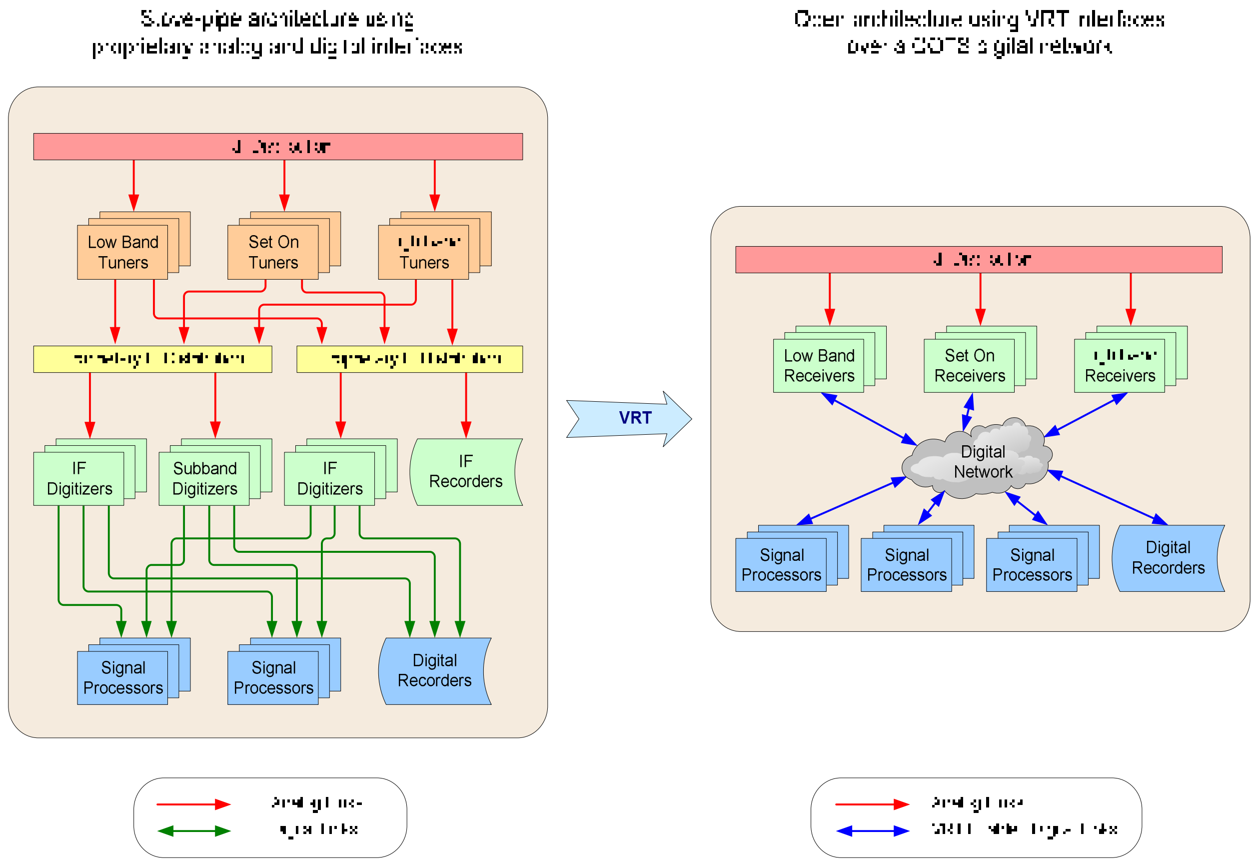


Figure 1-1: VRT Simplifies Traditional RF Signal Processing Systems
The combination of proprietary analog and digital connections are replaced with a COTS digital network, such as Ethernet.

The emergence of digitized IF, and high-speed serial interconnects that can carry it, allows system designers to replace complicated analog IF distribution networks with digital networks using commercial off-the-shelf (COTS) hardware. Such a network might be based, for example, on Gigabit Ethernet. VRT standardizes communication over such networks at the transport layer. As a result, equipment from a variety of providers can be inserted into the network with minimal impact on the application. VRT standardization also makes 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 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 enables transport-layer interoperability between equipment from multiple vendors. Full interoperability also requires that the underlying layers such as data-link and physical also be matched between equipment. VRT is designed to be independent of these underlying layers, therefore it may be carried over common protocols such as TCP, UDP, Serial RapidIO, Xilinx Aurora, Race++, Serial Front Panel Data Port (S-FPDP), PCI Express, and Gigabit Ethernet. Equipment that emit VRT compliant streams are always accompanied by supplementary documentation specifying which VRT features are used in the corresponding packet streams. This documentation can be used to gauge the level of interoperability between equipment.

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 ability 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 context information that can be conveyed includes:

- Frequency
- Bandwidth
- ADC sample rate
- Gain/attenuation settings
- Timestamp
- Timestamp delay (system latency)
- 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, thereby maximizing the amount of bandwidth available for signal data. The context information is timestamped so that changes can be precisely related to the associated data.

Context information can be sent for many components in a typical RF application, including:

- 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 integrated circuits 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.

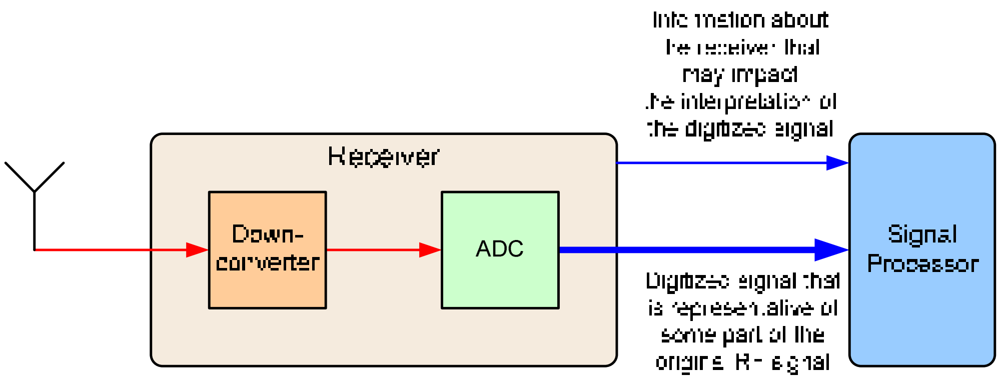


Figure 1.3-1: A simple VRT application

VRT is tailored to transport and describe signals in RF applications

In this example, VRT provides a standard data format for the Analog to Digital Converter (ADC) samples to be sent to the signal processor. VRT also provides a standardized means of conveying information, such as equipment settings, that 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. However, packetization comes at the expense of some overhead. The VRT protocol can minimize this overhead by allowing the volume of data, the format of data, and the type of data to be configured for optimal link utilization. Thus the overhead of the protocol can typically be configured to be a small fraction of the overall signal data bandwidth, providing 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 processing 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 that demonstrate the use of fields in the VRT Context packet.

Appendix C contains an example that demonstrates the use of the VRT Extension packets.

Appendix D contains an explanation and examples of VRT floating-point numbers.

2 VITA Information

2.1 *Working Group Members*

The Working Group for this standard had the following sponsors and observers:

Company	Name	Status
Acqiris	William Accolla	Observer
Applied Signal Technology	Larry Corsa	Observer
Applied Signal Technology	Matthew Cottrell	Observer
Boeing	William A. Hanna	Observer
Bustronic Corporation	Melissa Heckman	Observer
Curtiss Wright	Ian Stalker	Observer
Curtiss Wright	Jeff Smith	Observer
Curtiss Wright	Lee Brown	Observer
Curtiss Wright	Paul Davis	Observer
Curtiss Wright	Steve Edwards	Observer
Digital Receiver Technology	Richard Shaner	Sponsor
Digital Receiver Technology	Mary Parker	Participant
DRS Signal Solutions	James Taylor	Observer
DRS Signal Solutions	Richard Sims	Sponsor
DRS Signal Solutions	Robert Normoyle	Sponsor
DRS Signal Solutions	Robert Watson	Observer
DRS Signal Solutions	Steve Byrd	Participant
Echotek	Ray Blakeman	Observer
Eclipse Electronic Systems	Conrad Romberg	Sponsor
Eclipse Electronic Systems	Jeffrey Sanders	Observer
Foxconn Electronics	Elwood Parsons	Observer
GE FANUC	Chris Eckert	Observer
GE FANUC	Jonathan Jones	Participant
Mercury Computer Systems	Chris Kent	Observer
Mercury Computer Systems	Craig Lund	Sponsor
Mercury Computer Systems	Dean Holman	Participant
Mercury Computer Systems	Marc Trepanier	Participant
Mercury Computer Systems	Stephen Pereira	Participant
Nallatech	Malachy Devlin	Observer
Pentek	Paul Mesibov	Sponsor
Pentland Systems	John Walker	Observer
Praxis Engineering Technologies	Dave Sherwood	Participant
Raytheon	Doug Endo	Observer
Spectrum Signal Processing	Aaron Kaiway	Sponsor
TEK Microsystems	Andy Reddig	Observer
TEK Microsystems	D Bosworth	Observer
Tyco Electronics	Robert Patterson	Observer
VMETRO	Jeff Bateman	Observer
VMETRO	Terje Melsom	Observer

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

Company	Representatives
Digital Receiver Technology	Richard Shaner
DRS Signal Solutions	Robert Normoyle, Richard Sims
Eclipse Electronic Systems	Conrad Romberg
Pentek	Paul Mesibov

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

Name	Company

During the course of the development of this specification the active participation in the working group evolved significantly. The following two individuals, although not active in the working group at the time of the final ballot, made substantial contributions:

- Aaron Kaiway, Spectrum Signal Processing
- Stephen Pereira, Mercury Computer Systems

2.2 *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.

2.3 *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.

2.4 *Change Bars*

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

2.5 *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.

3 References and Terminology

The following publications are either referenced by or 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)
1. ANSI/VITA 17.1-2003, Serial Front Panel Data Port specification.
2. VITA-41.0-200X VXS VME bus Switched Serial Standard specification
3. VITA-46.0-200X, VPX specification

The following are available from their respective maintainers:

4. RapidIO™ Interconnect Specification, Part 6: 1x/4x LP-Serial Physical Layer Specification
5. Global Positioning System Standard Positioning Service Signal Specification, 2nd Edition, June 2, 1995 - http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf
6. National Marine Electronics Association, NMEA 0183, Standard for Interfacing Marine Electronic Devices, Version 3.01, January 1, 2002
7. PICMG 2.18, Serial RapidIO specification
8. ANSI/IEEE 802 specifications
9. IEEE 754-1985 Standard for Binary Floating Point Arithmetic
10. IEEE OUI Request Form - http://standards.ieee.org/regauth/oui/forms
11. ITU-R TF.460-6 Standard-frequency and time-signal emissions
12. Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, ISO/IEC International Standard 7498-1 : 1994 (E)

3.1 *Requirements Terminology*

To avoid confusion and to make the requirements for compliance very clear, 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
- Documentation Rule
- Documentation Recommendation

Any text not labeled with one of these keywords should be interpreted as descriptive in nature. These are 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 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. All rules use "shall" or "shall not" to emphasize the importance of the rule. The words "shall" and "shall not" 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 words "should" or "should not" to emphasize the importance of the recommendation. The words "should" and "should not" 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 that is helpful but not vital. The reader is encouraged to consider the advice before discarding it. Some design decisions 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 word "may" to emphasize the importance of the permission. The word "may" is reserved exclusively for stating permissions in this standard and is 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.

Documentation Rule <Section>-<number>:

A documentation rule is simply a rule that governs required documentation. It is expressed using “shall” and/or “shall not” to emphasize that conformance to the specification requires conformance to these rules also.

Documentation Recommendation <Section>-<number>:

A documentation recommendation is a recommendation that applies to documentation. It is expressed using “should” or “should not” just as are other recommendations.

3.2 *General Technical Terminology*

ADC

Analog-to-Digital Converter: 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 Gain Control: 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 nominal range.

ASIC

Application Specific Integrated Circuit: 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 subbands) from a wider band signal that contains them all.

DDC

Digital Downconverter: 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 Finding: Any method that derives direction of arrival information from a received signal or signals.

FFT

Fast Fourier Transform. 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.

GMT

Greenwich Mean Time: The time in Greenwich England.

GPS

Global Positioning System: 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.

IF

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

lsb

Least-Significant Bit: Labeled as bit position 0 in this document.

LSB

Least-Significant Byte

Metadata

Data associated with signal data but not contained within it, e.g. a channel number or other encoded information, as specified by the equipment manufacturer.

MGC

Manual Gain Control: 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.

msb

Most-Significant Bit

MSB
Most-Significant Byte

OUI
Organizationally Unique Identifier: A 24-bit, IEEE-assigned code that identifies the organization producing the product. (See [10])

PDW
Pulse Descriptor Word: A data structure describing temporal features of interest in a signal.

PLL
Phase-Locked Loop: 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 Frequency: 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 Be Determined.

TDOA
Time Difference Of Arrival: A method for calculating position based on the fact that signals emitted or received from different locations are received at different times.

Upstream
A previous point in the flow of processing of signal. An upstream process is a process whose output data affects the output data of other (downstream) processes. An upstream signal is a signal generated by an upstream process.

UTC
Coordinated Universal Time: Also known Greenwich Mean Time, which it replaced. UTC is calculated from a collection of clocks around the world.

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

VITA-49.0
This standard.

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.

3.3 *VRT-Specific Terminology*

The following terms are either defined specifically for this specification, or are used in a manner somewhat different within this specification than in other contexts. In this document these terms always start with a capital letter.

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 primary application of VRT is for the packetized transport of time domain digital samples of RF, IF, or baseband signals, along with any necessary metadata, across a digital link. Any such digitized signal is herein referred to as *IF Data*. Metadata specifically assigned to these signals, for example RF center frequency and power level, are herein referred to as *IF Context.* These terms, and others that will be used extensively throughout this document, are defined below.

Definition 4-1: A Data Sample is a digital representation of an analog signal at a point in time. A Data Sample may be either real or complex valued. A sequence of Data Samples constitutes a time-domain representation of the whole signal.

Definition 4-2: IF Data is a sequence of digital Data Samples of an RF, IF, or baseband signal. These samples represent the information from some region of RF spectrum. Typically this region has been frequency translated to an IF or to baseband for transport. The time-domain samples are taken at a constant sample rate. Examples include:

- The output of an ADC that samples an analog IF signal
- The output of a DDC whose input was the ADC just described

Definition 4-3: The Described Signal is the signal for which Context packets are providing additional metadata.

Definition 4-4: IF Context is certain common metadata that provides a more complete description of the Described Signal or of the circumstances surrounding its reception or processing. The collection of metadata defined as IF Context is given in Section 7. Examples include:

- RF frequency
- Power information
- Timing
- Antenna GPS location.

In addition to the transport of IF Data and Context, VRT supports the transport of a wide variety of other signal types and metadata. Any type of signal data other than IF Data is referred to herein as *Extension Data*.

Definition 4-5: Extension Data is data representing any signal that is not an RF, IF, or baseband signal. In typical VRT applications such data would be derived, directly or indirectly, from a region of RF spectrum. Examples include:

- Pulse-Descriptor Words (PDWs)
- A demodulated bit stream
- An FFT of a signal

VRT also supports *Extension Context*. This is context of a type that does not fall into the category of IF Context for IF Data. That is, it is any metadata that is not part of the predefined “standard” collection given in Section 7 of this specification.

Definition 4-6: Extension Context is metadata that is not part of the standard metadata set described in Section 7. Examples of Extension Context include:

- Vendor-specific information such as equipment serial numbers or operator information
- The estimated carrier frequency and symbol rate of a PSK signal being demodulated

Definition 4-7: Data is an alias that may mean IF Data, or Extension Data, or both IF and Extenstion Data. This alias is used for brevity, but only used when no ambiguity results.

Definition 4-8: Context is an alias that may mean IF Context, or Extension Context, or both IF and Extenstion Context. This alias is used for brevity, but only used when no ambiguity results.

As just described, VRT supports the transport of four types of information: IF Data, IF Context, Extension Data, and Extension Context. VRT sets forth rules controlling the structure and function of packets that carry these four types of information. It also specifies a way to associate together all the different packet streams that carry the different portions of the information related to a signal or set of related 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 when the meaning is clear simply as a *Packet Stream*. The collection of VRT Packet Streams needed to convey all the required information about a signal, or signals, is referred to as a *VRT Information Stream*, or simply as an *Information Stream*.

Definition 4-9: 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 IF Data, or IF Context, or Extension Data, or Extension Context.

Definition 4-10: A VRT Information Stream is a collection of VRT Packet Streams, associated together according to VRT rules in order to transport a complete set of Data and Context for one or more related signals. In general, each Packet Stream in an Information Stream may contain packets of a different type.

The relationship between VRT Information Streams and Packet Streams is shown in Figure 4-1. It depicts three systems each communicating with a fourth system over a data link using the VRT protocol.

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, and unrelated to the VRT standard, 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. Figure 4-1 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 (white, black, or grey) 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 GPS information while the second conveys the model and serial numbers of installed modules. VRT supports putting any number of Context Packet Streams into an Information Stream. Only one Data Packet Stream is allowed in an Information Stream however.[*]

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.

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 provides 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 key constructs such as the VRT Information Stream.

[*] *Although only one Data Packet Stream is allowed in an Information Stream, this does not mean that an Information Stream can carry only one Described Signal. A Data Packet Stream may contain multiple channels, and thus multiple Described Signals.*

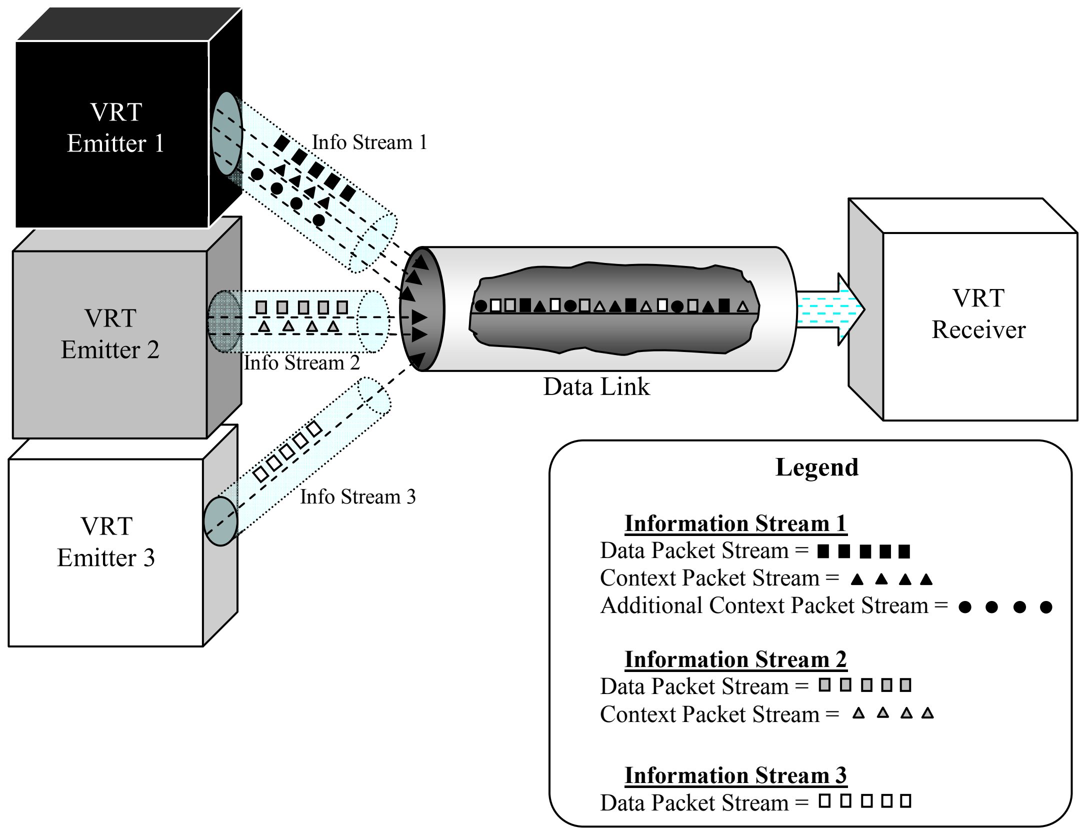


Figure 4-1: Three VRT Information Streams on a Data Link. Each Information Stream consists of one or more VRT Packet Streams. Each Packet Stream is dedicated to conveying a particular portion of the Information Stream to the VRT receiver.

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 4.1-1 shows the use of VRT transport both for intermodule and for intersystem communication. Within the tuner system the tuner module emits a VRT Packet Stream (1) 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 emits this in another VRT Packet Stream to the high-speed serial interface. The DDC also emits a Context Packet Stream along with the IF Data Packet Stream. Together these two Packet Streams constitute another VRT Information Stream (2). The high-speed serial interface receives this Information Stream and additional tuner Context (3) that is not in VRT format. It combines the tuner Context with the DDC Information Stream to create a new Information Stream (4) 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. It does this in such a way that the signal processor can sort out all the received packets and verify that all of the intended Packet Streams are being received.

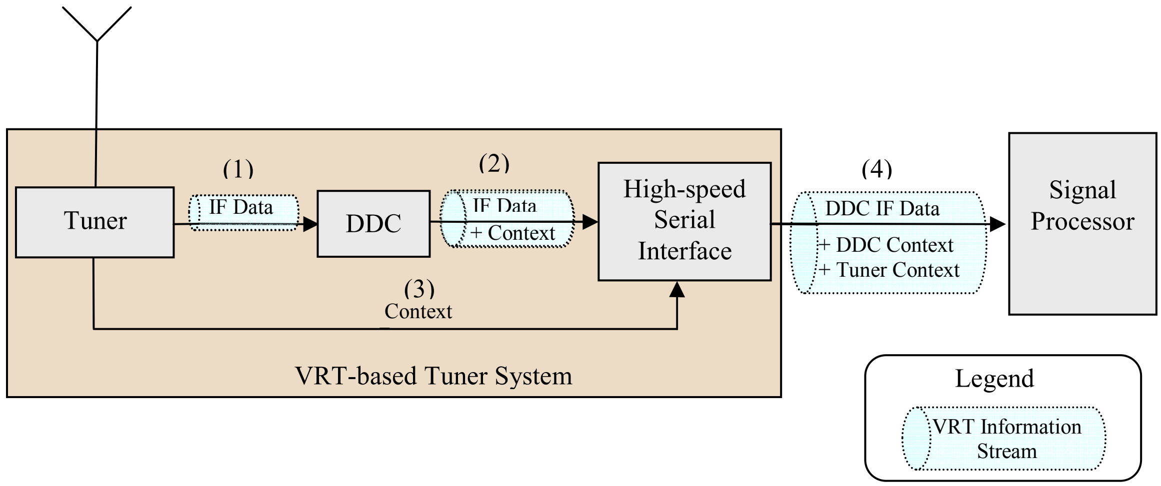


Figure 4.1-1: A System Involving Four VRT Information Streams. The tuner outputs one IF Data Information Stream and one non-VRT Context stream. The DDC outputs one Information Stream consisting of two Packet Streams. The system outputs an Information Stream consisting of three Packet Streams.

Every Information Stream is as unique as the application behind it. The simplest possible Information Stream contains 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 Information Stream complexity, the structure and function of all constituent Packet Streams is precisely defined by rules within this specification and by documentation that must be provided for VRT compliance.

4.1.1 VRT Packet Streams

As previously described, VRT supports the transport of four types of information: IF Data, IF Context, Extension Data, and Extension Context. Correspondingly, VRT defines four types of VRT Packet Streams, as seen in Table 4.1.1-1. As the table shows, there are two types for conveying Data and two types for conveying Context.

IF Data Packet Streams convey IF Data. The packet formats for this type of Packet Stream are more tightly regulated in order to facilitate interoperability. Conversely Extension Data Packet Streams may be used to convey data of any other type, so the payload portion of these packets is largely unregulated. IF Context Packet Streams convey IF Context. As with IF Data Packet Streams this is considered a standard application, so the packet formats are more tightly regulated in order to facilitate interoperability. Extension Context Packet Streams, by contrast, may convey any other kind of Context. Therefore the payload portion of these packets is largely unregulated.

The collection of two types of VRT Packet Streams for Data and two types for Context provides for both a standard way to convey IF Data and Context, and a flexible method to convey the many other types of Data and Context that may be present in a VRT system.

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 6 and Section 7, respectively.

Definition 4.1.1-1: An IF Data Packet Stream is a VRT Packet Stream that conveys IF Data.

The IF Data Packet Stream forms the heart of the VRT Standard. It conveys one or more digitized IF channels. Section 6.1 describes the IF Data Packet Stream.

Contents	Standard Formats	Custom Formats
Data	**IF Data Packet Stream** Conveys a digitized IF signal (IF Data) • 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 type of data • Custom packet format
Context	**IF Context Packet Stream** Conveys common Context for IF Data • Frequency • Power • Timing • Geolocation • etc.	**Extension Context Packet Stream** Conveys additional Context for IF Data or Extension 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.

Definition 4.1.1-2: An Extension Data Packet Stream is a VRT Packet Stream that conveys Extension Data. This may be whatever Data the application requires. The packets used in this type of Packet Stream typically use a format tailored to the purpose.

The Extension Data Packet Stream is intended to extend the applicability of VRT to a wider range of applications than simple IF Data. Section 6.2 describes the Extension Data Packet Stream.

Definition 4.1.1-3: An IF Context Packet Stream is a VRT Packet Stream that conveys IF Context.

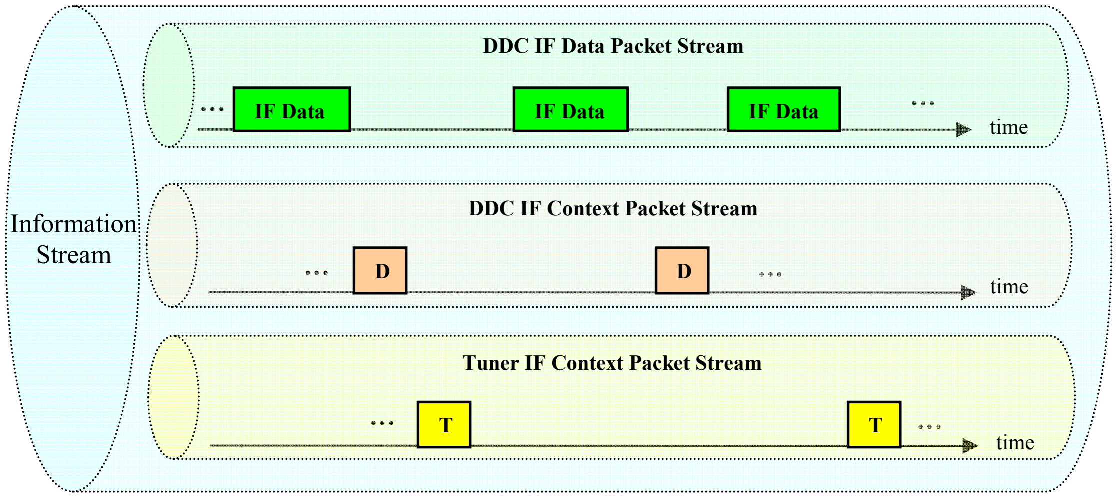
Section 7.1 describes the IF Context Packet Stream.

Definition 4.1.1-4: An Extension Context Packet Stream is a VRT Packet Stream that conveys Extension Context. The packets used in this type of Packet Stream typically use a format tailored to the purpose.

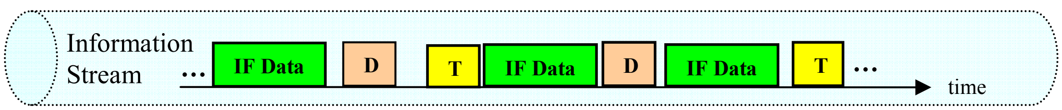
Section 7.2 describes the Extension Context Packet Stream.

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 were 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.

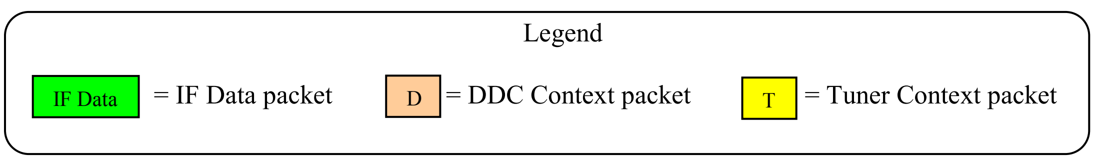


Figure 4.1.2-1: The Components of an Example VRT Information Stream. This Information Stream consists of three VRT Packet Streams, each of a different type (a). The three Packet Streams are interleaved onto a data link (b).

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 would have a relatively high rate of packet transmission while an associated Context Packet Stream would have 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. It is embedded in the packets of the Packet Stream. The number is identical in all the packets of a given 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 Sections 4.1.3.2.8 and 7.1.5.25.

In order to fully define an Information Stream, it is necessary to specify all packet types used in that Information Stream and the associations 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 controlling how the structure and function are to be specified. For the purpose of specifying Packet Streams and Information Streams, VRT borrows aspects of the object-oriented programming concept of a *class*. In the context of VRT the term C*lass* 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: A VRT Packet Class is the specification of the name, structure, and function of the packets in a VRT Packet Stream. Specifically, it specifies the following:

1. **Packet Class name and code**
   Every Packet Class is assigned a name. This name typically reflects its purpose. The code is a number, unique within an organization, which identifies the Packet Class. This number can be inserted into packets to identify their structure and function.

2. **The structure of the Packet Class**
   The structure specification describes the format and location of each field in the packet.

3. **The function of the Packet Class**
   The functional specification describes the interpretation of each field in the packet, as well as the purpose of the Packet Stream.

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 different 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 created Packet Stream would 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.

It should be noted that a given Packet Class may be used to create multiple Packet Streams within an Information Stream and may also be used for multiple Information Streams. The only requirement is that every packet in a Packet Stream must originate from the same Packet Class. For example, the same IF Data Packet Class may be used for 16 different output channels from a bank of 16 DDCs. In this case, the packets (the data format, data packing, etc.) are all identical. The only difference between these Packet Streams is that each carries data for a different channel.

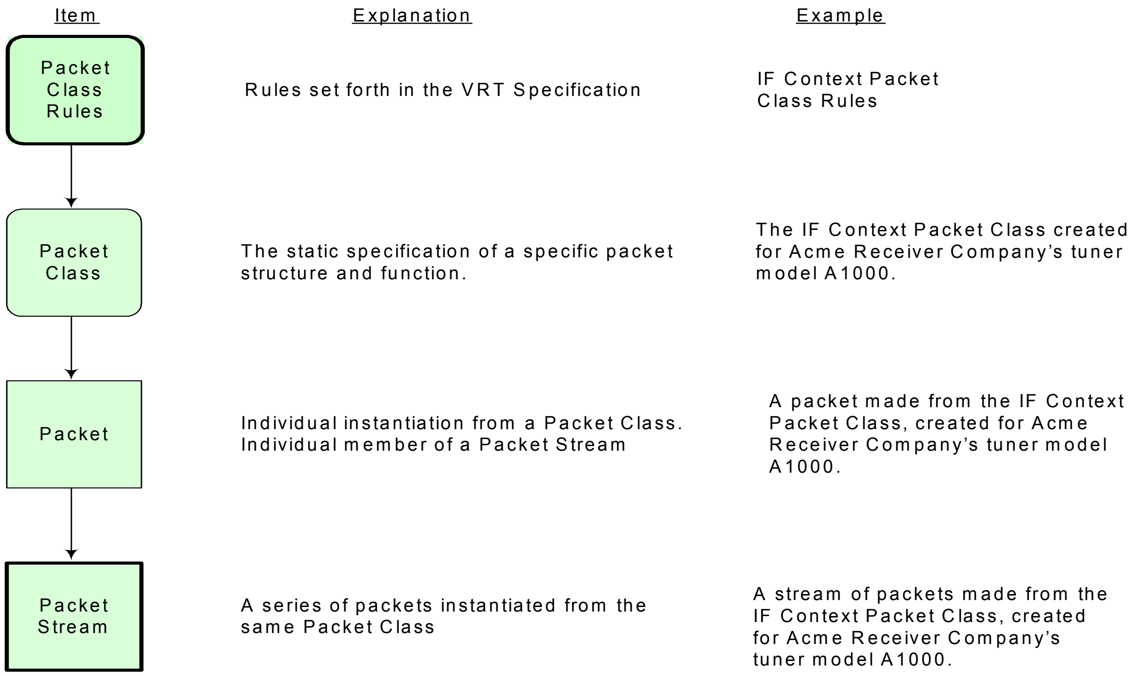


Figure 4.1.3.1-1: From the VRT Specification to a Resulting Application-Dependent Packet Stream.

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.

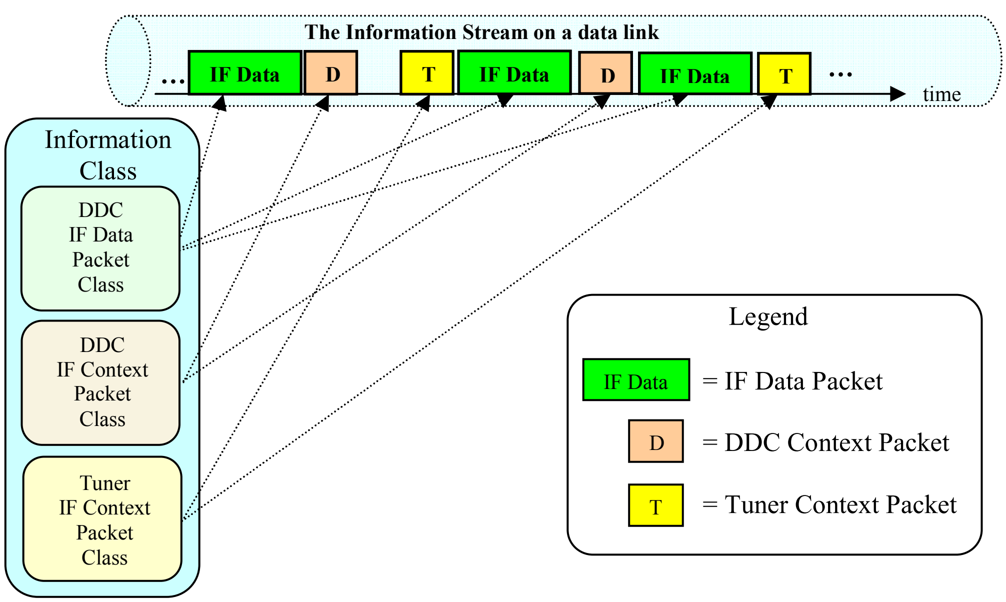


Figure 4.1.3.1-2: Three VRT Packet Streams made from Three Packet Classes.
Every time a Packet is transmitted it is based on the corresponding predefined Packet Class.

4.1.3.2 Information Classes

Definition 4.1.3.2-1: A VRT Information Class is a specification of the structure and function of the Information Stream. It acts as a controlling document for the creation of an Information Stream, or possibly an arbitrary number of functionally identical Information Streams.

There are eight components to an Information Class. These components encompass every aspect of the Information Stream. An Information Class includes the Packet Classes for the included Packet Streams, as previously 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. The eight components are listed below. For an example Information Class, see Appendix A.

1. Class Name and Code
2. Information Stream Purpose
3. Names of included VRT Packet Streams
4. Purpose of each included Packet Stream
5. Packet Classes
6. Packet Stream Details
7. Reference Points
8. Packet Stream Associations

These components are explained in the following subsections.

4.1.3.2.1 Class Name and Code

An Information Class is given a name as an identifier. This name does not appear in the resulting Information Stream. It is for use by system designers to refer to the type of Information Stream being created. The Information Class Code is a number assigned to the class. This number can be inserted into packets made from the class to identify the class defining their structure and function.

4.1.3.2.2 Information Stream Purpose

The Information Class must state the purpose for the resulting Information Stream(s). This purpose may be very general such as, "To convey any 70 MHz IF Data," or may be more specific, such as "To convey 70 MHz IF Data from the A1000 tuner to the signal processor in system S1000." In general, Information Streams with more constituent Packet Streams are likely to have more specific statements of purpose.

4.1.3.2.3 Names of included VRT Packet Streams

Each Packet Stream within an Information Stream is given a name. These names do not appear in the Information Stream. They are for use by system designers to refer to the Packet Streams within an Information Stream. Each Packet Stream name within an Information Stream must be unique. It will be reused however across Information Streams made from the same Information Class. It is often convenient to choose a Packet Stream name that indicates the purpose of the Packet Stream.

4.1.3.2.4 Purpose of each included Packet Stream

An Information Class must include a statement of the purpose of each Packet Stream included in the resulting 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 generally more specific than the purpose stated within the corresponding Packet Class since the Packet Class may be used in multiple Information Classes. Therefore, the purpose specified within the Packet Class generally specifies only the type of information conveyed rather than particulars related to the application.

For example, if the purpose of a Packet Stream stated in the Packet Class is "to convey IF Data in real 16-bit samples," the purpose for that Packet Stream stated in the Information Class might be, "To convey the IF Data in real 16-bit samples from the A1000 tuner made by the Acme Receiver Company." The Packet Stream purposes are stated as an aid to understanding the structure and uses of the resulting Information Streams.

4.1.3.2.5 Packet Classes

An Information Class must include all of the Packet Classes needed to build the Information Stream. Each Packet Class specifies the structure and function of a type of Packet Stream as previously described. A separate Packet Class is provided for use in creating each type of Packet Stream in the Information Stream. The same Packet Class may be used to create multiple Packet Streams of the same type, however.

4.1.3.2.6 Packet Stream Details

In addition to the information provided by the Packet Stream purpose statement and the associated Packet Class, an Information Class may include any other information that is useful for the application. The exact information depends on the application. One example would be a statement of the expected packet rate for each Packet Stream. Another would be the conditions under which each type of Context packet would be sent.

4.1.3.2.7 Reference Points

A typical system emitting VRT Packet Streams will consist of more than one processing subsystem. Often it is desirable to emit Context information related to the Described Signal as it was at one or more upstream points in the system. It is also often desirable to emit Context related to the processing of that signal at one or more upstream points. For example, suppose the Described Signal is a demodulated bit stream from an RF carrier. One might wish to convey the center frequency and power level of that Described Signal as it was in its modulated form at the antenna, while also conveying the bit rate and estimated bit error rate (BER) from the demodulator. A *Reference Point* is a point in a system where the Context applies. In the example just given, the antenna would be one Reference Point and the demodulator would be another. The antenna would be the Reference Point for the RF frequency and signal power, while the demodulator would be the Reference Point for the bit rate and BER. These different Context items in a sense relate to different signals, but they also directly relate to the Described Signal since it traces its origin through all these points and intermediate signal formats. The locations of such Reference Points are specified in the Information Class.

Definition 4.1.3.2-2: A Reference Point is a point in a system where a Timestamp, or other signal-related Context conveyed in an IF Context or Extension Context packet is understood to properly explain the Described Signal. A Reference Point may indicate a VRT Data Packet Stream, an analog signal, or a digitized signal in any non-VRT format.

Appendix B contains several examples of Reference Point utilization.[*]

[*] *B.1 provides an example of Reference Points for frequency and power level Context. B.3, illustrates Reference Points in a multichannel Context application. B.8 shows how Reference Points and the Timestamp Adjustment feature can compensate for processing delay in a system.*

4.1.3.2.8 Packet Stream Associations

The purpose of an Information Stream is to convey some IF or Extension Data and all related Context.* Depending on the purpose of the system and the type of Data, different kinds of Context will be conveyed along with the Data. When more than one Context Packet Stream is needed in an Information Stream it is useful to organize the Context into separate Packet Streams according to some obvious categories. This facilitates the proper interpretation of the contents of the various Context packets by the system receiving them. VRT provides five types of Packet Stream associations that can be used to identify which pieces of Context are conveyed in which Context Packet Stream. The five types of associations are:

- Data-Context pairing
- Source Context association
- Vector-component Context association
- Asynchronous-channel Context association
- System Context association

The first of these associations, Data-Context pairing, defines a unique one-to-one relationship between a Data Packet Stream and a Context Packet Stream. This unique association to the Data Packet Stream is typically reserved for the Context Packet Stream that is most critical for the interpretation of the Data. Data-Context pairing is indicated on a link by the sharing of a Stream Identifier code between the paired Packet Streams. Pairing is the only instance in which it is allowable to use the same Stream Identifier for two Packet Streams in an Information Stream.

The remaining four types of associations are used to append various types of additional Context to a Data-Context pair. Each of the four remaining types of associations allows for a particular type of additional Context to be associated with the Data Packet Stream. Such additional Context Packet Streams are intended for specific types of applications, and are optional. The inclusion of such additional Context into an Information Stream is indicated on a link by the contents of *Context Association Lists*. These are explained in detail in Section 7.1.5.25.

Data-Context pairing is discussed further in the next section. The other four types of associations are discussed in subsequent sections.

4.1.3.2.8.1 Data-Context Pairing

Data-Context pairing is the primary method of associating a Context Packet Stream with a Data Packet Stream. It associates with the Data Packet Stream carrying the Described Signal a Context Packet Stream bearing information that is most critical for the interpretation of that Data. For example, a paired Context Packet Stream would typically include the RF center frequency, sampling rate and the numbering format used in the Data Packet Stream. Not only is Data-Context pairing the primary method of associating Context with a Data Packet Stream, in many cases it is possible to pack all of the necessary Context information into the paired Context Packet Stream. In such cases Data-Context pairing provides a simple way to include all required Context for the Described Signal without the complexity of associating additional Context Packet Streams. Data-Context pairing is represented pictorially in this document as shown in Figure 4.1.3.2.8.1-1.

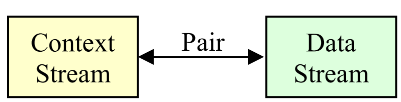


Figure 4.1.3.2.8.1-1: Pictorial Representation of a Data-Context Pair

The boxes represent packet streams and the bidirectional arrow indicates that they form a Data-Context pair. Data Stream carries the Described Signal.

---

* *A VRT Information Stream could possibly consist of only Context Packet Streams. In this case the Context Packet Streams would provide Context for some non-VRT data stream or analog signal.*

4.1.3.2.8.2 Source Context Association

Source Context associations organize Context into a set of Packet Streams that mirrors the sequence of processing steps in the system generating the Described Signal. One example would be a system composed of a downconverter, an ADC, and a demodulator. In this case there might be three Context Packet Streams with each conveying Context for one of the three processing steps. A Source Context association relates one Context Packet Stream to another such that it is clear which Context comes from the upstream process and which comes from the downstream process. Source Context associations are represented pictorially in this document as shown in Figure 4.1.3.2.8.2-1.

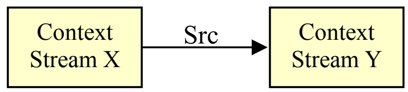


Figure 4.1.3.2.8.2-1: Pictorial Representation of a Source Context Association
The boxes represent packet streams and the arrow labeled “Src” indicates that Stream X contains Context for a Data source for the process generating Stream Y.

Here Context Packet Stream X is understood to relate to the process immediately upstream of the process to which Context Packet Stream Y relates. The arrow can be thought of as pointing downstream in the signal flow. A chain of Source Context associations can be made to reflect, and provide Context for, an entire chain of processing stages.

4.1.3.2.8.3 Vector-component Context Association

Vector-component Context associations are used when the Data Packet Stream contains vector data. In such cases there is usually Context information related to each component of the data vectors. For example, there may be a gain factor associated with each component that needs to be taken into account when calculating eigenvectors. Vector-component Context associations allow for a separate Context Packet Stream to provide Context for each vector component within a Data Packet Stream. Vector-component Associations are represented pictorially in this document as shown in Figure 4.1.3.2.8.3-1.

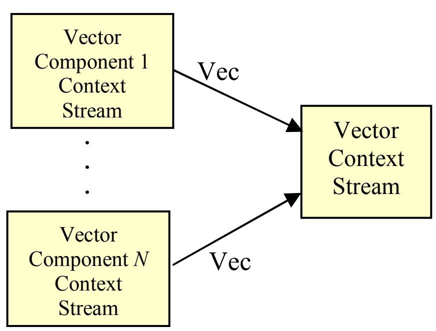


Figure 4.1.3.2.8.3-1: Pictorial Representation of Vector Context Association
The boxes represent Packet Streams and the arrows labeled “Vec” each point from a stream containing Context for one component of the vector to the Context stream for the vector as a whole.

Here each of the *N* Vector-component Context Packet Streams provides Context information that is related to exactly one component of the vector. These *N* Packet Streams are each associated by Vector-component Context association to the “Vector Context” Packet Stream, which carries Context information that relates to the vector signal as a whole. Each arrow points from the component-specific Context stream to the Context stream for the vector as a whole.

4.1.3.2.8.4 Asynchronous-Channel Context Association

A single Data Packet Stream may be used to convey multiple asynchronous data channels. When this occurs, a separate Context Packet Stream will generally be required to provide Context for the Data in each of the channels. This is done via Asynchronous-Channel Context association. Asynchronous-Channel Context associations are represented pictorially in this document as shown in Figure 4.1.3.2.8.4-1.

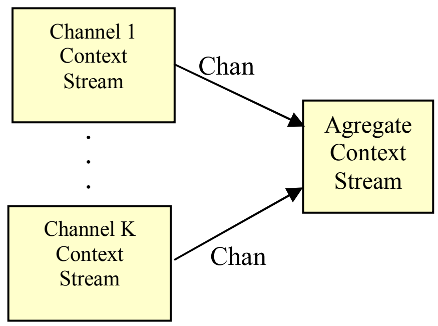


Figure 4.1.3.2.8.4-1: Pictorial Representation of Asynchronous-Channel Context Association

The boxes represent Packet Streams and the arrows labeled "Chan" each point from a stream containing Context for one channel to the Context stream common to the collection of channels.

Each of the *K* Asynchronous-Channel Context Packet Streams shown in the figure provides Context information for exactly one of the channels carried in the associated Data Packet Stream. Each channel-specific Packet Stream is associated by Asynchronous-Channel Context association to the "aggregate" Context Packet Stream, which contains Context common to the entire collection of channels. Each arrow in the figure points from the channel-specific Context stream to the aggregate Context stream. Note that typically the aggregate Context Packet Stream is paired with the Data Packet Stream carrying the Described Signals.

4.1.3.2.8.5 System Context Association

System Context associations are used to attach general Context to an Information Stream. Such Context might include device serial numbers, power supply voltages, temperatures, etc. System associations are represented pictorially in this document as shown in Figure 4.1.3.2.8.5-1.

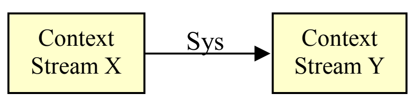


Figure 4.1.3.2.8.5-1: Pictorial Representation of a System Context Association

The boxes represent packet streams and the arrow labeled "Sys" points from the stream containing system Context to the stream containing more signal-specific Context.

Here it is understood that Context Packet Stream X adds system Context to the information already provided by Context Packet Stream Y. The arrow points from the more general system Context Packet Stream to the (typically) more signal-specific Context Packet Stream with which it is being associated.

4.1.3.2.8.6 Collections of Associations

In the following discussion we refer to Packet Stream associations as being either *direct* or *indirect*. When we describe an association as direct we simply mean that it is via either Data-Context pairing* or one of the other four types of association described in the four previous sections. When we refer to Packet Stream associations as being indirect, we mean that they are inferred by a chain of two or more direct associations. Figure 4.1.3.2.6-1 illustrates some direct and indirect Context associations in a hypothetical Information Stream.

Each of the four types of optional Context associations described in the previous sections is used to associate a specific type of additional application-specific Context with a Data Packet Stream. Any such additional Context Packet Streams are always associated indirectly with the Data Packet Stream however. That is, there is always at least one Context Packet Stream that serves as an intermediary between the Data Packet Stream and these additional Context Packet Streams. The one required intermediary is the Context in a Data-Context pair, as shown in the figure.

The figure also shows how an indirect Context association may be inferred by a longer chain of direct associations. Ultimately every Context Packet Stream in an Information Stream is associated with the Data Packet Stream in that Information Stream by some collection of direct associations. In this example all the Packet Streams are associated into the Information Stream by a collection of three direct associations, two source Context associations and one Data-Context pairing.

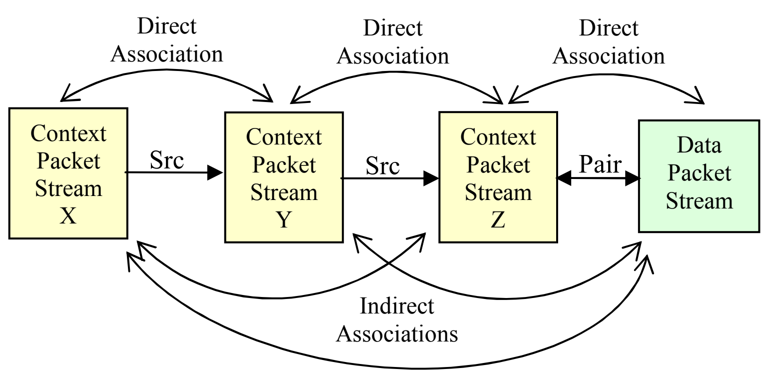


Figure 4.1.3.2.6-1: Some Direct and Indirect Context Associations

The specification of Packet Stream associations in an Information Class may take the form of a diagram, as shown in the figure, or of an outline-like text description, or possibly some other form devised by the creator of the Information Class.

4.1.3.2.9 An Information Class Example

The eight components of an Information Class are demonstrated below. Figure 4.1.3.2.9-1 shows a system that emits a non-trivial Information Stream. This Information Stream consists of an IF Data Packet Stream and an IF Context

---

* *Although we here refer to Data-Context pairing as "direct," we always use the the term "paired" vs. "directly associated" in this document to avoid confusing pairing with the other types of direct association, which are quite different in their implementation.*

Packet Stream from the adaptive combiner, plus a separate IF Context Packet Stream from each of the tuners and from each of the DDCs. Altogether this Information Stream includes six Packet Streams.

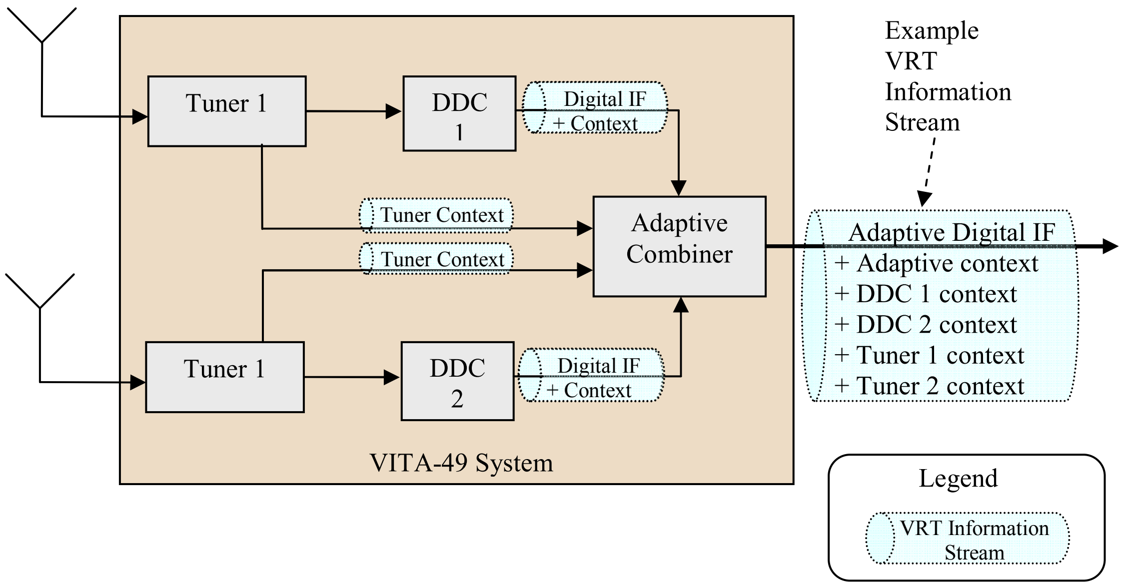


Figure 4.1.3.2.9-1: An Example System Emitting a Non-trivial Information Stream
The IF Data Packet Stream from the adaptive combiner is transported over the link along with Context Packet Stream from the adaptive combiner and from each of the four upstream modules.

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 adaptive combiner output 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:
   - For example, the purpose of the Adaptive Combiner IF Data Packet Stream is “To convey Adaptive Combiner output Data to the follow-on processor”
5. The Packet Classes from which the Packet Streams are made:
   - For every different type of Packet Stream in the Information Stream, provide Packet Class documentation. This is explained in detail in Section 8, and Appendix A has detailed examples.
6. Packet Stream Details:
   - For example, “Adaptive Combiner Data packets are transmitted at a rate of 100 packets per second”
   - For example, “Context packets are transmitted whenever any Context information changes”
7. Reference Points:
   - Tuner Context always references the tuner input.
   - DDC Context always references the tuner input.
   - Adaptive Combiner Context always references the tuner inputs.
8. A specification of the Packet Stream Associations:
   - Figure 4.1.3.2.9-2 shows a Packet Stream association diagram for this example. The arrows connecting the Packet Streams indicate the direct associations. The collection of direct associations also serves to indirectly associate all of the Context with the Data Packet Stream. This associates all the Packet Streams into one Information Stream.

Note that the association diagram bears a strong resemblance to the architecture of the system. This desirable result comes from the proper use of Source Context associations. The arrows indicate the direction of signal flow in the system. As previously explained, the bidirectional arrow between the adaptive combiner IF Data Packet Stream and the adaptive combiner IF Context Packet Stream indicates Data-Context pairing. This example of Data-Context pairing is typical. It creates a unique association between the Data Packet Stream and the Context Packet Stream from the process that actually produces the output Data. All other associated Context Packet Streams are related indirectly to the Data through the chain of associations shown.

As previously stated, other types of associations between Context Packet Streams may also be specified based on Vector-component, Asynchronous-Channel, or System Context. Also, any combination of these associations may be employed to best organize an Information Stream. The provision for these types of associations, and for their conveyance in the transmitted packets, facilitates auto-discovery of the architecture of a VRT system by the receiver of the Information Stream.

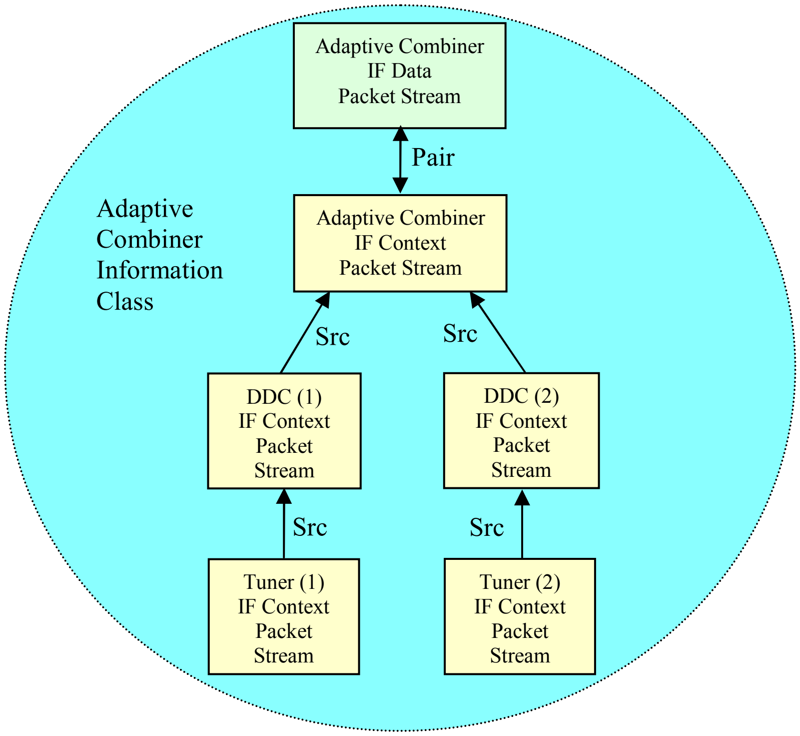


Figure 4.1.3.2.9-2: The Association of Packet Streams within the example Information Stream. One Data-Context pairing and four Source Context associations are used.

4.2 *VRT Timestamps*

The capability to Timestamp Data and Context is one of the most important capabilities provided by VRT. VRT Timestamps allow for the precise determination of when a signal was received or when it was processed. VRT Timestamps also allow for the precise indication of the timing of Context such as PLL lock, signal detection, or ADC overflow.

VRT Timestamps convey the timing of Data and/or events as seen at some specified Reference Point* in the system. The choice of a Reference Point is necessary because, in general, the propagation or processing of a signal introduces delay. Figure 4.2 shows a simple system illustrating this.

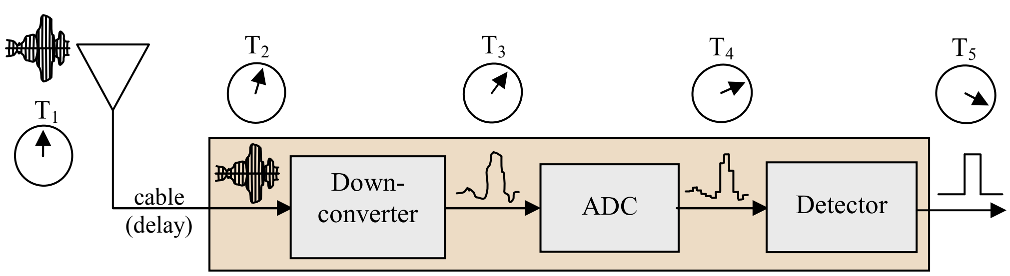


Figure 4.2-1: The Timing of a Signal at Various Points in a System.
Depending on the Reference Point chosen the time associated with the signal is different.

* *Reference points are explained in Sections 4.1.3.2.7, 7.1.5.3, and 8.1.7.*

In this simple example, a signal containing some information arrives at the antenna at time T_1. This signal travels through the cable to arrive at the downconverter at time T_2. Due to filtering and various electronic circuit delays the downconverted version of this signal emerges at time T_3 from the downconverter. The ADC has a conversion delay resulting in a digitized version of the downconverted signal at time T_4. Finally, the detector has some processing delay resulting in an indicator that the feature of interest was detected at time T_5.

In this example the Timestamp on the information detected in the signal might reasonably be any of the five times shown. If we specify the Reference Point to be the antenna however, then the proper Timestamp value will be T_1 because this is the time at that the information was present at the Reference Point. Throughout this specification the time at the Reference Point is known as the *Reference-Point Time*.

Definition 4.2-1: The Reference-Point Time of an event is the time that the information (in the signal) corresponding to the event was present at the Reference Point. The Reference-Point Time of a Data Sample in an IF Data packet is the time that the information represented by that Data Sample was present at the specified Reference Point.

It should be noted that a VRT-enabled system can have multiple Reference Points for time as well as for other types of Context. For example in a system that records Data from a received signal for later processing, both the time of reception and the time of processing might be important. VRT supports this.

5 Compliance

This specification applies the concept of compliance only to the Information Streams emitted/received by systems, and not to the systems themselves.* 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.

Packet Class documentation and Information Class documentation are cornerstones of this specification. 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. Section 8 sets forth rules for Information Class documentation, and Appendix A provides an example.

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

Documentation Rule 5-1: A VRT Information Class **shall** provide all of the defining details for the Information Stream, as required by this specification.

Observation 5-1: Since the VRT protocol relates only to the transport layer, a VRT emitter and receiver may be fully interoperable at that layer and yet not be truly interoperable due to differing link layers and/or physical layers.

Observation 5-2: When equipment is termed "VRT compliant," it simply means that it emits or receives some VRT Information Stream. A VRT compliant emitter and receiver are not interoperable at the transport layer unless the receiver can receive the particular Information Stream that the emitter emits.

---

* *It is understood that systems utilizing VRT may also emit and/or receive other protocols. These systems may have some operational modes that use VRT and other modes that do not. Also, some equipment may use VRT only when certain software is running. So the concept of equipment compliance is considered not to be useful.*

6 Data Packet Classes and Streams

This section sets forth the rules controlling Data Packet Classes and Data Packet Streams. As explained in Section 4, a Data Packet Class is the specification of a structure and function for a packet type that conveys some type of signal data. VRT defines two categories of Data Packet Classes: one for IF Data and one for Extension Data. Within each category any number of Data Packet Classes may be created. Each can have its own unique choice of parameters such as data type, data packing, timestamp type, etc.

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

Many applications involving IF Data also involve other kinds of Data. Such Data might be derived from IF Data. This is the case, for example, with a demodulated bit-stream from a PSK carrier in the IF Data. Non-IF Data may also take many other forms, so more flexibility is required 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 a wide variety of non-IF Data types. In order to accommodate this variety the rules for Extension Packet Classes and Streams are more relaxed than those applying to IF Data Packet Streams.

The following two rules are provided 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** only be used to convey IF Data, as defined in Section 4.

Rule 6-2: An Extension Data Packet Class **shall** only be used to convey data that is not IF Data.

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 Packet 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 template for every IF Data packet is shown in Figure 6.1-1. The packet is to be transmitted in big-endian order, i.e., most significant byte first[*]. 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 organized as shown in Figure 6.1-1. Packets **shall** be transmitted in big-endian byte order.

Rule 6.1-2: When 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: When 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 Class ID field might be specified as present in only every tenth packet.

[*] *This transmission order is consistent with other transport layers, such as TCP, UDP, and RTP. It may require software reordering of bytes or half-words on some systems in order to convert the packet to or from the format desirable for processing.*

<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">Header (1 Word, Mandatory)</td></tr>
<tr><td colspan="32">Stream Identifier (1 Word, Optional)</td></tr>
<tr><td colspan="32">Class Identifier (2 Words, Optional)</td></tr>
<tr><td colspan="32">Integer-seconds Timestamp (1 Word, Optional)</td></tr>
<tr><td colspan="32">Fractional-seconds Timestamp (2 Words, Optional)</td></tr>
<tr><td colspan="32">Data Payload (Variable, Mandatory)</td></tr>
<tr><td colspan="32">Trailer (1 Word, Optional)</td></tr>
</table>

Figure 6.1-1: The Template for IF Data Packet Classes

6.1.1 The IF Data Packet Header

Rule 6.1.1-1: Every VRT IF Data packet **shall** have a header with the format shown in Figure 6.1.1-1.

<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="4">Pk. Type
0 0 0 X</td><td>C</td><td>T</td><td colspan="2">R R</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="4">Packet Count</td><td colspan="16">Packet Size</td></tr>
</table>

Figure 6.1.1-1: The IF Data Packet Header

Rule 6.1.1-2: The IF Data packet header **shall** contain all the fields described in the remainder of this section.

The "Packet Type" field defines the type of VRT packet. Table 6.1.1-1 shows the types of VRT packets and corresponding packet type codes. As shown, valid IF Data Packet type 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 VRT 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 packet without Stream Identifier
0001 (1)	IF Data packet with Stream Identifier
0010 (2)	Extension Data packet without Stream Identifier
0011 (3)	Extension Data packet with Stream Identifier
0100 (4)	IF Context packet (see Section 7)
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

The "C" bit-field indicates whether the Class Identifier (Class ID) field is included in a packet. Each packet in an IF Data Packet Stream may or may not include the Class ID field.

Rule 6.1.1-4: The C bit **shall** be set to one in IF Data packets that include the Class ID field. It **shall** be set to zero in IF Data packets that do not include the Class ID field.

The "T" bit-field indicates whether the trailer is included in a packet. Each packet in an IF Data Packet Stream may or may not include the trailer.

Rule 6.1.1-5: The T bit **shall** be set to one in IF Data packets that include the trailer. It **shall** be set to zero in IF Data packets that do not include the trailer.

The "R" bits are reserved for future use in VRT.

Rule 6.1.1-6: The reserved bits **shall** be set to 0.

The "TSI" (TimeStamp-Integer) field is an encoded field indicating which, if any, type of Integer-seconds Timestamp is present in the packet. Table 6.1.1-2 shows the type of Timestamp corresponding to each valid TSI code. The meaning of each type of Integer-seconds Timestamp is given in Section 6.1.4.

TSI code	Meaning
00	No Integer-seconds Timestamp field included
01	Coordinated Universal Time (UTC)
10	GPS time
11	Other

Table 6.1.1-2: The Meaning of the TSI Codes

Rule 6.1.1-7: The TSI field in an IF Data packet **shall** accurately indicate the type of Integer-seconds Timestamp included the packet according to the code assignments in Table 6.1.1-2.

Rule 6.1.1-8: All the packets in an IF Data Packet Stream **shall** have the same TSI code, and thus the same Integer-seconds Timestamp.

The "TSF" (TimeStamp-Fractional) field is an encoded field indicating which, if any, type of Fractional-seconds Timestamp is present in the packet. Table 6.1.1-3 shows the type of Timestamp corresponding to each valid TSF code. The meaning of each type of Fractional-seconds Timestamp is given in Section 6.1.5.

Rule 6.1.1-9: The TSF field in an IF Data packet **shall** accurately indicate the type of Fractional-seconds Timestamp included the packet according to the code assignments in Table 6.1.1-3.

TSF code	Meaning
00	No Fractional-seconds Timestamp field included
01	Sample Count Timestamp
10	Real Time (Picoseconds) Timestamp
11	Free Running Count Timestamp

Table 6.1.1-3: The Meaning of the TSF Codes

Rule 6.1.1-10: All the packets in an IF Data Packet Stream **shall** have the same TSF code, and thus the same Fractional-seconds Timestamp.

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

Rule 6.1.1-11: 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 16-bit field indicates the total number of 32-bit words present in the IF Data packet, including the header, payload and any optional fields.

Rule 6.1.1-12: The Packet Size field within each packet **shall** correctly indicate the total number of 32-bit words in that packet.

Observation 6.1.1-1: Every IF Data packet consists of an integer number of 32-bit words less than 2^{16} (65,536). 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 within a Packet Stream.

6.1.2 The Stream Identifier

A Stream Identifier (Stream ID) is a 32-bit number assigned to a VRT Packet Stream. Each packet in a Packet Stream contains the Stream ID for that Packet Stream. This identifies all the packets in the Packet Stream as belonging to to it. In most cases each VRT Packet Stream will have a unique Stream ID, and the different Stream IDs used in different Packet Streams enable a receiver of those packets to separate them into their respective Packet Streams.

It is useful, but not necessary, to include the Stream ID in systems where only a single Packet Stream is emitted on each data link since in this case there is no chance a packet will be associated with the wrong Packet Stream. In general however, systems with multiple Packet Streams need to include the Stream ID in every Packet Stream. Also, since VRT enables the design of systems using equipment from multiple vendors, user selecability of Stream IDs is desired. This allows users of the equipment to ensure all the Stream IDs are unique within a system without coordinating with multiple vendors in advance.

Rule 6.1.2-1: The Stream ID **shall** be a 32-bit number assigned to a VRT Packet Stream. When used, the same Stream ID **shall** be carried in every packet in the Packet Stream.

Rule 6.1.2-2: The Stream ID Field **shall** be either included in, or omitted from, every IF Data packet in an IF Data Packet Stream according to the packet type as described in Table 6.1.1-1.

Recommendation 6.1.2-1: Every Data Packet Stream within a system that includes multiple VRT Packet Streams **should** have a unique Stream ID.

Recommendation 6.1.2-2: A system **should** be designed to allow modification of the Stream IDs in the emitted Packet Streams. This will allow for easy integration of multiple VRT-enabled systems in to one larger system where all the Stream IDs may need to be kept unique.

Rule 6.1.2-3: When Data-Context pairing is used to associate Context with an IF Data Packet Stream the IF Data packet format used **shall** include a Stream ID.

Observation 6.1.2-1: When Data-Context pairing is not in use[*], and Packet Streams are not multiplexed on a link or multiplexing is controlled at a lower level in the protocol stack, the Stream ID is not needed.[†]

---

[*] *See Section 4.1.3.2.8.1 for an overview of Data-Context pairing, and Section 7.1.2 for details.*

[†] *When, for example, a separate TCP/IP socket is opened for each VRT Packet Stream, the Stream ID field is not required because the sockets provide the Packet Stream multiplexing and demultiplexing capability.*

6.1.3 The Class Identifier

The Class Identifier (Class ID) field makes it possible for the receiver of a VRT Packet Stream to determine the identity of both the Information Class used for the application and the Packet Class from which each received packet was made. Figure 6.1.3-1 shows the contents of the Class ID field. This field contains three subfields. The first subfield contains the Organizationally Unique Identifier, or OUI [10]. This is an IEEE-assigned 24-bit number which indicates the identity of the company that created the Information Class and the Packet Class generating the IF Data Packet Stream. The second field contains the Information Class code. This Class Code indicates which of that company's Information Classes defines the Information Stream containing the Packet Stream. The third field contains the Packet Class code. This code identifies which of the company's Packet Classes was used to make the packet.

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	Reserved	OUI	
2	Information Class Code		Packet Class Code

Figure 6.1.3-1: The Contents of the Class ID Field.

Rule 6.1.3-1: When the Class ID field is present in an IF Data packet the OUI field **shall** contain the OUI of the company that created the Information Class and the Packet Class.

Observation 6.1.3-1: The Company that created the Information Class and Packet Classes used in an IF Data Packet Stream may or may not be the manufacturer of the equipment emitting the Packet Stream.

Rule 6.1.3-2: When the Class ID field is present in an IF Data packet the Information Class Code field **shall** contain an originator-assigned code indicating the Information Class defining the Information Stream to which the packet belongs. A value of zero in this field **shall** indicate that the Information Class is unspecified.

Rule 6.1.3-3: When the Class ID field is present in an IF Data packet the Packet Class Code field **shall** contain an originator-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.

Observation 6.1.3-2: Companies that define Information and Packet Classes are responsible for generating Information and Packet Class Codes that identify the corresponding classes without ambiguity. These codes need only be unique within the organization that created the codes however.

Observation 6.1.3.3: A system made by one Company may emit a VRT Information Stream defined by another Company. In this case the appropriate OUI for the Class ID field is the OUI of the Company that defined the Information Stream rather than the OUI of the Company that designed, built, or sold the equipment involved.

Rule 6.1.3-4: All of the reserved bits in the Class ID Field **shall** be set to zero.

6.1.4 The Integer-Seconds Timestamp

The Timestamp in an IF Data packet is divided into two components: an integer-seconds component and a fractional-seconds component. Together these two components precisely specify the Reference-Point Time of the first Data Sample contained in the packet.

The integer-seconds part specifies the Reference-Point Time only to one-second resolution, while the fractional-seconds part adds additional resolution. The *Integer-seconds Timestamp* consists of a single 32-bit word. It may be used to convey UTC time, GPS time, or some user-specified time-code. The type of time conveyed by this word is indicated by the value of the TSI bits.

Rule 6.1.4-1: When present, the Integer-seconds Timestamp in an IF Data packet **shall** represent the Reference-Point Time of the first Data Sample in the packet, in seconds, relative to the starting time. The starting time shall be either the starting time for UTC, the starting time for GPS, or "Other," as defined in the following subsections. 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 7.1.5.13

Rule 6.1.4-2: When present, the Integer-seconds Timestamp in an IF Data packet **shall** represent the Reference-Point Time to one-second resolution. .

Rule 6.1.4-3: When present, the Integer-seconds Timestamp **shall** consist of one 32-bit unsigned integer located at the position shown in Figure 6.1-1. The lsb of this 32-bit number **shall** be on the right.

6.1.4.1 Coordinated Universal Time (UTC)

Rule 6.1.4.1-1: When the Integer-seconds Timestamp conveys UTC it **shall** provide the Reference-Point Time of the first Data Sample in the packet in seconds, including leap seconds, since midnight January 1, 1970, Greenwich Mean Time [11].

6.1.4.2 GPS Time

Rule 6.1.4.2-1: When the Integer-seconds Timestamp conveys GPS time it **shall** provide the Reference-Point Time of the first Data Sample in the packet in seconds since midnight January 6, 1980, Greenwich Mean Time.

6.1.4.3 "Other" Time

IF Data packets may use other time references, standard or nonstandard. The use of non-standard time-bases is not expected to facilitate interoperability, but it is allowed.

Rule 6.1.4.3-1: When the Integer-seconds Timestamp conveys "Other" time it shall provide the Reference-Point Time of the first Data Sample in the packet in seconds since some documented starting time.

Documentation Rule 6.1.4.3-1: When an IF Data packet contains an Integer-seconds Timestamp of type "Other" the corresponding Packet Class documentation **shall** document the corresponding starting time.

6.1.5 The Fractional-Seconds Timestamp

The *Fractional-seconds Timestamp* conveys the Reference-Point Time to a higher resolution than does the Integer-seconds Timestamp. There are three types of Fractional-second Timestamps:

- The "Sample-Count" Timestamp
- The "Real-Time" Timestamp
- The "Free-Running Count" Timestamp.

The first two of these typically serve to add resolution to the Integer-seconds Timestamp, so that together they provide a range of years and a precision down to either the sample-period or one picosecond respectively. The third Fractional-seconds Timestamp, the Free-Running Count Timestamp, provides an incrementing sample count from any chosen starting time. It has no constant relationship to the Integer-seconds Timestamp. The three types of Fractional-seconds Timestamps are described in the following three subsections.

Each of the three Fractional-seconds Timestamps consists of an unsigned 64-bit integer which occupies two consecutive 32-bit words. Figure 6.1.5-1 shows the relative positions of these 32-bit words.

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	Most-significant (upper 32 bits)																															
2	Least-significant (lower 32 bits)																															

Figure 6.1.5-1: The Format of Fractional-Second Timestamps

Rule 6.1.5-1: The Fractional-seconds Timestamp **shall** be a 64-bit unsigned integer occupying two consecutive 32-bit words in the packet at the location specified in Figure 6.1-1. The most significant 32 bits **shall** be in the first of these two words. The lsb of each word **shall** be on the right.

Permission 6.1.5-1: The Fractional-seconds Timestamp **may** be used with or without the Integer-seconds Timestamp.

6.1.5.1 The Sample Count Timestamp

The Sample Count Timestamp extends the resolution of the Integer-seconds Timestamp down to one Data Sample period. It accomplishes this by conveying the sample number, as counted at the Reference Point, of the first Data Sample in the IF Data packet relative to the time of the last Integer-seconds Timestamp increment. Thus it is reset to zero at each increment of the Integer-seconds Timestamp. In the case where the Integer-seconds Timestamp is not in use, this timestamp still resets to zero when it reaches the number of Data Samples in one second. The timing of these events is not specified in this case however.

Rule 6.1.5.1-1: When the Integer-seconds Timestamp is also in use, the Sample-Count Timestamp in an IF Data packet **shall** convey the Reference-Point Time of the first Data Sample in the packet in sample counts relative to the time of the most recent increment of the Integer-seconds Timestamp.

Rule 6.1.5.1-2: When the Integer-seconds Timestamp is also in use, the Sample Count **shall** be reset to zero on each increment of the Integer-seconds Timestamp.

Permission 6.1.5.1-1: When the Integer-seconds Timestamp is not used, the Sample-Count **may** be set to zero at any chosen one-second intervals.

6.1.5.2 The Real-Time Timestamp

The Real-Time Timestamp extends the resolution of the Integer-seconds Timestamp down to one picosecond. It accomplishes this by conveying the Reference-Point Time of the first Data Sample in the IF Data packet, in picoseconds, relative to the time of the last Integer-seconds Timestamp increment. Thus it is reset to zero at each increment of the Integer-seconds Timestamp. In the case where the Integer-seconds Timestamp is not in use, this timestamp still resets to zero when it reaches the number of picoseconds in one second. The timing of these events is not specified in this case however.

Rule 6.1.5.2-1: When the Integer-seconds Timestamp is also in use, the Real-Time Timestamp in an IF Data packet **shall** convey the Reference-Point Time of the first Data Sample in the packet in picoseconds relative to the time of the most recent increment of the Integer-seconds Timestamp.

Rule 6.1.5.2-2: When the Integer-seconds Timestamp is also in use, the picoseconds count **shall** be reset to zero on each increment of the Integer-seconds Timestamp.

Permission 6.1.5.2-1: When the Integer-seconds Timestamp is not used, the Real-Time Timestamp **may** be set to zero at any chosen one-second intervals.

6.1.5.3 The Free Running Count Timestamp

The Free Running Count Timestamp conveys the Reference-Point Time of the first Data Sample in the IF Data packet, in sample counts, relative to any chosen starting time. 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 Integer-seconds Timestamp.

Rule 6.1.5.3-1: The Free Running Count Timestamp in an IF Data packet **shall** convey the Reference-Point Time of the first Data Sample in the packet in sample counts, modulo-N, relative to the counter starting time.

Rule 6.1.5.3-2: This Free Running Count **shall not** reset to zero except as when it rolls over modulo-N, where N may be any constant positive number up to 2^{64}.

Documentation Rule 6.1.5.3-1: When the Free Running Count Time-Code is used, either the Packet Class or the Information Class documentation **shall** specify the modulus, N, for the free running counter.

6.1.6 The Data Payload

The data payload of an IF Data packet contains a contiguous sequence of the Data Samples from an IF Data Sample stream. The number of words in the data payload is variable from packet to packet, and can be determined at the receiving end of the link from the Packet Size by subtracting the number of words dedicated to the header, trailer, and other additional fields. The presence or absence of these fields can be determined entirely from information in the header.

Rule 6.1.6-1: The maximum number of data payload words **shall** be 2^{16}-1 minus the number of words required for the header and any optional fields (i.e. Stream ID, Class ID, Timestamps, and trailer).

Rule 6.1.6-2: The data payload **shall** consist of an integer number of contiguous 32-bit words.

Rule 6.1.6-3: When the number of payload bits to pack is not an integer multiple of 32 bits, the payload size, in bits, **shall** be rounded up to the nearest multiple of 32 bits.

Permission 6.1.6-1: 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 6.1.6-4: Payloads in IF Data packets **shall** be packed in accordance with the rules set forth in the following sections.

Documentation Rule 6.1.6-1: The Packet Class documentation **shall** specify how the data in the data payload is packed.

6.1.6.1 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 defines a type of payload field called an Item Packing Field. An Item Packing Field is a virtual container for up to three types of information: a Data Item, Event Tags, and a Channel Tag. The three types of information are defined below. When included in an Item Packing Field, the subfields are always packed in a certain order, as shown in Figure 6.1.6.1-1. Several examplesof Item Packing fields are given in Appendix B.9.

Definition 6.1.6.1-1: An Item Packing Field is a virtual container for a Data Item, Event Tags, and a Channel Tag.

Definition 6.1.6.1-2: A Data Item is a binary number representing all or part of a Data Sample. 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 complex polar sample.

Definition 6.1.6.1-3: 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 6.1.6.1-4: A Channel Tag is a label associating a Data Item with a particular signal conveyed by an IF Data packet. It supports demultiplexing of transported signals by the receiver of the Packet Stream when more than one signal is conveyed in each packet.

Figure 6.1.6.1-1 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.

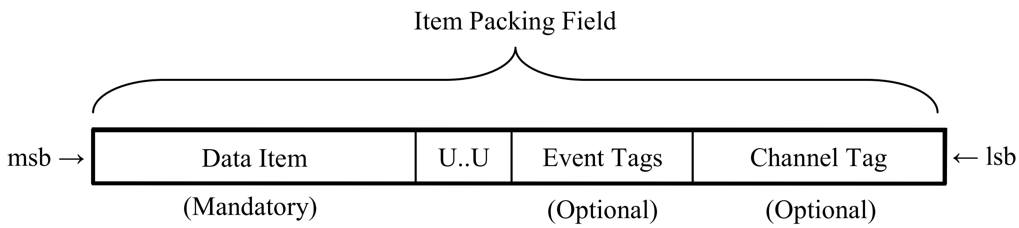


Figure 6.1.6.1-1: Relative Locations of Items within the Item Packing Field
(Bits labeled "U" are unused)

Rule 6.1.6.1-1: 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 6.1.6.1-2: 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 6.1.6.1-1: Rule 6.1.6.1-2 applies even when complex polar 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 6.1.6.1-3: An Item Packing Field **shall** contain either a single Channel Tag or none. 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 6.1.6.1-4: An Item Packing Field **shall** contain either a single Event Tag field or none. 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 6.1.6.1-1: The precise meanings of the Event Tag bits are not specified. However, the encodings **should** be chosen such that the Event Tag field contains all zeros when no events have occurred.

Documentation Rule 6.1.6.1-1: When Event Tag bits are present the Packet Class or Information Class documentation **shall** specify their meaning and format.

Rule 6.1.6.1-5: When an event occurs and is indicated by some bit pattern in an Event Tag field, the Item Packing Field containing this bit pattern **shall** be the one containing the Data Item whose Reference-Point Time most closely matches the Reference-Point Time of the event.

Rule 6.1.6.1-6: 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.

Rule 6.1.6.1-7: The size of the Item Packing Field **may** be larger than required to contain the included items. In this case the unused bits **shall** reside immediately to the right of the Data Item.

Observation 6.1.6.1-2: 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.

Documentation Rule 6.1.6.1-2: The Packet Class documentation **shall** specify the Item Packing Field size, the Data Item size, the Event Tag field size, and the Channel Tag size.

6.1.6.2 The Arrangement of Item Packing Fields in 32 Bit Words

Item Packing Fields may be packed into 32-bit words in the payload section according to either of two methods. The first method is called "link-efficient" packing, and the second is called "processing-efficient" packing. Figure 6.1.6.2-1 shows an example of each for an Item Packing Field size of 15 bits.

Rule 6.1.6.2-1: Item Packing Fields **shall** be arranged in the 32-bit wide payload region either according to link-efficient rules or according to processing-efficient rules. All payloads in an IF Data Packet Stream **shall** be packed identically.

Rule 6.1.6.2-2: When link-efficient packing is implemented all bits in each 32-bit word **shall** be used for the packing of Item Packing Fields. The only exception is 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 **shall** "wrap" from one 32-bit word to the next.

Rule 6.1.6.2-3: When processing-efficient packing is implemented, the number of Item Packing Fields per 32-bit word **shall** be the maximum number that fit into a 32-bit word. When 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. For processing-efficient packing every 32-bit word **shall** be packed identically.

Documentation Rule 6.1.6.2-1: The Packet Class documentation **shall** specify whether processing-efficient or link-efficient packing is used.

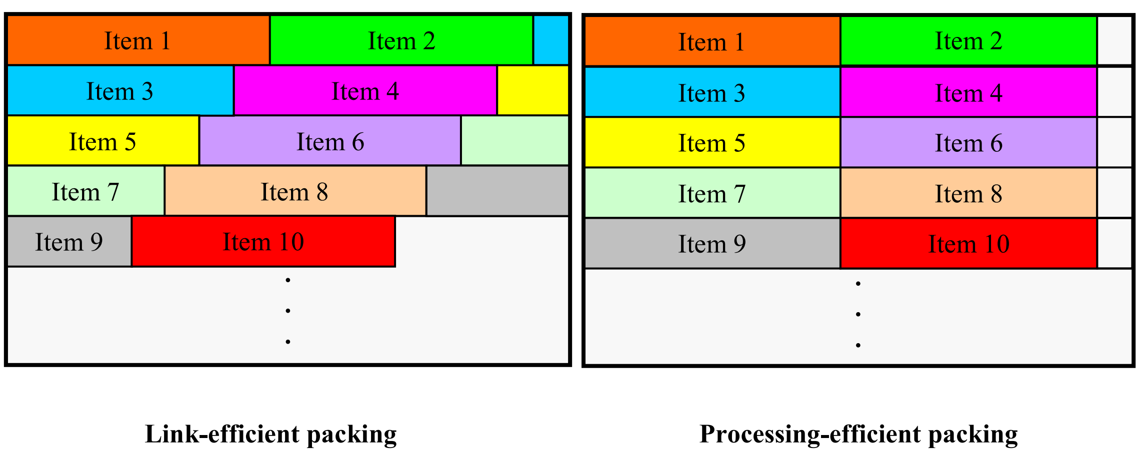


Figure 6.1.6.2-1: Fifteen-bit Item Packing Fields Packed into 32-bit Words.
For link-efficient packing the leftover bits at the right of each word hold the most-significant bits of an Item Packing Field, and the rest of the field is packed into the left-most bits of the next 32-bit word. For processing-efficient packing the rightmost bits are not used.

6.1.6.3 The 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. For example when the Data Samples are complex Cartesian, half of the Data Items are "I" data and the other half are "Q" data. As another example, a real sample vector consists of multiple identical

Data Items, with each corresponding to a different component of the vector. 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:

- The Item Packing Field occupying the left-most bits in the first 32-bit word of the payload is the "first" Item Packing Field.
- Within a 32-bit word, an Item Packing Field to the left is always numbered lower than one to the right.
- An Item Packing Field in an earlier 32-bit word is always numbered lower than one in a later 32-bit word.
- 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 6.1.6.3-1: This section gives rules for ordering Data Items within sequential Item Packing Fields. These rules **shall** be observed independent of whether link-efficient or processing-efficient packing is used.

Definition 6.1.6.3-1: A Channel is an independent Data Sample stream within an IF Data Packet Stream.

Definition 6.1.6.3-2: 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.

Observation 6.1.6.3-1: In the case of multi-dimensional Data streams each Data 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 component of 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 6.1.6.3-3: Sample-Component Repeating is a method of arranging the real and imaginary components (or amplitude and phase components) of a set of complex Data Samples, such that all of the real (amplitude) components in the set are packed first followed by the associated imaginary (phase) components. The number of sequential components of the same type is referred to as the "Repeat Count."

Definition 6.1.6.3-4: Channel Repeating is a method of arranging Data Items such that some number of Data Samples for each Channel is packed sequentially prior to packing Data Samples from the next Channel. The number of consecutive Data Samples per Channel to be packed is referred to as the "Repeat Count."

Observation 6.1.6.3-2: Channel Repeating is only useful in conjunction with Sample Vectors. It has no value when multiple asynchronous Channels are carried in the Packet Stream since Channel Tags are required and they enable any kind of Channel arrangement. Channel Repeating has no meaning when only one Channel is carried by the Packet Stream.

Documentation Rule 6.1.6.3-1: The Packet Class documentation **shall** specify the following parameters affecting the ordering of Data Items in sequential Item Packing Fields:

1. Data Sample type: real, complex Cartesian, or complex polar.
2. Whether Sample-Component Repeating is in use.
3. Whether Channel Repeating is in use.
4. The Repeat Count value for Sample-Component Repeating or Channel Repeating.
5. The width of the Sample Vectors.

Rule 6.1.6.3-2: Earlier (i.e. corresponding to lower time values) Data Items **shall** always precede later Data Items in sequential Item Packing Fields.

Rule 6.1.6.3-3: The in-phase (I) component of a Cartesian complex Data 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 6.1.6.3-4: The amplitude component of a complex polar Data Sample **shall** always be packed 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 6.1.6.3-5: Sample-Component Repeating **shall** only be used with complex Data Samples.

Rule 6.1.6.3-6: When Sample-Component Repeating is used with complex Cartesian Data 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 6.1.6.3-7: When Sample-Component Repeating is used with complex polar Data 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 6.1.6.3-8: When Channel Repeating is used the number of Data 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 from one to 65,536 with a value of one indicating that Channel Repeating is not in use.

Rule 6.1.6.3-9: Sample-Component Repeating and Channel Repeating are mutually-exclusive features. They **shall not** be used together in the same IF Data Packet Stream.

Rule 6.1.6.3-10: Sample Vector size **may** range from one to 65,535. A value of zero or one **shall** indicate that Sample Vectors are not in use.

6.1.6.4 Payload Data Sample and Data Item Formats

This VRT provides a number of Data Sample and Data Item format options, which are described in this section.

Rule 6.1.6.4-1: An IF Data Packet Class **shall** use one of the following Data Sample formats:

1. Real samples
2. Complex Cartesian samples (i.e. in-phase and quadrature components)
3. Complex polar samples (i.e. amplitude and phase components)

Rule 6.1.6.4-2: An IF Data Packet Class **shall** use one of the following Data Item formats:

1. Fixed-point unsigned
2. Fixed-point signed
3. VRT unsigned floating-point*
4. VRT signed floating-point
5. IEEE 754 single-precision (32-bit) floating-point
6. IEEE 754 double-precision (64-bit) floating-point

Documentation Rule 6.1.6.4-1: The Packet Class documentation for an IF Data Packet Class **shall** specify the Data Sample format and Data Item formats.

* *"VRT" signed and unsigned floating-point numbers are explained in greater detail in Appendix D.*

Rule 6.1.6.4-3: All Data Items in all the packets in an IF Data Packet Stream **shall** use the same format. This **shall** apply whether the Data Samples are real, complex Cartesian, or complex polar.

Observation 6.1.6.4-1: In real and complex Cartesian Data Samples both the format and the interpretation of all Data Items is identical. In complex polar Data Samples however, Data Items representing phase must use the same format as the amplitude Data Items, but their interpretation is different since they represent phase rather than amplitude.

Throughout this section, and the rest of this specification, when the numeric range of a number, other than an IEEE floating-point number[*], is discussed, that range is based on the *Normalized Interpretation* of the number.

Definition 6.1.6.4-1: The <u>Normalized Interpretation</u> of the numeric range of a fixed-point or VRT floating-point number is the interpretation that the maximum positive range of the number is one step less than one. Specifically:

- For unsigned fixed-point N-bit numbers the numeric range is zero to $1\text{-}2^{-N}$ inclusive.
- For unsigned VRT floating-point numbers with an M-bit mantissa the numeric range is zero to $1\text{-}2^{-M}$ inclusive. (See Appendix D for more on VRT numbers.)
- For signed fixed-point N-bit numbers the numeric range is –1 to $1\text{-}2^{-(N-1)}$.
- For signed VRT floating-point numbers with an M-bit mantissa the numeric range is -1 to $1\text{-}2^{-(M-1)}$ inclusive.

The following rules related to non-IEEE number formats are all based on the Normalized Interpretation of the numeric range.

Rule 6.1.6.4-4: The fixed-point unsigned Data Item size **may** be any size, N, from 1 bit to 64 bits. For non-phase Data Items this **shall** represent a fractional number in range of zero to $1\text{-}2^{-N}$ inclusive.

Rule 6.1.6.4-5: Unsigned N-bit phase components **shall** have a range of zero to $[(1\text{-}2^{-N}) \cdot 2\pi]$ radians. The N-bit binary value "0…0" shall correspond to a phase of zero radians and the N-bit value "1…1" shall correspond to a phase of $[(1\text{-}2^{-N}) \cdot 2\pi]$ radians.

Rule 6.1.6.4-6: The fixed-point signed Data Item size **may** be any size, N, from 1 bit to 64 bits. The format **shall** be two's-complement. For non-phase Data Items, this **shall** represent a fractional number in range of –1 to $1\text{-}2^{-(N-1)}$ inclusive.

Rule 6.1.6.4-7: Signed N-bit phase components **shall** be two's-complement numbers with a range of $-\pi$ to $[(1\text{-}2^{-(N-1)}) \cdot \pi]$ radians. The N-bit binary value "10…0" shall correspond to a phase of $-\pi$ radians and the N-bit binary value "01…1" shall correspond to a phase of $[(1\text{-}2^{-(N-1)}) \cdot \pi]$ radians.

Rule 6.1.6.4-8: 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 6.1.6.4-1. The mantissa **shall** be of a size, M, from 1 to 63 bits. The exponent **shall** be of a size, E, from 1 to 6 bits such that $M+E = N$. 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. For non-phase Data Items the numeric range of this equivalent fixed-point number **shall** be 0 to $1\text{-}2^{-M}$ inclusive.

msb	msb-1	msb-2	******	lsb+2	lsb+1	lsb	msb	***	lsb
Mantissa (unsigned)							Exponent		

Figure 6.1.6.4-1: VRT Unsigned Floating Point Number Format

[*] *IEEE 754 floating-point numbers are interpreted by that specification.*

Rule 6.1.6.4-9: When used for the phase component of a complex polar Data Sample, the unsigned VRT Data Item **shall** have a range from 0 to [(1-2^{-M}) · 2π] radians inclusive.

Rule 6.1.6.4-10: 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 6.1.6.4-2. The mantissa **shall** be a signed two's complement 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** 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. For non-phase Data Items the range of this equivalent signed fixed-point number **shall** be -1 to 1-$2^{-(M-1)}$ inclusive.

<table>
<tr><td>msb</td><td>msb-1</td><td>msb-2</td><td>* * * * * *</td><td>lsb+2</td><td>lsb+1</td><td>lsb</td><td>msb</td><td>* * *</td><td>lsb</td></tr>
<tr><td colspan="7">Mantissa (two's compliment)</td><td colspan="3">Exponent</td></tr>
</table>

Figure 6.1.6.4-2: VRT Signed Floating Point Number Format

Rule 6.1.6.4-11: When used for the phase component of a complex polar Data Sample, the signed VRT Data Item **shall** have a range from - π to [(1-$2^{-(M-1)}$) · π] radians inclusive.

Observation 6.1.6.4-2: VRT numbers can be tailored to provide just the right combination of precision and dynamic range for an application. In many cases this can reduce the total number of bits required, and thus reduce link layer bandwidth compared to either fixed-point or IEEE 754 floating-point numbers. Appendix D explains the uses and operation of VRT floating-point numbers in more detail.

Rule 6.1.6.4-12: IEEE floating-point Data Items **shall** be either 32 bits or 64 bits in size. Their format **shall** be as specified in IEEE 754 [9]. The numeric range of the IEEE floating-point Data Items **shall** be as specified in IEEE 754.

Rule 6.1.6.4-13: When used for the phase component of a complex polar Data Sample, the IEEE floating-point number shall be interpreted in units of radians.

Documentation Rule 6.1.6.4-2: The Information Class documentation **shall** specify the physical range corresponding to the numeric range of the Data Item format used in the IF Data Packet Stream. This specification **shall** consist of a statement of the "Reference Level," as described in Section 7.1.5.9.

6.1.6.5 Item Packing Structure and Payload Size

Definition 6.1.6.5-1: An Item Packing 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 Item Packing Structures.

Observation 6.1.6.5-1: The combination of the various parameters affecting Data Item arrangement in the payload provides for a variety of different Item Packing 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 Data Samples are complex, 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 6.1.6.5-1: The number of 32-bit words in the data payload field **shall** be the smallest number needed to contain the desired (integer) number of Item Packing Structures. The size of the Item Packing Structure, i.e. the number of Item Packing Fields required to contain it, **shall** be calculated as the number of Data Items in a Data Sample multiplied by the vector size multiplied by the Repeat Count.

6.1.7 The Trailer

The IF Data packet trailer is an optional field whose presence is identified by the “T” bit in the header, as previously described. The trailer contains fields that indicate the validity of the Data and the status of the processes producing that Data. It also contains a field that indicates whether related Context is being sent in one or more separate “Context packets.” (Context packets are explained in Section 7.) The IF Data packet trailer contains four fields, as shown in Figure 6.1.7-1.

<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="12">Enables</td><td colspan="12">State and Event Indicators</td><td>E</td><td colspan="7">Associated Context Packet Count</td></tr>
</table>

Figure 6.1.7-1: The IF Data Packet Trailer

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 VRT emitter to a VRT receiver. 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 6.1.7-1 shows the bit positions and the functions of each bit in these fields.

Rule 6.1.7-1: State and Event Indicators and Enable bits **shall** be positioned as indicated in Table 6.1.7-1.

Rule 6.1.7-2: For each Indicator bit there is a corresponding Enable bit. The corresponding Enable bit for each Indicator bit **shall** be the one that is in the same position in the Enables field as the Indicator bit is in the State and Event Indicators field.

Rule 6.1.7-3: For the predefined indicators, when an Enable bit is set to one the corresponding indicator **shall** function as shown in Table 6.1.7-1. Otherwise, the corresponding indicator **shall** not be considered valid.

Permission 6.1.7-1: The user-defined indicators **may** be used for any purpose. They **may** be used together as well as individually.

Documentation Rule 6.1.7-1: The Packet Class documentation **shall** specify all “user-defined” indicators in the trailer along with the meaning of the binary codes.

Rule 6.1.7-4: 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 the indicator group is enabled. As with the individual Enables, a one **shall** indicate that the indicator group is enabled.

[*] *Events and state changes can be conveyed with more precise timing information using Context Packets, as described in Section 7.1.5.16.*

Enable Bit Position	Indicator Bit Position	Indicator Name
31	19	Calibrated Time Indicator
30	18	Valid Data Indicator
29	17	Reference Lock Indicator
28	16	AGC/MGC Indicator
27	15	Detected Signal Indicator
26	14	Spectral Inversion Indicator
25	13	Over-range Indicator
24	12	Sample Loss Indicator
[23..20]	[11..8]	User-Defined Indicators

Table 6.1.7-1: Indicator Bits and Enable bits
Each Indicator functions as indicated, but only when the corresponding Enable bit is set otherwise, the Indicator bit is undefined.

Rule 6.1.7-5: 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 it **shall** indicate that the Timestamp is free running and may be inaccurate.

Rule 6.1.7-6: The Valid Data Indicator, when set to one, **shall** indicate that the Data in the packet is valid. When set to zero it **shall** indicate that some condition exists that may invalidate the Data.

Documentation Rule 6.1.7-2: The Packet Class Documentation or the Information Class documentation **shall** specify the conditions impacting the Valid Data Indicator unless it is unused.

Observation 6.1.7-1: The meaning 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 later set to one when the frequency change is complete.

Rule 6.1.7-7: The Reference Lock Indicator, when set to one, **shall** indicate that any phase-locked loops affecting the Data are locked and stable. When set to zero it **shall** indicate that at least one PLL is not locked and stable.

Rule 6.1.7-8: The AGC/MGC Indicator, when set to one, **shall** indicate that AGC is active. When set to zero, it **shall** indicate MGC.

Rule 6.1.7-9: The Detected Signal Indicator, when set to one, **shall** indicate that the data contained in the packet contains some detected signal.

Documentation Rule 6.1.7-3: The Packet Class documentation or Information Class documentation **shall** specify what type of signal or signal feature is being detected.

Rule 6.1.7-10: The Spectral Inversion Indicator, when set to one, **shall** indicate that the signal conveyed in the data payload has an inverted spectrum with respect to the spectrum of the signal at the system Reference Point.[*]

Rule 6.1.7-11: The Over-range Indicator, when set to one, **shall** indicate that at least one Data Sample in the payload is invalid due to the signal exceeding the range of the Data Item.

[*] *The "Reference Point" is a point upstream in the processing chain. See Section 4 for the precise definition and Appendix B for a Reference Point example.*

Rule 6.1.7-12: The Sample Loss Indicator, when set to one, **shall** indicate that the packet contains at least one sample discontinuity due to processing errors and/or buffer overflow.

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 an indication 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 in the State and Event Indicators in the trailer, or may be of other types.[*]

The first, one-bit, field is the "Associated Context Packet Count Enable" bit. This bit is labeled "E" in Figure 6.1.7-1. The other field is the "Associated Context Packet Count" field.

Rule 6.1.7-13: 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. When the "E" bit is cleared, the "Associated Context Packet Count" is undefined.

Documentation Rule 6.1.7-4: When the Associated Context Packet Count accounts for only a subset of the associated Context packets, the Information Class documentation **shall** specify that subset.

Observation 6.1.7-2: 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 6.1.7-3: 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 6.1.7-14: 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 6.1.7-2: The "Associated Context Packet Count" **may** include associated Context packets transmitted by a process other than the one that transmits the IF Data packet containing the count.

Recommendation 6.1.7-1: All unused trailer bits **should** be set to zero.

Recommendation 6.1.7-2: 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. This helps avoid processing backlogs downstream.

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 and FFT data. An example of an Extension Data packet tailored to convey FFT data is shown in Appendix C.

The packet template for Extension Data packets is shown in Figure 6.2-1. It has a structure that is very similar to the IF Data packet structure. It contains a mandatory header, an optional Stream ID, an optional Class ID, a mandatory Data Payload, and an optional trailer. The only difference is in the payload, which is less constrained than that of an IF Data packet. The Packet Class documentation provides the specifications for the custom data payload format used.

[*] *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 the specific values of changing parameters. The capabilities of Context packets are discussed in Section 7.*

Rule 6.2-1: Every Extension Data packet **shall** be organized the same as an IF Data packet except that in the Extension Data packet the data payload is user-defined, as shown in Figure 6.2-1.

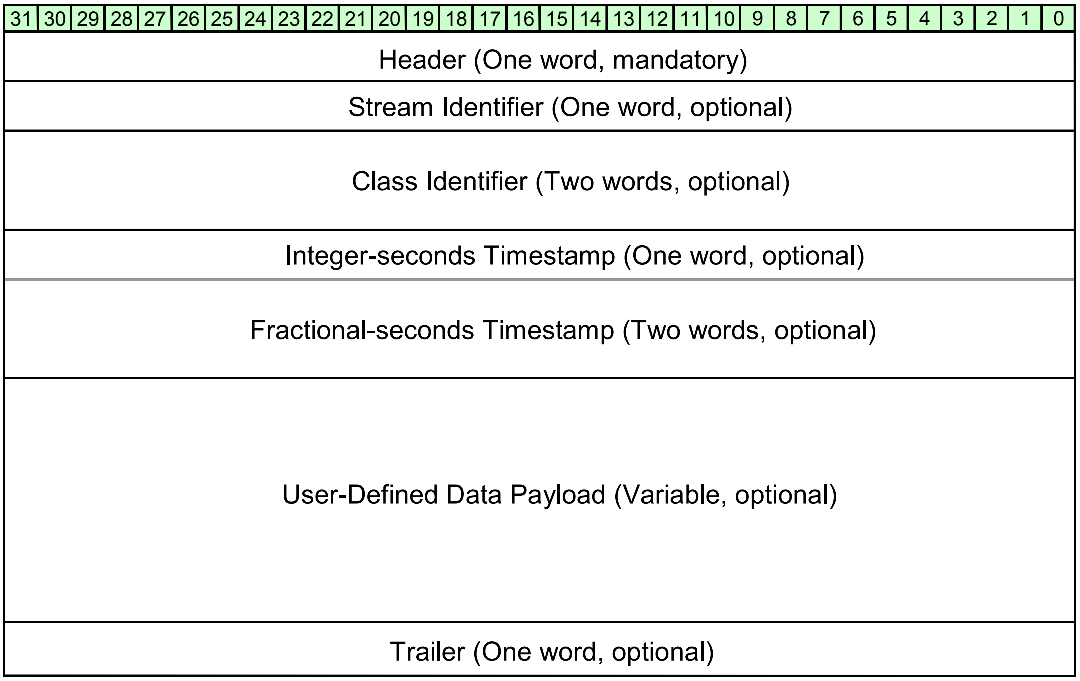


Figure 6.2-1: The Extension Data Packet Template

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 header **shall** include the Packet Type, C, T, TSI, TSF, Packet Count, 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 "0011" when a Stream ID is included or "0010" when it is not, as shown in Table 6.1.1-1.

Rule 6.2-5: The Stream ID field, when present in an Extension Data Packet Stream, **shall** function in the same way as in an IF Data Packet Stream.

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

Rule 6.2-7: The Integer-seconds Timestamp, when present in an Extension Data Packet Stream, **shall** use one of the options given in Section 6.1.4.

Rule 6.2-8: The Fractional-seconds Timestamp, when present in an Extension Data Packet Stream, **shall** use one of the options given in Section 6.1.5.

Observation 6.2-1: Although the Timestamp format used in an Extension Data packet must be one of the allowed formats, in the case of Extension Data its meaning is open to interpretation since the data may be non-time-domain data. For example, if the payload contains FFT Data then the first value in the payload would correspond to a frequency rather than to a time.

Documentation Rule 6.2-1: The Extension Data Packet Class documentation **shall** specify the interpretation of the Timestamp used.

Rule 6.2-9: Extension Data packet payload **shall** be an integer number of 32-bit words in size.

Permission 6.2-1: Extension Data packet payload **may** be structured in any way that does not violate Rule 6.2-9 and does not cause the packet to exceed the maximum size of 65,535 32-bit words.

Recommendation 6.2-1: An Extension Data Packet payload **should** be formatted to look as much like an IF Data Packet payload as possible, to facilitate the reuse of hardware and/or software.

Documentation Rule 6.2-2: The Extension Data Packet Class documentation **shall** describe the format of the payload section of the Extension Data packet, along with the interpretation of each field within it.

Rule 6.2-10: The trailer, when present in an Extension Data Packet Stream, **shall** function in the same way as in an IF Data packet.

Documentation Rule 6.2-3: The Extension Data Packet Class documentation **shall** specify the options chosen for the Stream ID, Class ID, Timestamps, and trailer.

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 description of the function of 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 types of Context Packet Classes: IF Context Classes and Extension Context Classes. From each type of class, 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 described in Section 6 are the means to convey discrete time sampled signal data. They are the backbone of the VRT specification, while the IF Context Packet Streams described in this section are the primary means of communicating metadata concerning these IF Data Packet Streams. IF Context Packet Streams can also communicate metadata about analog signals and receiver settings. 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.

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

Extension Context Packet Classes (see Section 7.2) are provided for cases when it is necessary to communicate metadata that can't be communicated in the IF Context Packet Class,. Extension Context Packet Classes can be 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 are typically transmitted as a result of a change in one of the states or parameters communicable by that Context Packet Class. Context packets are also typically transmitted at some periodic interval to ensure that a VRT receiver can recover from any miscommunication or loss of communication. Such periodic updates typically involve retransmitting all fields required by the application. 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.

The state of the sensor might be conveyed entirely by a single Context packet or by the combination of multiple Context packets. 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 dictate when Context packets are emitted and what they contain.

7.1 *IF Context Packet Classes and Streams*

This section sets forth rules controlling the structure and function of IF Context packets specified by an IF Context Packet Class. 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 IF Context packet **shall** be formatted as shown in Figure 7.1-1.

Rule 7.1-2: When 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: When 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.

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
Header (1 word, mandatory)																															
Stream Identifier (1 word, mandatory)																															
Class Identifier (2 words, optional)																															
Integer-seconds Timestamp (1 word, optional)																															
Fractional-seconds Timestamp (2 words, optional)																															
Context Section (variable length, mandatory)																															

Figure 7.1-1: The Template for IF Context Packet Classes

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 Stream Identifier, or Stream ID, field is the second 32-bit word in the IF Context packet. It is also mandatory. It indicates the Packet Stream to which the packet belongs. Stream IDs are used to associate different Packet Streams with an Information Stream as described in Section 7.1.5.25.

The Class Identifier, or Class ID, field is a two-word field that identifies the Information Class and Packet Class to which the Packet Stream belongs. It is optional and its presence or absence is indicated by the 'C' bit field in the header.

The Timestamp fields indicate the time of the Context update. They typically serve to correlate Context changes conveyed in the IF Context packet with specific Data Samples in the associated Data Packet Stream. The Integer-seconds and Fractional-seconds Timestamp fields are each optional and their inclusion or exclusion is indicated by the values in the TSI and TSF fields in the header, respectively.

The Context Section contains the metadata. It is mandatory.

7.1.1 The IF Context Packet Header

The header of the IF Context packet is the same as the header of the IF Data packet with the exception of the omission of the T 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 0 1 0 0				C	R	R	TSM	TSI		TSF		Packet Count				Packet Size															

Figure 7.1.1-1: IF Context Packet Header

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

Observation 7.1.1-1: The C, TSI, TSF, Packet Count, and Packet Size fields in IF Context Packets function the same way as for IF Data Packets.

Rule 7.1.1-2: In an IF Context packet, the Packet Type field **shall** contain the binary value 0100 as shown in Table 6.1.1-1.

Rule 7.1.1-3: The C bit **shall** be set to one in IF Context packets that include the Class ID field. It **shall** be set to zero in IF Context packets that do not include the Class ID field.

The Timestamp Mode (TSM) bit is used to indicate whether the Timestamp in the IF Context packet is being used to convey timing of Context events related to the Described Signal with fine or coarse resolution. The interpretation of the Timestamp for each of these modes is explained in Section 7.1.4.

Definition 7.1.1-1: The <u>Data Sampling Interval</u> of a Data packet is is the time from the first Data Sample to the last Data Sample of that Data packet.

Rule 7.1.1-4: The TSM bit **shall** be set to zero when the IF Context packet Timestamp is being used to convey the *precise* timing of events or Context changes. The resolution of this Timestamp Mode could be up to highest resolution supported by the TSF setting (either sample or picosecond resolution).

Rule 7.1.1-5: The TSM bit **shall** be set to one when the IF Context packet Timestamp is being used to convey the *general* timing of events or Context changes. The resolution of this Timestamp Mode is the Data Sampling Interval of a Data packet.

Rule 7.1.1-6: The TSM bit **shall not** be set to one in Context Packet Streams for which the Described Signal is not a Data Packet Stream.

Rule 7.1.1-7: The TSI field in an IF Context packet **shall** accurately indicate the type of Integer-seconds Timestamp included the packet according to the code assignments in Table 6.1.1-2.

Rule 7.1.1-8: All the packets in an IF Context Packet Stream **shall** have the same TSI code.

Rule 7.1.1-9: The TSF field in an IF Context packet **shall** accurately indicate the type of Fractional-seconds Timestamp included the packet according to the code assignments in Table 6.1.1-3.

Rule 7.1.1-10: All the packets in an IF Context Packet Stream **shall** have the same TSF code.

Rule 7.1.1-11: The Packet Count **shall** increment in each consecutive IF Context packet having the same Stream ID and the same packet type. The count **shall** roll over from “1111” to “0000.”

Rule 7.1.1-12: In each packet the Packet Size field **shall** correctly indicate the total number of 32-bit words in the packet.

Rule 7.1.1-13: When an IF Context Packet Stream is paired with a Data Packet Stream, the TSI and TSF fields for that IF Context Packet Stream **shall** match the TSI and TSF bits for the paired Data Packet Stream (i.e. they use the same type of Timestamp as described in Section 6.1.1).

7.1.2 The Stream Identifier

Stream Identifiers (Stream IDs) in the IF Context Packet Class are used in the same fashion as in IF Data Packet Classes (see Section 6.1.2) to identify particular packets as belonging to certain Packet Streams. In addition, the Stream ID in a Context Packet Stream may serve to pair it with a Data Packet Stream or to associate it with another Context Packet Stream. See Section 4.1.3.2.8 for a discussion of Data-Context pairing and Context Association.

Rule 7.1.2-1: The Stream ID **shall** be a 32-bit number assigned to a VRT Packet Stream. Every packet in the Packet Stream **shall** use the same Stream ID.

Rule 7.1.2-2: Every IF Context packet **shall** contain a Stream ID.

Recommendation 7.1.2-1: Every Context Packet Stream within a system that includes multiple VRT Packet Streams **should** have a unique Stream ID.

Recommendation 7.1.2-2: A system **should** be designed to allow modification of the Stream IDs in the emitted Packet Streams. This will allow for easy integration of multiple VRT-enabled systems in to one larger system where all the Stream IDs may need to be kept unique.

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 in Information Streams.

Rule 7.1.2-3: When an IF Context Packet Stream is paired with a Data Packet Stream the Stream ID of the IF Context Packet Stream **shall** match that of the Data Packet Stream.

Observation 7.1.2-2: Though every IF Context Packet Stream must contain a Stream ID, it is not necessary that it have a paired VRT Data Packet Stream. For example, an IF Context Packet Stream conveying geolocation information about a sensor may not have a paired Data Packet Stream. Regardless, this IF Context Packet Stream will contain a Stream ID.

Appendices A and B provide many examples on the use of Stream IDs.

7.1.3 The Class Identifier

The Class Identifier (Class ID) is used to identify the Information Class and Packet Class to which the IF Context packet belongs. It contains an OUI subfield which specifies the organization that specified the Information Class and Packet Class to which the IF Context Packet Stream belongs. It also contains codes that uniquely identify the Information Class and Packet Class from that organization.

Rule 7.1.3-1: The Class Identifier **shall** use the format shown in Figure 7.1.3-1.

Word	31–24	23–16	15–0
1	Reserved	Class Specifier OUI	
2	Information Class Code		Packet Class Code

Figure 7.1.3-1: Class Identifier

Rule 7.1.3-2: The Class ID OUI subfield **shall** contain the 24-bit IEEE-registered Organizationally Unique Identifier (company identifier) of the organization which specified the Information and Packet Classes to which the IF Context packet belongs.

Rule7.1.3-3: The Information Class Code subfield **shall** contain a 16-bit number identifying the Information Class to which this IF Context Packet Stream belongs. For each organization, the Information Class Code **shall** be unique for each Information Class used.

Rule 7.1.3-4: The Packet Class Code subfield **shall** contain a 16-bit number identifying the Packet Class to which this IF Context Packet Stream belongs. For each organization, the Packet Class Code **shall** be unique for each Information Class used.

Recommendation 7.1.3-1: Each organization **should** maintain a complete list of all Information Class Codes and Packet Class Codes used.

7.1.4 The Timestamp Fields

The options and formats for IF Context packet Timestamp fields are identical to those in an IF Data packet. However, the interpretation of the Timestamp in the Context Packet is dependent on the TSM bit in the header. When the Timestamp Mode is set to fine resolution (the TSM bit is set to zero) the Context packet Timestamp conveys the *precise* timing of events related to the Described Signal. This timing may be as precise as the resolution of the Fractional-Seconds Timestamp supports, i.e. either sample-period or picosecond resolution. It may also be less precise than this if the timing of the event is not known to this level of precision.

When the TSM bit is set to coarse resolution (the TSM bit is set to one) the Context packet Timestamp conveys *general* timing of events related to the Described Signal. That is, the Context packet conveys events that occurred sometime within the Data Sampling Interval of some Data packet in the paired Data Packet Stream.[*] To identify which Data Sampling Interval, the Context packet Timestamp must match the Timestamp of that Data packet in the paired Data Packet Stream. The less precise timing indication available with this TSM mode might be used because more precise information about the time of an event is not available, or not necessary, or because multiple events that occurred at different times within the Data packet sampling interval are grouped into a single Context packet for the sake of simplicity or efficiency.

Rule 7.1.4-1: When the Timestamp Mode is set to fine resolution (the TSM bit is set to zero) the Timestamp fields in an IF Context packet, when present, **shall** represent the *precise* timing of the Context changes and events contained within the Context packet. In this case, 'precise timing' means the best precision provided by the system. This **may** be as precise as the resolution of the Timestamp, i.e. either sample or picosecond resolution. This rule **shall** hold unless the Context Packet Stream uses the 'Timestamp Adjustment,' in which case the indicated time **shall** be adjusted as described in Section 7.1.5.13.

Permission 7.1.4-1: The precision of the Real-time Timestamp when the Timestamp mode is set to fine resolution **may** be something less than one picosecond. For example, it **may** be one nanosecond if that is the precision the system provides.

Rule 7.1.4-2: When the Timestamp Mode is set to coarse resolution (the TSM bit is set to one) the Timestamp fields in an IF Context packet, when present, **shall** represent the *general* timing of the Context changes and events

[*] *This mode cannot be used when the Described Signal is analog.*

contained within the Context packet. In this case, 'general timing' means that the contained Context events occurred sometime within the Data Sampling Interval of a Data packet in the paired Data Packet Stream.

Rule 7.1.4-3: When the Timestamp Mode is set to coarse resolution (the TSM bit is set to one) the Timestamp fields in an IF Context Packet, when present, **shall** match the Timestamp of a Data packet in the paired Data Packet Stream.

7.1.5 The Context Section

The last section of an IF Context packet is the Context Section which is organized as shown in Figure 7.1.5-1

Word	31	30	29	28	25	27	26	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	Context Indicator Field																															
⋮ ⋮ ⋮ ⋮ ⋮ ⋮	Context Fields (Present as indicated by the Context Indicators)																															

Figure 7.1.5-1: The Format of the Context Section

The first word of the Context Section is the Context Indicator field. The bits in the Context Indicator field indicate which of the optional Context Fields are present in the Context packet.

The remainder of the Context Section consists of the Context Fields which contain the metadata updates, i.e. Context updates. Each Context field is optional. Some contain information about the original RF frequency band and the resulting IF frequency band. Others contain information about power level, timing, and location. A full list is shown in Table 7.1.5-1 and each is described in detail in Sections 7.1.5.3 through 7.1.5.25.

A new Context packet does not need to be sent with each Data packet. Rather, Context packets only need to be sent when the relevant metadata changes. Context packets can be sent at a higher frequency if desired, but the minimum rate is determined by the rate of the underlying Context change. Therefore a receiver must assume that the value for a particular Context field does not change from the Timestamp of the Context packet that contained that Context field until the Timestamp of the next Context packet containing that same Context field. That is, the metadata contained in a Context field is understood to be *persistent* in between Context updates.

There are two exceptions to this general rule. First, the Over-range Count Context field is defined to give the number of over-ranged Data Samples in a single Data packet. Therefore the Over-range Count is not persistent between updates. Second, the user-defined fields in the "State and Event Indicators" Context field have user-defined persistence.

A period of validity listed as 'Persistent' in Table 7.1.5-1 indicates that it is not necessary to convey that Context field in a Context Packet unless there is a change to the underlying metadata because the metadata is understood not to have changed when the field is omitted. A period of validity listed as 'Single Data Packet' in Table 7.1.5-1 indicates that the underlying metatdata applies only to a single Data packet.

Rule 7.1.5-1: When the Context Packet Stream includes a 'Persistent' Context field, the emitter **shall** emit an IF Context Packet when the underlying metatdata changes. The emitter may emit that Context field more often if desired.

Permission 7.1.5-1: When the Context Packet Stream includes a 'Persistent' Context field, and the TSM bit corresponds to coarse resolution, multiple changes to the underlying metadata within a Data packet period may be aggregated into a single IF Context Packet whose Timestamp matches that Data packet's Timestamp.

Context Field	Number of Words in Context Field	Period of Validity
Context Field Change Identifier	0	N/A
Reference Point Identifier	1	Persistent
Bandwidth	2	Persistent
IF Reference Frequency	2	Persistent
RF Reference Frequency	2	Persistent
RF Reference Frequency Offset	2	Persistent
Band Frequency Offset	2	Persistent
Reference Level	1	Persistent
Gain	1	Persistent
Over-range Count	1	Single Data Packet
Sample Rate	2	Persistent
Timestamp Adjustment	2	Persistent
Timestamp Calibration Time	1	Persistent
Temperature	1	Persistent
Device Identifier	2	Persistent
State and Event Indicators	1	See Table 7.1.5.17-2
IF Data Packet Payload Format	2	Persistent
Formatted GPS (Global Positioning System) Geolocation	11	Persistent
Formatted INS (Inertial Navigation System) Geolocation	11	Persistent
ECEF (Earth-Centered, Earth-Fixed) Ephemeris	13	Persistent
Relative Ephemeris	13	Persistent
Ephemeris Reference Identifier	1	Persistent
GPS ASCII	Variable	Persistent
Context Association List	Variable	Persistent

Table 7.1.5-1: Size and Period of Validity of Context Fields

Table 7.1.5-1 also lists the number of 32-bit words required by each Context Field. The GPS ASCII and Context Association List fields are variable sized and each contains a subfield that specifies its size. See the corresponding Context field section for details.

7.1.5.1 The Context Indicator Field

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 Sections 7.1.5.3 through 7.1.5.25.

Rule 7.1.5.1-1: Every IF Context packet **shall** contain a Context Indicator field.

Rule 7.1.5.1-2: The bit fields within the Context Indicator field shall be arranged as shown in Table 7.1.5.1-1.

Rule 7.1.5.1-3: When a bit is set in the Context Indicator field, the corresponding Context field entry **shall** be included in the Context Section of the IF Context packet in the sequence shown in Table 7.1.5.1-1.

Bit Position	Context Indicator Field	Reference
31	Context Field Change Indicator	Section 7.1.5.2
30	Reference Point Identifier	Section 7.1.5.3
29	Bandwidth	Section 7.1.5.4
28	IF Reference Frequency	Section 7.1.5.5
27	RF Reference Frequency	Section 7.1.5.6
26	RF Reference Frequency Offset	Section 7.1.5.7
25	IF Band Offset	Section 7.1.5.8
24	Reference Level	Section 7.1.5.9
23	Gain	Section 7.1.5.10
22	Over-range Count	Section 7.1.5.11
21	Sample Rate	Section 7.1.5.12
20	Timestamp Adjustment	Section 7.1.5.13
19	Timestamp Calibration Time	Section 7.1.5.14
18	Temperature	Section 7.1.5.15
17	Device Identifier	Section 7.1.5.16
16	State and Event Indicators	Section 7.1.5.17
15	IF Data Packet Payload Format	Section 7.1.5.18
14	Formatted GPS (Global Positioning System) Geolocation	Section 7.1.5.19
13	Formatted INS (Inertial Navigation System) Geolocation	Section 7.1.5.20
12	ECEF (Earth-Centered, Earth-Fixed) Ephemeris	Section 7.1.5.21
11	Relative Ephemeris	Section 7.1.5.22
10	Ephemeris Reference Identifier	Section 7.1.5.23
9	GPS ASCII	Section 7.1.5.24
8	Context Association Lists	Section 7.1.5.25
7..0	Reserved	

Table 7.1.5.1-1: Context Indicator Field Definition

The following sections 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 just described.

7.1.5.2 The Context Field Change Indicator

In many systems it is desirable to periodically send Context information updates even when the Context information has not changed. Such periodic updates can be useful when Context packets may be corrupted or dropped and for handling start-up issues with downstream processors. However these updates may also cause unnecessary computational burdens on downstream processors. The Context Field Change Indicator bit indicates which IF Context packets contain new Context information and which contain Context information that was previously send in earlier IF Context packets.

Rule 7.1.5.2-1: The Context Field Change Indicator bit **shall** be set to zero when all the Context fields within the IF Context packet were conveyed in earlier IF Context packets. It **shall** be set to one when at least one Context field contains a new value.

Observation 7.1.5.2-1: The Over-range Count is not a persistent Context Field, and individual User-Defined State and Event Indicator Fields may not be persistent. These fields would not typically be included in an IF Context packet containing unchanged Context information..

7.1.5.3 The Reference Point Identifier

The purpose of the Reference Point Identifier is to indicate the point in the system where the Context applies. For example, when the RF Reference Frequency is sent in a Context Packet, it is desirable to state the location in the system where the Described Signal was centered at that frequency. The Reference Point Identifier (or Reference Point ID) contains the Stream ID assigned to the Reference Point. See Appendix B.1 for an example using the Reference Point Identifier field.

Rule 7.1.5.3-1: The Reference Point ID **shall** use the format shown in Figure 7.1.5.3-1.

Word	31..0
1	Reference Point Identifier

Figure 7.1.5.3-1: Reference Point ID Format

Rule 7.1.5.3-2: The Reference Point ID, when used, **shall** contain the Stream Identifier of the Reference Point.

Observation 7.1.5.3-1: The Reference Point for a VRT Data Packet Stream is often an analog signal somewhere upstream in the processing flow. In order for this signal to be indicated by the Reference Point ID it must be assigned a Stream ID.

Observation 7.1.5.3-2: 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 7.1.5.3-3: When an application needs to specify several unique Reference Points relating to a single Data Packet Stream, a separate Context Packet Stream must be generated to convey each Reference Point ID. These additional Context Packet Streams are each associated with the paired Context Packet Stream using the Context Association List mechanism described in Section 7.1.5.25.

Observation 7.1.5.3-4: 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.

7.1.5.4 The Bandwidth Field

The Bandwidth field is used to describe the amount of usable spectrum at the output of a process. The definition of usable spectrum is determined and specified by the VRT equipment provider. It is typically related to band-limiting filters, but may be limited by other criteria. For sampled data streams, the amount of usable spectrum is typically smaller than the Nyquist bandwidth. The amount of usable spectrum is also typically larger than the bandwidth of any particular signal of interest within that band. See Appendices B.2 and B.4 for examples using the Bandwidth field.

Rule 7.1.5.4-1: The Bandwidth field **shall** convey the amount of usable spectrum at the output of a process.

Documentation Rule 7.1.5.4-1: The Context Packet Class documentation **shall** specify the band-edge definition(s) used in calculating the Bandwidth field.

Observation 7.1.5.4-1: 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. The VRT equipment provider is responsible for defining and specifying the meaning of “usable spectrum” or “bandwidth.”

Rule 7.1.5.4-2: 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 7.1.5.4-1. 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.

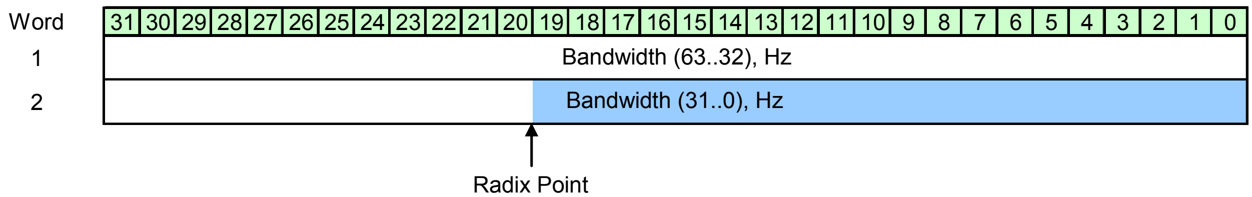


Figure 7.1.5.4-1: Bandwidth Field

Observation 7.1.5.4-2: 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 Band Frequency Offset field described in Section 7.1.5.8.

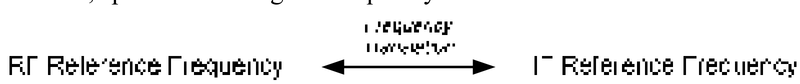
Observation 7.1.5.4-3: The value of the Bandwidth field has a valid range of 0.00 to 8.79 terahertz with a resolution of 0.95 microhertz. Negative values of Bandwidth are not valid.

Observation 7.1.5.4-4: A Bandwidth field value of 0x0000 0000 0010 0000 represents a bandwidth of 1 Hz. A Bandwidth field value of 0x0000 0000 0000 0001 represents a bandwidth of 0.95 microhertz.

7.1.5.5 The IF Reference Frequency Field

The IF Reference Frequency field has two distinct purposes. Its first purpose is to work in conjunction with the RF Reference Frequency field to identify the original RF frequency band from which a digital or analog IF signal* has been translated. Its second purpose is to identify the center of the band described by the Bandwidth field. The value in this field may be equal to the band center, or in some cases it may work in conjunction with the IF Band Offset field to indicate the band center.

The IF Reference Frequency and RF Reference Frequency fields work together to communicate the original RF frequency band of an IF signal. Neither field has meaning without the other. The IF Reference Frequency marks a point within the energy band of the Described Signal, typically its center frequency. The RF Reference Frequency, described in the next section, specifies the original frequency that was translated to the 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 7.1.5.7.

* *When used with the IF and RF Reference Frequency fields, the term “IF signal” refers to any signal after it has undergone frequency translation. The frequency of this signal may or may not be at typical IF frequencies. The signal may be anything from a complex baseband signal to a true RF signal.*

The IF Reference Frequency also serves as the point of reference for the center of the band described by the Bandwidth field. For many cases the center of the band will be at the IF Reference Frequency.

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

For cases where the IF Reference Frequency is not at the center of the band, a Band Frequency Offset field can be used to relate the IF Reference Frequency to the desired location in the band. When the Band Frequency Offset field is utilized, the relationship to the band center, as described here, is replaced by the relationship described in Section 7.1.5.8.

See Appendices B.2, B.3, and B.4 for examples using the IF Reference Frequency field.

Rule 7.1.5.5-1: The IF Reference Frequency field, when used, **shall** indicate a frequency within the range of usable spectrum described by the Bandwidth and Band Frequency Offset fields.

Rule 7.1.5.5-2: When the associated Data Packet Stream contains real Data Samples, the IF Reference Frequency **shall** be within the range of 0 and $+f_S/2$, where f_S is the sample rate. When the associated Data stream contains complex Data Samples, the IF Reference Frequency **shall** be within the range of $-f_S/2$ and $+f_S/2$. When the Described Signal is an analog signal, there are no restrictions on the IF Reference Frequency.

Recommendation 7.1.5.5-1: When the Described Signal is an analog signal, the IF Reference Frequency should be the IF center frequency.

Rule 7.1.5.5-3: 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 7.1.5.5-1. 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.

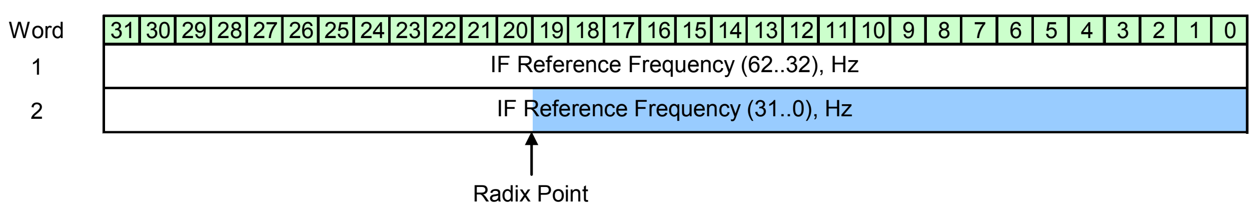


Figure 7.1.5.5-1: IF Reference Frequency Field

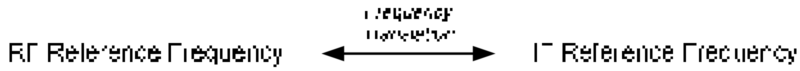
Observation 7.1.5.5-1: The spectrum of the Described Signal may be inverted. This is indicated by the Spectral Inversion Indicator bit in the State and Event Indicators field, described in Section 7.1.5.17.

Observation 7.1.5.5-2: The value of the IF Reference Frequency field has a range of ±8.79 terahertz with a resolution of 0.95 microhertz.

Observation 7.1.5.5-3: An IF Reference Frequency field value of 0x0000 0000 0010 0000 represents a frequency of 1 Hz. An IF Reference Frequency field value of 0xFFFF FFFF FFF0 0000 represents a frequency of –1 Hz. An IF Reference Frequency field value of 0x0000 0000 0000 0001 represents a frequency of 0.95 microhertz. An IF Reference Frequency field value of 0xFFFF FFFF FFFF FFFF represents a frequency of –0.95 microhertz.

7.1.5.6 The RF Reference Frequency Field

When a signal is downconverted to a lower frequency band, it is usually necessary to communicate the band of origin for the signal. In most cases this is accomplished using the IF and RF Reference Frequency fields. The location in the signal path that corresponds to the original frequency is indicated by the Reference Point Identifier described in 7.1.5.3. The original frequency that was ultimately translated to the IF Reference Frequency is specified with the RF Reference Frequency field.



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 7.1.5.7.

See Appendices B.2 and B.3 for examples using the RF Reference Frequency field.

Rule 7.1.5.6-1: 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 7.1.5.6-2: 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 7.1.5.6-1. 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.

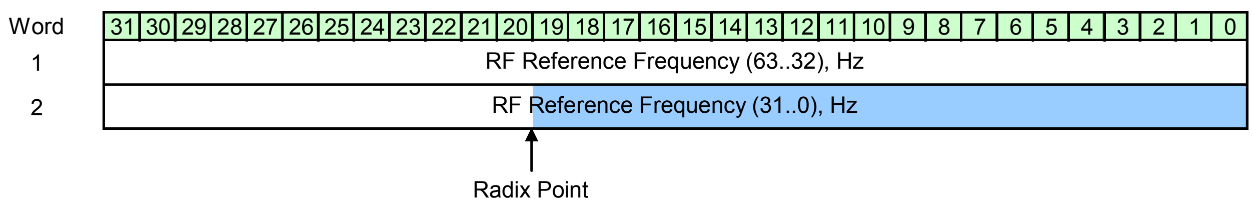


Figure 7.1.5.6-1: RF Reference Frequency Field

Observation 7.1.5.6-1: The value of the RF Reference Frequency field has a range of ±8.79 terahertz with a resolution of 0.95 microhertz.

Observation 7.1.5.6-2: An RF Reference Frequency field value of 0x0000 0000 0010 0000 represents a frequency of +1 Hz. An RF Reference Frequency field value of 0xFFFF FFFF FFF0 0000 represents a frequency of –1 Hz. An RF Reference Frequency field value of 0x0000 0000 0000 0001 represents a frequency of +0.95 microhertz. An RF Reference Frequency field value of 0xFFFF FFFF FFFF FFFF represents a frequency of –0.95 microhertz.

7.1.5.7 The RF Reference Frequency Offset Field

This field can be used to minimize link congestion for channelized data. Some processes, such as channelizers, create a large number of narrowband signals from a wideband input signal. Each of the channelizer 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 an update in only a single Context packet when the tuner frequency changes, 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 7.1.5.5 and 7.1.5.6. 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.

$$\text{RF Reference Frequency} + \text{RF Reference Frequency Offset} \xleftrightarrow{\text{Frequency translation}} \text{IF Reference Frequency}$$

See Appendix B.3 for an example using the RF Reference Frequency Offset field.

Rule 7.1.5.7-1: When the RF Reference Frequency Offset field is used, the sum of the RF Reference Frequency Offset and RF Reference Frequency conveyed in the Context Packet Stream **shall** be the frequency at the Reference Point that is translated to the frequency specified in the IF Reference Frequency field.

Observation 7.1.5.7-1: To find the frequency that translates to the IF Reference Frequency at a point upstream from the Reference Point of a particular Context Packet Stream, one must cascade several of the above relationship equations.

Rule 7.1.5.7-2: When the RF Reference Frequency Offset field is not specified in a Context Packet Stream it **shall** be assumed to be zero.

Rule 7.1.5.7-3: 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 7.1.5.7-1. 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.

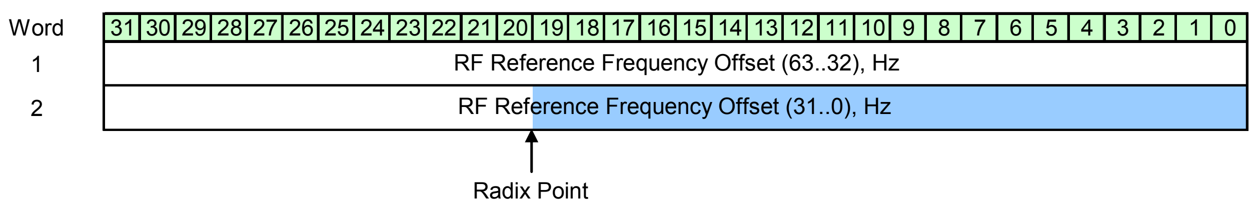


Figure 7.1.5.7-1: RF Reference Frequency Offset Field

Observation 7.1.5.7-2: The RF Reference Frequency Offset field is useful for describing the frequencies of a bank of filters, FFT channelizers, DDCs, etc.

Observation 7.1.5.7-3: The value of the RF Reference Frequency Offset field has a range of ±8.79 terahertz with a resolution of 0.95 microhertz.

Observation 7.1.5.7-4: An RF Reference Frequency Offset field value of 0x0000 0000 0010 0000 represents a frequency offset of +1 Hz. An RF Reference Frequency Offset field value of 0xFFFF FFFF FFF0 0000 represents a frequency offset of –1 Hz. An RF Reference Frequency Offset field value of 0x0000 0000 0000 0001 represents a frequency offset of +0.95 microhertz. An RF Reference Frequency Offset field value of 0xFFFF FFFF FFFF FFFF represents a frequency offset of –0.95 microhertz.

7.1.5.8 The IF Band Offset Field

Typically the spectral band whose width is described by the Bandwidth field of Section 7.1.5.4 is symmetric about the IF Reference Frequency described in Section 7.1.5.5. For cases where it is not symmetric about this frequency, the IF Band Offset field is used to specify the frequency offset from the IF Reference Frequency to the center of the band. When the IF Band 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 IF Band Offset.

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

See Appendix B.4 for an example using the IF Band Offset field.

Rule 7.1.5.8-1: The value of the IF Band Offset field **shall** be the band center frequency minus IF Reference Frequency, where the band center frequency is the center of the band whose width is described by the Bandwidth field.

Observation 7.1.5.8-1: The IF Band Offset field contains 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 7.1.5.8-2: The value of the IF Band Offset **shall** be expressed in units of Hertz. The IF Band Offset field **shall** use the 64-bit, two's-complement format shown in Figure 7.1.5.8-1. 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.

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	IF Band Offset (63..32), Hz																															
2													IF Band Offset (31..0), Hz																			

Radix Point

Figure 7.1.5.8-1: IF Band Offset Field

Observation 7.1.5.8-2: The value of the IF Band Offset field has a range of ±8.79 terahertz with a resolution of 0.95 microhertz.

Observation 7.1.5.8-3: A IF Band Offset field value of 0x0000 0000 0010 0000 represents a band offset of +1 Hz. A IF Band Offset field value of 0xFFFF FFFF FFF0 0000 represents a band offset of –1 Hz. A IF Band Offset field value of 0x0000 0000 0000 0001 represents a band offset of +0.95 microhertz. A IF Band Offset field value of 0xFFFF FFFF FFFF FFFF represents a band offset of –0.95 microhertz.

7.1.5.9 The Reference Level Field

The purpose of the Reference Level field is to relate the physical signal amplitude at the Reference Point (as identified by the Reference Point ID) with the values of the Data Samples in an 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 single sine wave at the Reference Point that results in a digitized sine wave with peak amplitude of one[*], in the payload of the paired Data Packet Stream.

The Reference Level field has no meaning for processes with analog outputs. See Appendix B.6 for an example using the Reference Level field.

Definition 7.1.5.9-1: A <u>Unit-Scale Sinusoid</u> is a hypothetical sinusoid whose peak values are -1 and +1 for signed number formats and 0 and +1 for unsigned number formats. Figure 7.1.5.9-1 shows the Unit-Scale Sinusoids for several data formats. These formats are explained in detail in Section 6.1.6.4.

- For real, signed Data Samples, the Unit-Scale Sinusoid has an amplitude of 1.
- For complex, signed Data Samples, the Unit-Scale Sinusoid falls on the unit circle on the complex plane.
- For unsigned Data Samples, the Unit-Scale Sinusoid is offset and scaled to the range from 0 to 1.
- For IEEE-754 floating-point numbers, the Unit-Scale Sinusoid has an amplitude of 1.

[*] *This is true for signed fixed-point and VRT numbers. For unsigned fixed-point and VRT numbers the peak-to-peak amplitude is one. The Normalized Interpretation is assumed.*

Observation 7.1.5.9-1: The Normalized Interpretation of fixed-point and VRT floating-point data formats does not allow them to actually represent the value +1. Nevertheless the Unit-Scale Sinusoid for these data formats has peak amplitude of one, just as in the case of floating-point Data Items. As an example, the Normalized Interpretation of 4-bit, signed, fixed-point data only ranges from -1 to +7/8, yet the corresponding Unit-Scale Sinusoid has peak values of -1 and +1.

Rule 7.1.5.9-1: The Reference Level field **shall** express the AC power of a single sinusoid at the Reference Point that would result in a Unit-Scale Sinusoid conveyed in the payload of the paired Data Packet Stream.

Rule 7.1.5.9-2: The value of the Reference Level field **shall** be expressed in units of dBm, where 0 dBm is one milliwatt (0.001 watt) measured with respect to a load termination of 50 ohms. The Reference Level field **shall** use the 32-bit format shown in Figure 7.1.5.9-2. 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.

Observation 7.1.5.9-2: The value of the Reference Level field has a range of nearly ±256 dBm with a resolution of 1/128 dBm (0.0078125 dBm).

Observation 7.1.5.9-3: A Reference Level field value of 0x0000 0080 represents a reference level of +1 dBm. A Reference Level field value of 0xFFFF FF80 represents a reference level of -1 dBm. A Reference Level field value of 0x0000 0001 represents a reference level of +0.0078125 dBm. A Reference Level field value of 0xFFFF FFFF represents a reference level of -0.0078125 dBm.

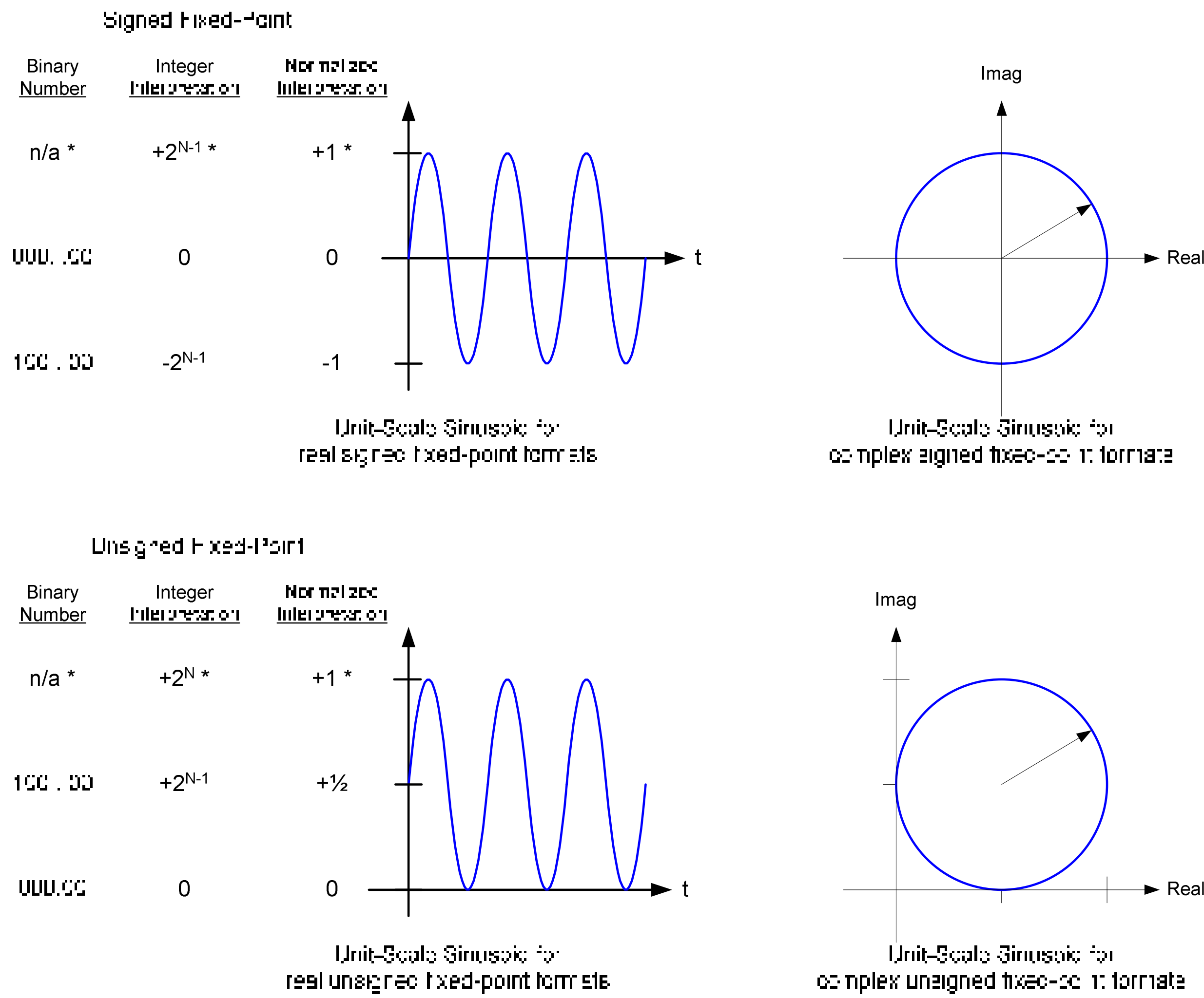


Figure 7.1.5.9-1: Unit-Scale Sinusoids*

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	Reserved																Reference Level (15..0), dBm															

Radix Point

Figure 7.1.5.9-2: Reference Level Field

* *According to the integer interpretation, a Unit-Scale Sinusoid spans the range from -2^{N-1} to $+2^{N-1}$ for N-bit signed fixed-point data. Although the most positive value of the Unit-Scale Sinusoid, $+2^{N-1}$, cannot be represented by an N-bit signed fixed-point number, Unit-Scale Sinusoids are still useful in describing the levels of signals represented by fixed-point numbers.The Normalized Interpretataion has a similar limitation, and the same answer applies.*

7.1.5.10 The Gain Field

The Gain field describes the amount of signal gain or attenuation from the Reference Point to the Described Signal. It can be used in conjunction with the Reference Level field to infer the signal level at various locations. See Appendix B.7 for an example using the Gain field.

The Gain Field contains two 16-bit subfields, Stage 1 Gain and Stage 2 Gain, which occupy the lower and upper 16 bits of the Gain Field, respectively. In RF equipment such as tuners and receivers, the total gain of the equipment is typically distributed to allow tradeoffs between noise power and linearity. For such equipment, Stage 1 Gain conveys the front-end or RF gain, and Stage 2 Gain conveys the back-end or IF gain. For equipment that does not require gain distribution, Stage 1 Gain provides the gain of the device, and Stage 2 Gain is set to zero.

Rule 7.1.5.10-1: The Gain field **shall** express the gain from the Reference Point to the Described Signal. When the Reference Point and the Described Signal are both digitized signals, the gain **shall** be calculated with respect to the Unit-Scale Sinusoids at the two points.

Observation 7.1.5.10-1: Consider a dgital Described Signal that has a Unit-Scale Sinusoidal signal at its digital Reference Point. A Gain of zero dB indicates that the Described Signal is also a Unit-Scale Sinusoid. Positive or negative gain values indicate that the Described Signal is greater-than or less-than a Unit-Scale Sinusoid, respectively. Note that it is not necessary for a Unit-Scale Sinusoid signal to be placed at the Reference Point or that it even be possible for a Unit-Scale Sinusoid signal to be expressed at the Reference Point. Also note that non-linear effects such as saturation are not considered when calculating the Gain. See Appendix B.7 for further details.

Rule 7.1.5.10-2: The Gain field **shall** only be used when the Reference Point and the Described Signal are either both analog signals or both digitized signals. A digital Described Signal with an analog Reference Point is more properly described using the Reference Level field.

Rule 7.1.5.10-3: The Gain field **shall** use the 32-bit format shown in Figure 7.1.5.10-1.

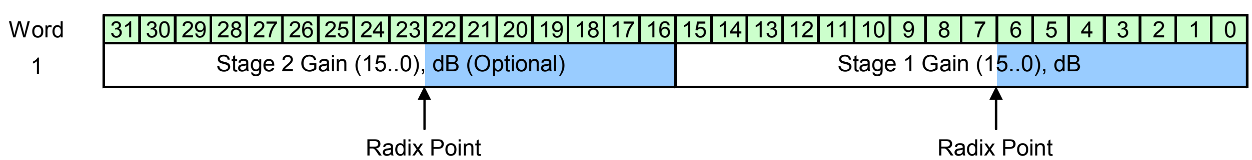


Figure 7.1.5.10-1: Gain Field

Rule 7.1.5.10-4: The Stage 1 Gain subfield **shall** be expressed in units of decibels (dB). The Stage 1 Gain value **shall** be expressed in two's-complement format in the lower 16 bits of the Gain field. This subfield has an integer and a fractional part, with the radix point to the right of bit 7 of the subfield.

Rule 7.1.5.10-5: The Stage 2 Gain subfield **shall** be expressed in units of decibels (dB). The Stage 2 Gain value **shall** be expressed in two's-complement format in the upper 16 bits of the Gain field. This subfield has an integer and a fractional part, with the radix point to the right of bit 7 of the subfield.

Rule 7.1.5.10-6: Equipment whose gain can be described with a single number **shall** use the Stage 1 Gain subfield. The Stage 2 Gain subfield **shall** be set to zero.

Observation 7.1.5.10-2: Processes introducing attenuation produce negative Gain values.

Observation 7.1.5.10-3: The values of the Gain subfields have a range of near ±256 dB with a resolution of 1/128 dB (0.0078125 dB).

Observation 7.1.5.10-4: A Gain field value of 0x0000 0080 represents a gain of +1 dB. A Gain field value of 0x0000 FF80 represents a gain of -1 dB. A Gain field value of 0x0000 0001 represents a gain of +0.0078125 dB. A Gain field value of 0x0000 FFFF represents a gain of -0.0078125 dB.

Rule 7.1.5.10-7: Equipment whose composite gain can be described with two numbers **shall** use the Stage 1 Gain subfield to describe the front-end or RF gain and the Stage 2 Gain subfield to describe the back-end or IF gain.

Rule 7.1.5.10-8: The sum of the Stage 1 and Stage 2 Gain subfields **shall** equal the total gain of the device being described by the Context packet.

Observation 7.1.5.10-5: A Gain field value of 0x0080 0080 represents front and back-end gains of +1 dB. A Gain field value of 0xFF80 FF80 represents front and back-end gains of -1 dB. A Gain field value of 0x0001 0001 represents front and back-end gains of +0.0078125 dB. A Gain field value of 0xFFFF FFFF represents front and back-end gains of -0.0078125 dB.

7.1.5.11 The Over-range Count Field

The Over-range Count field is used to convey the number of over-range Data Samples in a single paired Data packet.

Rule 7.1.5.11-1: The Over-range Count field **shall** contain the number of Data Samples in the paired Data packet whose amplitudes were beyond of the range of the Data Item format.

Rule 7.1.5.11-2: For complex Cartesian Data Samples, the Over-range Count field **shall** contain the total number of complex Data Samples in the paired Data packet for which either the real or imaginary component was beyond of the range of the Data Item format.

Rule 7.1.5.11-3: The Over-range Count field **shall** use the 32-bit, unsigned integer format shown in Figure 7.1.5.11-1.

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	Overflow Count (31..0)																															

Figure 7.1.5.11-1: Over-range Count Field

Observation 7.1.5.11-1: The value of the Over-range Count field applies only to the paired Data packet with the corresponding Timestamp. The Over-range Count does not accumulate over multiple Data packets.

Rule 7.1.5.11-4: When conveying the Over-range Count, the Timestamp Mode (TSM) field of the Context packet may indicate either coarse or fine resolution. When fine resolution is specified (the TSM bit is one), the Context packet Timestamp **shall** correspond to the Reference Point Time of a Data Sample when the reported Over-range Count was valid. That is, the Context packet Timestamp **shall** be equal to or later than the Reference Point Time of the Data Sample in the paired Data packet where the number of previous over-range events is equal to the number contained in the Over-range Count field and equal to or before the Reference Point Time of the last Data Sample in the Data packet. When fine resolution is specified (the TSM bit is zero), the Timestamp **shall** exactly match the Timestamp of the paired Data packet containing the over-range Data Samples.

7.1.5.12 The Sample Rate Field

Rule 7.1.5.12-1: The Sample Rate field **shall** express the sample rate of the Data Samples in the payload of the paired Data Packet Stream.

Rule 7.1.5.12-2: 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 7.1.5.12-1. 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.

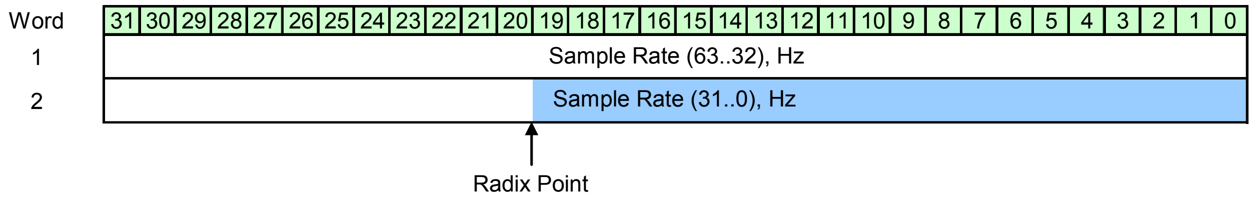


Figure 7.1.5.12-1: Sample Rate Field

Observation 7.1.5.12-1: The value of the Sample Rate field has a valid range of 0.00 to 8.79 terahertz with a resolution of 0.95 microhertz.

Observation 7.1.5.12-2: A Sample Rate field value of 0x0000 0000 0010 0000 represents a sample rate of 1 Hz. A Sample Rate field value of 0x0000 0000 0000 0001 represents a sample rate of 0.95 microhertz. Negative values of Sample Rate are not valid.

7.1.5.13 The Timestamp Adjustment Field

Typically, the purpose of a VRT system is to extract some information from a signal detected by one or more sensors. As an RF signal propagates through the system, it is typically delayed by the various processes, both before and after digitization. When the information is extracted from the signal, it is often important to calculate exactly when this information arrived at the sensor. The Timestamp field in the IF Data packet is often used to give the time of signal digitization. The Timestamp Adjustment field is used to adjust this Timestamp so that together they reflect the Reference Point Time. The Reference Point Time is when the information was present at the Reference Point, which may be some upstream analog process such as an antenna. See Appendix B.8 for an example using the Timestamp Adjustment field.

Rule 7.1.5.13-1: The sum of the Timestamp field in a Data packet and the most recently sent Timestamp Adjustment field in a paired Context packet **shall** be the Reference Point Time of the first Data Sample in that packet

Rule 7.1.5.13-2: When the Timestamp Mode is set to fine resolution (the TSM bit is set to zero), the sum of the Timestamp of the Context packet and the most recently sent Timestamp Adjustment **shall** be the Reference Point Time of the Context changes indicated in the Context packet.

Permission 7.1.5.13-1: The Timestamp Adjustment field **may** be set to zero, as long as Rule 7.1.5.13-1 and Rule 7.1.5.13-2 are not violated.

Permission 7.1.5.13-2: The Timestamp Adjustment field **may** be used to account for only a select subset of delays, as long as Rule 7.1.5.13-1 and Rule 7.1.5.13-2 are not violated. The Timestamp field would be adjusted to account for the remaining delays.

Observation 7.1.5.13-1: The Timestamp Adjustment field might only represent the cumulative analog delays upstream of an ADC.

Observation 7.1.5.13-2: In a typical application the Timestamp of the paired Data packet would give the ADC conversion time, and the Timestamp Adjustment field would adjust this time to indicate when the information represented by the first sample arrived at some upstream point such as an antenna.

Observation 7.1.5.13-3: In general, the use of the Timestamp Adjustment field does not negate the need to recalculate Timestamps. For example, VRT packet size typically varies from the input to the output of a process, necessitating the synthesis of new Timestamps. As another example, when resampling occurs, new Timestamps are usually needed.

Rule 7.1.5.13-3: The Timestamp Adjustment field **shall** be expressed in units of picoseconds. The Timestamp Adjustment field **shall** use the 64-bit, two's-complement "Real Time" format described in Section 6.1.5 and shown in Figure 7.1.5.13-1.

Observation 7.1.5.13-4: The Timestamp Adjustment field is expressed in picoseconds because this format allows the expression of time independent of the signal sampling rate. This makes it equally usable for timing related to either analog or digital signals.

<table>
<tr><th>Word</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td colspan="32">Timestamp Adjustment (63..32), picoseconds</td></tr>
<tr><td>2</td><td colspan="32">Timestamp Adjustment (31..0), picoseconds</td></tr>
</table>

Figure 7.1.5.13-1: Timestamp Adjustment Field

Observation 7.1.5.13-5: The range of the Timestamp Adjustment field is ±9.2 million seconds. The resolution of the Timestamp Adjustment field is 1 picosecond.

7.1.5.14 The Timestamp Calibration Time Field

The Timestamp Calibration Time field conveys the date and time at which the Timestamp in the Data and Context packets was known to be correct. The Timestamp Calibration Time is useful in situations where GPS loss-of-signal causes the local reference oscillator to become free-running or where UTC time is not constantly monitored to detect the insertion or removal of leap-seconds. The Timestamp Calibration Time field uses the same timebase format used for the Integer-seconds Timestamp field as indicated by the TSI field.

Rule 7.1.5.14-1: The Timestamp Calibration Time field **shall** use the 32-bit format shown in Figure 7.1.5.14-1.

<table>
<tr><th>Word</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td colspan="32">Timestamp Calibration Time (31..0), seconds</td></tr>
</table>

Figure 7.1.5.14-1: Timestamp Calibration Time Field

Rule 7.1.5.14-2: The Timestamp Calibration Time field **shall** express the most recent time when the Timestamp field was known to be correct.

Rule 7.1.5.14-3: The Timestamp Calibration Time field **shall** use the same timebase format used in the Timestamp field as given by the TSI field of the Context packet.

7.1.5.15 The 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 Described Signal.

Rule 7.1.5.15-1: 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 7.1.5.15-1. 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.

Word	31 … 16	15 … 0
1	Reserved	Temperature (15..0), degrees Celsius

Radix Point

Figure 7.1.5.15-1: Temperature Field

Observation 7.1.5.15-1: The valid range of the Temperature field is -273.15 °C to +511.984375 °C. The resolution of the Temperature field is 0.015625 °C (1/64 °C).

Observation 7.1.5.15-2: A Temperature field value of 0x0000 0040 represents a temperature of +1 °C. A Temperature field value of 0xFFFF FFC0 represents a temperature of -1°C. A Temperature field value of 0x0000 0001 represents a temperature of +0.015625 °C. A Temperature field value of 0xFFFF FFFF represents a temperature of -0.015625 °C.

7.1.5.16 The Device Identifier Field

The Device Identifier field is used to identify the manufacturer and model of the device generating an IF Context Packet Stream. It contains a manufacturer OUI subfield which specifies the manufacturer of the emitting device and a subfield that contains a code that uniquely identifies a particular model for that manufacturer. The Device Identifier field differs from the Class Identifier field optionally included in all packet types. The Device Identifier field specifies the manufacturer of the device emitting the VRT Packet Stream, whereas the Class Identifier field specifies the organization that defined the format of the VRT Packet Stream and a unique code to identify that format.

Rule 7.1.5.16-1: The Device Identifier field **shall** use the format shown in Figure 7.1.5.16-1.

Word	31 … 24	23 … 16	15 … 0
1	Reserved	Manufacturer OUI (23..0)	
2	Reserved		Device Code (15..0)

Figure 7.1.5.16-1: Device Identifier Field

Rule 7.1.5.16-2: The Manufacturer OUI subfield **shall** contain the 24-bit, IEEE-registered, Organizationally Unique Identifier (company identifier) of the manufacturer of the device which generated the IF Context packet.

Rule 7.1.5.16-3: The Device Code subfield **shall** contain a 16-bit number to identify the model of the device emitting the IF Context Packet Stream. For each manufacturer the Device Code shall be unique for each device model that emits VRT Packet Streams.

Recommendation 7.1.5.16-1: The manufacturer **should** maintain a complete list of all Device Codes.

Permission 7.1.5.16-1: The Manufacturer OUI subfield in the Device Identifier **may** be different than the OUI subfield in the Class ID Field of the IF Context packet described in Section 7.1.3.

7.1.5.17 The State and Event Indicator Field

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

Rule 7.1.5.17-1: The form of the State and Event Indicator field **shall** follow that shown in Figure 7.1.5.17-1, shown in greater detail in Table 7.1.5.17-2.

Word	31..20	19..8	7..0
1	Enables	Indicators	User-Defined

Figure 7.1.5.17-1: State and Event Indicator Fields

Table 7.1.5.17-2 also lists the period of validity of each State and Event Indicator. Definitions and Rules regarding these periods of validity are given in Section 7.1.5.

Enable Bit Position	Indicator Bit Position	Indicator Name	Period of Validity
31	19	Calibrated Time Indicator	Persistent
30	18	Valid Data Indicator	Persistent
29	17	Reference Lock Indicator	Persistent
28	16	AGC/MGC Indicator	Persistent
27	15	Detected Signal Indicator	Persistent
26	14	Spectral Inversion Indicator	Persistent
25	13	Over-range Indicator	Single Data Packet
24	12	Sample Loss Indicator	Single Data Packet
[23..20]	[11..8]	Reserved	N/A
Bit Position		Function	
[7..0]		User-Defined	User-Defined

Table 7.1.5.17-2: State and Event Indicator Bit Definitions

Rule 7.1.5.17-2: For each Indicator in bit position [19..8] there is a corresponding Enable in bit position [31..20]. The Indicator/Enable pairs **shall** be in the bit positions indicated in Table 7.1.5.17-2.

Rule 7.1.5.17-3: Indicator/Enable pairs that are supported by a Packet Class or Information Class (that is, those that are included in the Packet Class or Information Class documentation) **shall** function as described in the remainder of this section. The meanings of Indicator bits that are not supported by a Packet Class or Information Class are undefined.

Observation 7.1.5.17-1: The time at which Indicators are valid is given by Timestamp of the Context packet, and is governed by the rules in Section 7.1.4.

Rule 7.1.5.17-4: The Calibrated Time Indicator, when set to one, **shall** indicate that the Timestamps in the Context Packet Stream and in the associated Data Packet Stream are calibrated to some external reference. When set to zero this Indicator **shall** indicate that the Timestamps are free-running and may be inaccurate.

Rule 7.1.5.17-5: The Valid Data Indicator, when set to one, **shall** indicate that the Data in the associated Data packet is valid. When set to zero it **shall** indicate that some condition exists that may invalidate the Data.

Documentation Rule 7.1.5.17-1: The Packet Class Documentation or the Information Class documentation **shall** specify the meaning of the Valid Data Indicator unless it is unused.

Rule 7.1.5.17-6: The Reference Lock Indicator, when set to one, **shall** indicate that any phase-locked loops (PLL) affecting the Described Signal are locked and stable. When set to zero it **shall** indicate that at least one PLL is not locked and stable.

.

Rule 7.1.5.17-7: The AGC/MGC Indicator, when set to one, **shall** indicate that AGC (Automatic Gain Control) is active. When set to zero, it **shall** indicate MGC (Manual Gain Control).

Rule 7.1.5.17-8: The Detected Signal Indicator, when set to one, **shall** indicate that Described Signal contains some detected signal.

Documentation Rule 7.1.5.17-2: The Packet Class documentation or Information Class documentation **shall** specify what type of signal or signal feature is being detected.

Rule 7.1.5.17-9: The Spectral Inversion Indicator, when set to one, **shall** indicate that the spectrum of the signal conveyed in the data payload is inverted with respect to the signal at the Reference Point.[*]

Rule 7.1.5.17-10: When the Timestamp Mode is set to fine resolution (the TSM bit is set to zero) , the Over-range Indicator, when set to one, **shall** indicate that the Data Sample in the paired Data packet stream was over-range at the time given in the Context packet Timestamp. When the Timestamp Mode is seto to coarse resolution (the TSM bit is set to one), the Over-range Indicator, when set to one, **shall** indicate that at least one Data Sample in the paired Data packet with corresponding Timestamp is over-range. The definition of over-range is given in Rule 7.1.5.11-1.

Rule 7.1.5.17-11: When the Timestamp Mode is set to fine resolution (the TSM bit is set to zero) , the Sample Loss Indicator, when set to one, **shall** indicate that the paired Data packet contains a Data Sample discontinuity due to processing errors and/or buffer overflow at the time given in the Context packet Timestamp. When the Timestamp Mode is seto to coarse resolution (the TSM bit is set to one), the Sample Loss Indicator, when set to one, **shall** indicate that the paired Data packet with corresponding Timestamp contains at least one Data Sample discontinuity due to processing errors and/or buffer overflow.

Rule 7.1.5.17-12: The Indicator and Enable bits that are reserved **shall** be set to zero.

Rule 7.1.5.17-13: When an Enable bit is set to zero in a particular IF Context packet, the assumed state of the corresponding Indicator bit **shall** be the state most recently communicated by a previous Context packet.

Recommendation 7.1.5.17-1: IF Context packet emitters **should** keep the Enable bit asserted if behavior described in Rule 7.1.5.17- is not desired.

Rule 7.1.5.17-14: The Indicators used in an IF Context packet **shall** apply to the Data Packet Stream associated with the Context packet sending the Indicators.

Permission 7.1.5.17-1: The user-defined bits in positions [7..0] **may** be used for any purpose.

Observation 7.1.5.17-2: The user-defined bits in positions [7..0] 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.

Documentation Rule 7.1.5.17-3: The Context Packet Class documentation **shall** provide a complete specification of all “user-defined” bits in use in this field along with the binary codes used.

[*] *See Appendix B.5 for an example using the Spectral Inversion Indicator.*

Observation 7.1.5.17-3: When using persistent State and Event Indicators, IF Context packet emitters must send two Context packets to indicate the timing of events: one packet to indicate the start time of the event and one packet to indicate the finish time of the event.

7.1.5.18 The Data Packet Payload Format Field

The Data Packet Payload Format field is used to convey the parameters described in Section 6.1.6 that specify the packing and content of the payload of the paired Data Packet Stream. See Appendix B.9 for examples using the Data Packet Payload Format field.

Rule 7.1.5.18-1: The Data Packet Payload Format field **shall** be arranged as shown in Figure 7.1.5.18-1. The constituent parameter fields are described in Table 7.1.5.18-1 and Table 7.1.5.18-2.

<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>Pack</td><td colspan="2">Real/ Cmplx</td><td colspan="5">Data Item Format</td><td>Rpt</td><td colspan="3">Event-Tag Size</td><td colspan="4">Channel-Tag Size</td><td colspan="4">Reserved</td><td colspan="6">Item Packing Field Size</td><td colspan="6">Data Item Size</td></tr>
<tr><td>2</td><td colspan="16">Repeat Count</td><td colspan="16">Vector Size</td></tr>
</table>

Figure 7.1.5.18-1: Organization of the Data Packet Payload Format Field

Bit Position	Field	Field Width (bits)
[31]	Packing Method	1
[30..29]	Real/Complex Type	2
[28..24]	Data Item Format	5
[23]	Sample-Component Repeat Indicator	1
[22..20]	Event-Tag Size	3
[19..16]	Channel-Tag Size	4
[15..12]	Reserved (set to zero)	4
[11..6]	Item Packing Field Size	6
[5..0]	Data Item Size	6

Table 7.1.5.18-1: Location of Parameter Fields in the First Payload Format Word

Bit Position	Field	Field Width (bits)
[31..16]	Repeat Count	16
[15..0]	Vector Size	16

Table 7.1.5.18-2: Location of Parameter Fields in the Second Payload Format Word

Rule 7.1.5.18-2: The "Packing Method" field **shall** be set to one when link-efficient packing is used in the paired Data Packet Stream. It **shall** be set to zero when processing-efficient packing is used.

Rule 7.1.5.18-3: The "Real/Complex Type" field **shall** indicate whether the Data Samples are real or one of the complex types using the appropriate code listed in Table 7.1.5.18-3.

Code	Data Sample Type
00	Real
01	Complex, Cartesian
10	Complex, Polar
11	Reserved

Table 7.1.5.18-3: Data Sample Format Codes

Rule 7.1.5.18-4: The "Data Item Format" field **shall** contain the appropriate 5-bit code to indicate the type of Data Items used in the paired Data Packet Stream. The code **shall** be chosen according to Table 7.1.5.18-4.

Code	Data Item Type	Code	Data Item Type
00000	Signed Fixed-Point	10000	Unsigned Fixed-Point
00001	Signed VRT, 1-bit exponent	10001	Unsigned VRT, 1-bit exponent
00010	Signed VRT, 2-bit exponent	10010	Unsigned VRT, 2-bit exponent
00011	Signed VRT, 3-bit exponent	10011	Unsigned VRT, 3-bit exponent
00100	Signed VRT, 4-bit exponent	10100	Unsigned VRT, 4-bit exponent
00101	Signed VRT, 5-bit exponent	10101	Unsigned VRT, 5-bit exponent
00110	Signed VRT, 6-bit exponent	10110	Unsigned 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	Reserved
01110	IEEE-754 Single-Precision Floating-Point	11110	Reserved
01111	IEEE-754 Double-Precision Floating-Point	11111	Reserved

Table 7.1.5.18-4: Data Item Format Codes

Rule 7.1.5.18-5: The "Sample-Component Repeat Indicator" **shall** be set to one when Sample Component Repeating is in use in the paired Data Packet Stream. Otherwise it **shall** be set to zero.

Rule 7.1.5.18-6: The "Event-Tag Size" field **shall** contain an unsigned number equal to the Event-Tag size used in the paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 7.1.5.18-7: The "Channel-Tag Size" field **shall** contain an unsigned number equal to the Channel-Tag size used in the paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 7.1.5.18-8: 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 paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 7.1.5.18-9: The "Data Item Size" field **shall** contain an unsigned number that is one less than the actual Data Item size in the paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Rule 7.1.5.18-10: The "Repeat Count" field **shall** contain an unsigned number that is one less than the actual Repeat Count used in the paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Observation 7.1.5.18-1: The “Repeat Count” refers to either the Channel Repeating count or the Sample Component Repeating count. Only one of these repeat methods may be used at one time.

Rule 7.1.5.18-11: The “Vector Size” field **shall** contain an unsigned number that is one less than the actual Vector size in the paired Data Packet Stream. The lsb of the field **shall** be the rightmost bit in the field.

Observation 7.1.5.18-2: The “Item Packing Field Size”, “Data Item Size”, “Repeat Count”, and “Vector Size” fields contain a number that is *one less* than the size or count being used. The “Event-Tag Size” and “Channel-Tag Size” fields contain a number *equal* to the size of those fields.

7.1.5.19 The 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 7.1.5.19-1.

Word	31..28	27..26	25..24	23..0
1	Reserved	TSI	TSF	GPS/INS Manufacturer OUI
2	Integer-second Timestamp of Position Fix (31..0)			
3	Fractional-second Timestamp of Position Fix (63..32)			
4	Fractional-second Timestamp of Position Fix (31..0)			
5	Latitude (31..0), degrees			
6	Longitude (31..0), degrees			
7	Altitude (31..0), meters			
8	Speed over Ground (31..0), meters/second			
9	Heading Angle (31..0), degrees			
10	Track Angle (31..0), degrees			
11	Magnetic Variation (31..0), degrees			

Figure 7.1.5.19-1: Formatted GPS and INS Geolocation Fields

Rule 7.1.5.19-1: The Formatted GPS Geolocation field **shall** be formatted as shown in Figure 7.1.5.19-1.

Rule 7.1.5.19-2: The GPS/INS Manufacturer OUI subfield **shall** contain the 24-bit field for the IEEE-assigned Organizationally Unique Identifier (company identifier) of the GPS/INS manufacturer.

The definitions of the TSI and TSF subfield codes in the Formatted GPS and INS Geolocation Fields match that of the IF Data packet except that the ‘00’ code is redefined to mean that the corresponding Timestamp field is included in the packet but it’s contents are not specified. With this redefinition, the Geolocation field maintains a constant length which is helpful when parsing IF Context packets.

Rule 7.1.5.19-3: The TSI field in the Formatted GPS Geolocation field **shall** accurately indicate the type of Integer-seconds Timestamp included in the packet according to the code assignments in Table 7.1.5.19-1.

TSI code	Meaning
00	Undefined
01	Coordinated Universal Time (UTC)
10	GPS Time
11	Other

Table 7.1.5.19-1 : Codes for the Integer-second Portion of the Timestamp of Position Fix

Rule 7.1.5.19-4: The TSF field in the Formatted GPS Geolocation field **shall** accurately indicate the type of Fractional-seconds Timestamp included in the packet according to the code assignments in Table 7.1.5.19-2.

TSF code	Meaning
00	Undefined
01	Sample Count Time
10	Real (Picoseconds) Time
11	Free-Running Count Time

Table 7.1.5.19-2 : Codes for the Fractional-second Portion of the Timestamp of Position Fix

Rule 7.1.5.19-5: When the TSI or TSF fields are non-zero the corresponding Timestamp of Position Fix subfield **shall** express the time of the most recent location fix in the format given in Sections 6.1.4 and 6.1.5.

Rule 7.1.5.19-1: When the TSI or TSF fields are zero the corresponding Timestamp of Position Fix subfield words **shall** take the value 0xFFFFFFFF.

Observation 7.1.5.19-1: The Timestamp of Position Fix subfields may differ from the Timestamp fields in the Context packet that contains them. Note that the Timestamp fields in the Formatted GPS/INS Geolocation field refer to the time of the most recent GPS/INS position fix, which may not be the current time or the timestamp of the IF Context packet itself. These fields are useful for indicating the loss of the GPS signal.

Definition 7.1.5.19-1: 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 7.1.5.19-2. This field has an integer and a fractional part with the radix point to the right of bit 22.

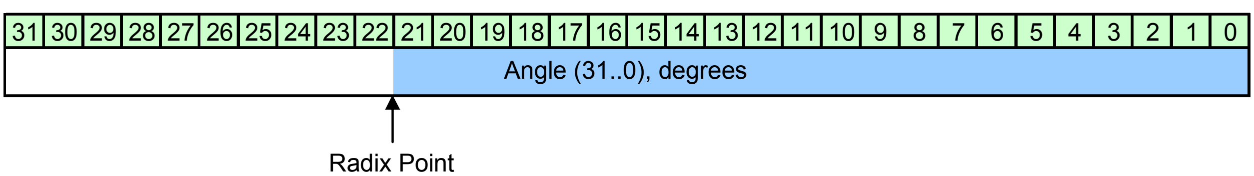


Figure 7.1.5.19-2: Geolocation Angle Format

Observation 7.1.5.19-2: The Geolocation Angle Format has a possible range of ±512 degrees and a resolution of 2.38e-7 degrees. Particular angular measurements 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 7.1.5.19-3: On the surface of the Earth the latitude and longitude angular resolution provide a Cartesian resolution on the order of a few centimeters.

Rule 7.1.5.19-6: The Latitude and Longitude subfields **shall** use the Geolocation Angle Format shown in Figure 7.1.5.19-2.

Rule 7.1.5.19-7: The Latitude subfield in the GPS Geolocation Field value **shall** range from –90.0 (South) to +90.0 (North) degrees.

Rule 7.1.5.19-8: The Longitude subfield in the GPS Geolocation Field value **shall** range from –180.0 (West) to +180.0 (East) degrees.

Rule 7.1.5.19-9: The Altitude subfield in the GPS Geolocation Field **shall** use the 32-bit, two's-complement format shown in Figure 7.1.5.19-3. 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.

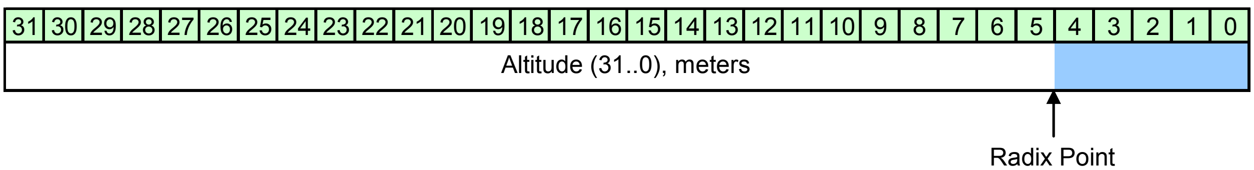


Figure 7.1.5.19-3: Altitude Subfield Format

Observation 7.1.5.19-4: The Altitude subfield has a range of ±67108 kilometers and a resolution of 3.1 centimeters.

Documentation Rule 7.1.5.19-1: The Context Packet Class documentation **shall** specify the Altitude reference. For GPS hardware this is typically the height above the WGS-84 ellipsoid. For INS hardware this is typically the height above mean sea level.

Rule 7.1.5.19-10: The Speed Over Ground subfield **shall** use the 32-bit, two's-complement format shown in Figure 7.1.5.19-4. 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.

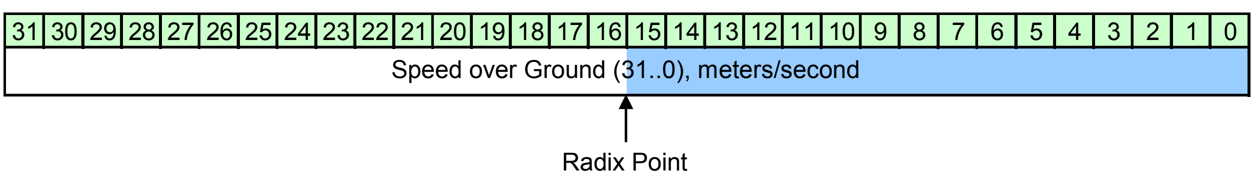


Figure 7.1.5.19-4: Speed Over Ground Subfield Format

Observation 7.1.5.19-5: The Speed Over Ground subfield has a range of 0 to 65636 m/s and a resolution of 1.5e-5 m/s.

Rule 7.1.5.19-11: The Heading Angle subfield **shall** use the Geolocation Angle Format shown in Figure 7.1.5.19-2. 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.

Rule 7.1.5.19-12: The Track Angle subfield **shall** use the Geolocation Angle Format shown in Figure 7.1.5.19-2. The Track Angle **shall** express the platform's direction of travel with respect to true North in decimal degrees. The Track Angle value **shall** range from 0.0 to +359.999999761582 degrees.

Observation 7.1.5.19-6: Figure 7.1.5.19-5 illustrates the definition of several navigation terms. The course is the direction from the starting point to the destination. The bearing is the direction from the current platform location to the destination. While the platform is moving the bearing may change but the course does not. The track is the actual path the platform takes while moving from the starting point to the destination. The Track Angle is the direction the platform is moving (at the rate indicated by the Speed Over Ground subfield) and the Heading Angle is the direction the platform is pointed. External forces on the platform such as wind or water currents may cause the Track Angle to differ from the Heading Angle. Only the Heading Angle and Track Angle are conveyed in the Formatted GPS/INS Geolocation field.

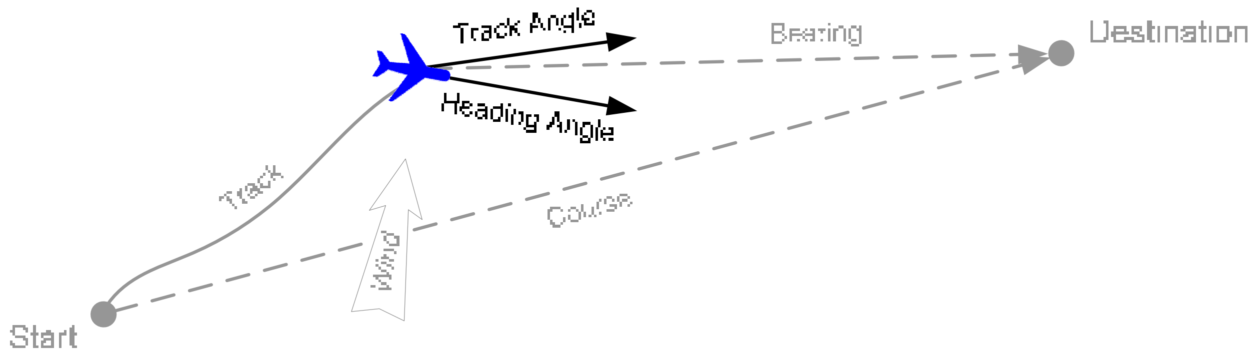


Figure 7.1.5.19-5: Navigation Angles

Rule 7.1.5.19-13: 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 7.1.5.19-14: The Latitude, Longitude, Altitude, Speed Over Ground, Heading, Track Angle, and Magnetic Variation subfields **shall** take the value 0x7FFFFFFF when unspecified.

Observation 7.1.5.19-7: These default subfield values are invalid or practically impossible for all GPS and INS Geolocation subfields.

7.1.5.20 The 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 7.1.5.20-1: The Formatted INS Geolocation field **shall** follow the same rules as the Formatted GPS Geolocation field.

7.1.5.21 The 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. See Appendix B.10 for an example using the ECEF Ephemeris field.

Figure 7.1.5.21-1 shows the interpretation of the various components of ECEF location.

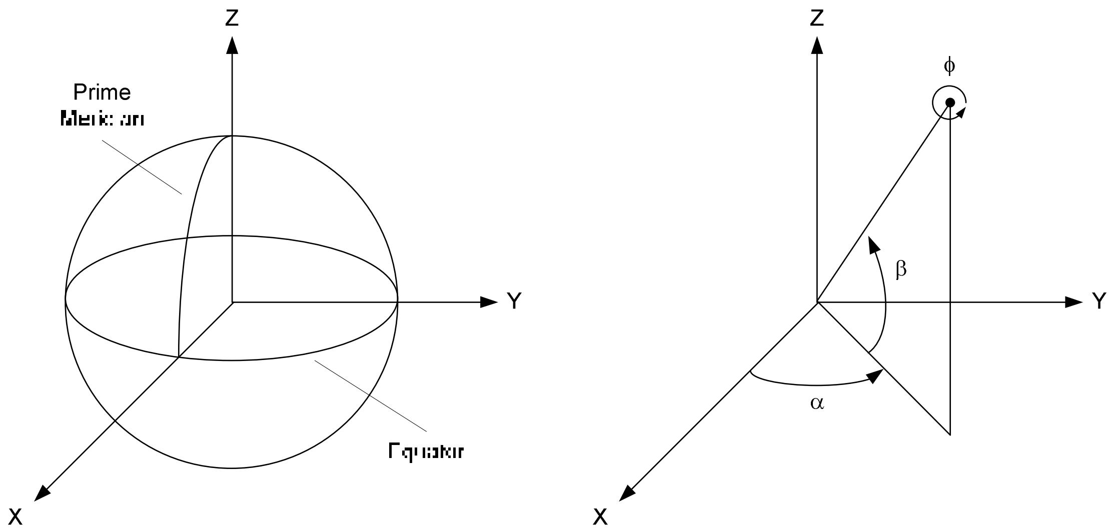


Figure 7.1.5.21-1: ECEF Ephemeris Coordinates

Rule 7.1.5.21-1: 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 7.1.5.21-1, 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 7.1.5.21-2: The attitude coordinates of the ECEF Ephemeris fields **shall** be specified in the Earth Centered, Earth Fixed Coordinate System as shown in Figure 7.1.5.21-1, 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 7.1.5.22) share the same format, shown below.

Rule 7.1.5.21-3: The ECEF Ephemeris field **shall** be expressed using the format shown in Figure 7.1.5.21-2.

<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="4">Reserved</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="24">GPS/INS Manufacturer OUI</td></tr>
<tr><td>2</td><td colspan="32">Integer-second Timestamp of Position Fix (31..0)</td></tr>
<tr><td>3</td><td colspan="32">Fractional-second Timestamp of Position Fix (63..32)</td></tr>
<tr><td>4</td><td colspan="32">Fractional-second Timestamp of Position Fix (31..0)</td></tr>
<tr><td>5</td><td colspan="32">Position X (31..0), meters</td></tr>
<tr><td>6</td><td colspan="32">Position Y (31..0), meters</td></tr>
<tr><td>7</td><td colspan="32">Position Z (31..0), meters</td></tr>
<tr><td>8</td><td colspan="32">Attitude Alpha (31..0), degrees</td></tr>
<tr><td>9</td><td colspan="32">Attitude Beta (31..0), degrees</td></tr>
<tr><td>10</td><td colspan="32">Attitude Phi (31..0), degrees</td></tr>
<tr><td>11</td><td colspan="32">Velocity dX (31..0), meters/second</td></tr>
<tr><td>12</td><td colspan="32">Velocity dY (31..0), meters/second</td></tr>
<tr><td>13</td><td colspan="32">Velocity dZ (31..0), meters/second</td></tr>
</table>

Figure 7.1.5.21-2: ECEF and Relative Ephemeris Fields

Rule 7.1.5.21-4: The TSI, TSF, OUI, and Timestamp of Position Fix fields **shall** follow the rules of the corresponding Formatted GPS Geolocation fields given in Section 7.1.5.19.

Rule 7.1.5.21-5: The position coordinates of the ECEF Ephemeris field **shall** use the 32-bit, two's-complement "Position" format shown in Figure 7.1.5.21-3. 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 7.1.5.21-6: The attitude coordinates of the ECEF Ephemeris field **shall** use the 32-bit, two's-complement "Attitude" format shown in Figure 7.1.5.21-3. 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 7.1.5.21-7: The velocity coordinates of the ECEF Ephemeris field **shall** use the 32-bit, two's-complement "Velocity" format shown in Figure 7.1.5.21-3. 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.

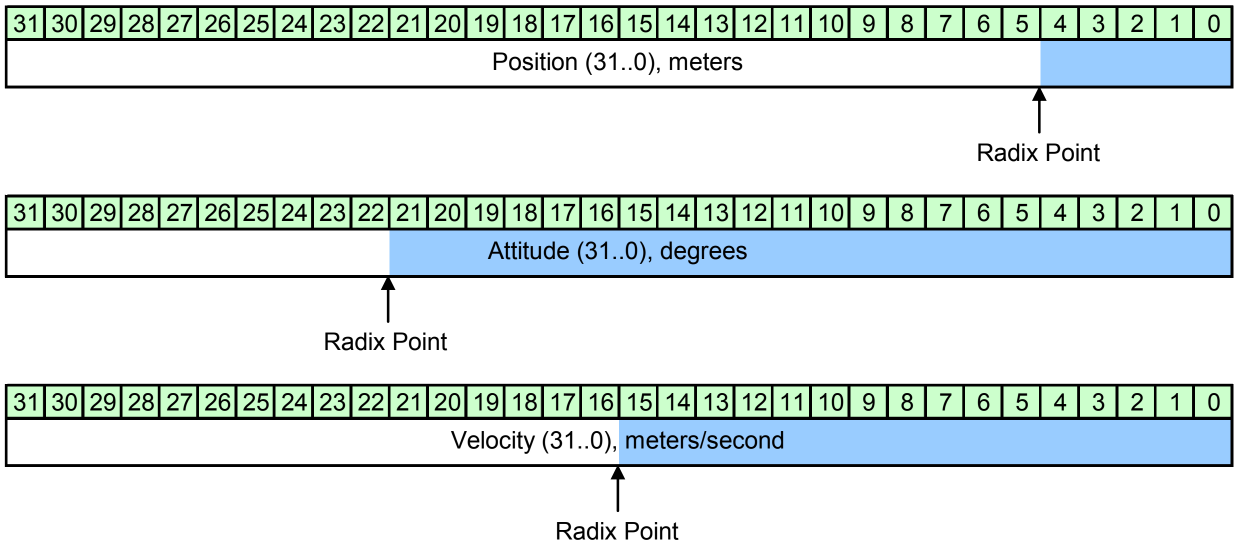


Figure 7.1.5.21-3: Position, Attitude and Velocity Coordinate Formats

Rule 7.1.5.21-8: Each word of the ECEF Ephemeris field **shall** take the value 0x7FFF FFFF when unknown.

7.1.5.22 The 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 reference frame of the Relative Ephemeris coordinate system is user-defined. The ECEF and Relative Ephemeris coordinate systems may be used together if the transformation 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 origin and axis orientation for the Relative Ephemeris coordinate system are equipment provider specified. If desired, the Relative Ephemeris origin may be given in the ECEF Ephemeris field of the process pointed to by the Ephemeris Reference Identifier described in Section 7.1.5.23.

See Appendix B.10 for an example using the Relative Ephemeris field.

Rule 7.1.5.22-1: The Relative Ephemeris field **shall** be expressed using the formats shown in Figure 7.1.5.21-2 and Figure 7.1.5.21-3.

Documentation Rule 7.1.5.22-1: The Packet Class documentation **shall** specify the coordinate system the Relative Ephemeris is referenced to.

Documentation Rule 7.1.5.22-2: When the Relative Ephemeris field is used in conjunction with the ECEF Ephemeris field, the Context Packet Class documentation **shall** provide the transformation between the two coordinate systems.

Observation 7.1.5.22-1: 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 antennas on the aircraft are specified in the Relative Ephemeris coordinates. See Appendix B.10 shows this example in two dimensions.

Observation 7.1.5.22-2: 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. See Appendix B.10 for an example where Relative Ephemeris is converted into ECEF coordinates.

Observation 7.1.5.22-3: The attitude coordinates can take on more than one meaning, as demonstrated in Table 7.1.5.22-1. 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 7.1.5.22-1: Two Common Attitude Coordinate Interpretations

7.1.5.23 The 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. See Appendix B.10 for an example using the Ephemeris Reference Identifier field.

The Ephemeris Reference ID uses the format shown in Figure 7.1.5.23-1.

Word	31–0
1	Ephemeris Reference Identifier

Figure 7.1.5.23-1: Ephemeris Reference Identifier Format

Rule 7.1.5.23-1: The Ephemeris Reference Identifier, when used, **shall** contain the Stream ID of the VRT Context Packet Stream whose ECEF Ephemeris is necessary to translate the Relative Ephemeris given in this Context Packet Stream to ECEF coordinates.

7.1.5.24 The 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 7.1.5.24-1: The GPS ASCII field **shall** follow the format shown in Figure 7.1.5.24-1.

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	Reserved								GPS/INS Manufacturer OUI																							
2	Number of Words																															
3	Byte 1								Byte 2								Byte 3								Byte 4							
⋮	⋮								⋮								⋮								⋮							
N+2	Byte 4N-3								Byte 4N-2								Byte 4N-1								Byte 4N							

Figure 7.1.5.24-1: GPS ASCII Field

Rule 7.1.5.24-2: The GPS/INS Manufacturer OUI field **shall** follow the rules of the corresponding Formatted GPS Geolocation fields, given in Section 7.1.5.19.

Rule 7.1.5.24-3: 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 7.1.5.24-1: The total number of words in the GPS ASCII field will be the number contained in the "Number of Words" subfield plus two.

Rule 7.1.5.24-4: The GPS ASCII Sentence subfield (words 3…N+2 in Figure 7.1.5.24-1) **shall** only contain complete ASCII sentences, such as defined in NMEA-0183 or other valid GPS output format**.**

Permission 7.1.5.24-1: Multiple ASCII sentences **may** be concatenated and sent in a single Context packet.

Rule 7.1.5.24-5: 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.

7.1.5.25 The Context Association Lists Section

The Context Association Lists Section in an IF Context Packet Class contains four types of Context Association Lists. Each Context Association List supports associating a slightly different kind of metadata with the Described Signal. A Context Packet Stream is associated with another Context Packet Stream by including its Stream ID in one of the Context Association Lists of the other Context Packet Stream. In addition, the Context Association Lists Section contains a Channel Tag list for support of asynchronous Channels, as described in Section 7.1.5.25.4. See Appendices B.11, B.12, B.13, and B.14 for examples using the four types of Context Association Lists.

Figure 7.1.5.25-1 shows the organization of the Context Association Lists Section of the Context packet. The first two 32-bit words contain the sizes of the lists. The lists themselves follow the two words containing the list sizes.

Word	31–25	24–16	15	14–9	8–0
1	Reserved	Source List Size	Reserved		System List Size
2	Vector-Component List Size		A	Asynchronous-Channel List Size	
⋮	Source Context Association List				
⋮	System Context Association List				
⋮	Vector-Component Context Association List				
⋮	Asynchronous-Channel Context Association List				
⋮	Asynchronous-Channel Tag List (optional)				

Figure 7.1.5.25-1: The Format of the Context Association Lists Section

Rule 7.1.5.25-1: The organization of the Context Association Lists Section **shall** be as shown in Figure 7.1.5.25-1.

Rule 7.1.5.25-2: The Source List Size field **shall** contain a number from zero to 511 containing the number of Stream IDs in the Source Context Association List.

Rule 7.1.5.25-3: The System List Size field **shall** contain a number from zero to 511 containing the number of Stream IDs in the System Context Association List.

Rule 7.1.5.25-4: The Vector-component List Size field **shall** contain a number from zero to 65,535 containing the number of Stream IDs in the Vector-component Context Association List.

Rule 7.1.5.25-5: The Asynchronous-Channel List Size field **shall** contain a number from zero to 32.767 containing the number of Stream IDs in the Asynchronous-Channel Context Association List.

Rule 7.1.5.25-6: When the “A” bit is set to zero, the Asynchronous-Channel Tag List **shall** be omitted from Context Association Lists Section. When the “A” bit is set to one, the Asynchronous-Channel Tag List **shall** be included in the Context Association Lists Section, and the size of the Asynchronous-Channel Tag List **shall** be the same as the size of the Asynchronous-Channel Context Association List, given by the Asynchronous-Channel List Size field.

Rule 7.1.5.25-7: If the value in the List Size field is zero, then the corresponding List(s) **shall** be omitted.

Rule 7.1.5.25-8: The “Reserved” bits shown in Figure 7.1.5.25-1 **shall** be set to zero.

Rule 7.1.5.25-9: Any Context Packet Stream to be associated with a Data Packet Stream **shall** be either paired with or indirectly associated with the Data Packet Stream. Data-Context pairing is governed by Rule 7.1.2-3. If a

Context Packet Stream is indirectly associated with a Data Packet Stream, its Stream ID **shall** be included in a Context Association List of another Context Packet Stream that is itself either paired with or indirectly associated with the Data Packet Stream.

Observation 7.1.5.25-1: The Context Association List method does not inherently limit the length of the chain of Context associations.

Observation 7.1.5.25-2: 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 Context Association Lists present, has been received. The time required for full communication of the associations is therefore dependent on the frequency with which the Context Association Lists are included in the emitted Context packets.

Permission 7.1.5.25-1: Context Association Lists **may** be used to associate multiple Context Packet Streams without any of them being associated to a Data Packet Stream.

7.1.5.25.1 The Source Context 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 Source Context Association List provides this capability. See Appendix B.11 for an example using this list.

Rule 7.1.5.25.1-1: The Source Context Association List **shall** only be used to associate Context for input Data and/or immediately upstream processes. Specifically, when an IF Context Packet Stream is associated with a process, or with Data emitted by that process, that Context Packet Stream's Source Context Association List **shall** only list those Context Packet Streams associated with the Data being input to that process, or with the source(s) generating the input Data.

Observation 7.1.5.25.1-1: When an IF Context Packet Stream is paired with a Data Packet Stream it is typically the case that the resulting Context packets contain metadata relating to the process generating the Data Packet Stream, as well as metadata explaining the Data itself. In such cases, chains of associated Context Packet Streams may simultaneously pertain both to a chain of Data Packet Streams and to a chain of processes generating those streams.

Rule 7.1.5.25.1-2: Whenever metadata regarding an immediate upstream process affecting a Data Packet Stream is to be conveyed via a Context Packet Stream, the Context Packet Stream of the upstream process **shall** be associated with the Context Packet Stream that is paired with the Data Packet Stream. The two Context Packet Streams **shall** be associated using the Source Context Association List, with the Stream ID of the upstream Context Packet Stream appearing in the Source Context Association List of the downstream Context Packet Stream. This **shall** hold except when the metadata is related to Vector components, in which case the Vector-components Context Association List **shall** be used.

Observation 7.1.5.25.1-2: An entire processing architecture may be represented by a collection of IF Context Packet Streams that are associated via Source Context Association Lists in a way that reflects the signal flow.

7.1.5.25.2 The System Context Association List

The System Context Association List facilitates association with a Data Packet Stream additional metadata that is not related to vector components, asynchronous Channels, 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 paired IF Context packet may contain a Stream ID in the System Context Association List that associates 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. See Appendix B.12 for an example using this list.

Rule 7.1.5.25.2-1: The System Context Association List **shall not** be used to associate in Context Packet Streams that can be correctly associated and interpreted via the Source Context Association List, the Asynchronous Channel Context Association List, or the Vector-component Context Association List.

7.1.5.25.3 The Vector-component Context Association List

When a Data Packet Stream contains Sample Vectors, each component of these vectors may have metadata associated with it. The Vector-component Context 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 Context 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. See Appendix B.13 for an example using this list.

Rule 7.1.5.25.3-1: When the paired Data Packet Stream contains a Sample Vector, the Vector-component Context Association List, when present, **shall** have exactly *N* entries, where *N* is the dimension of the Sample Vector.

Rule 7.1.5.25.3-2: The ordering of the Stream IDs in the Vector-component Context Association List **shall** match the ordering of vector components in the paired Data Packet Stream.

Rule 7.1.5.25.3-3: The Vector-component Context Association List **shall not** contain any Stream IDs other than the Stream IDs required to associate metadata with each vector component.

7.1.5.25.4 The Asynchronous-Channel Context Association List

When Channel Tags are used in the paired Data Packet Stream, a Context Packet Stream can be associated with each Channel using the Asynchronous-Channel Context Association List and the optional Asynchronous Channel Tag List. The Asynchronous-Channel Context Association List contains an ordered list of the Stream IDs of the Context Packet Streams to be associated with the Channels. When included, The Asynchronous-Channel Tag List contains an ordered list of all the Channel Tags used in the paired Data Packet Stream.

The presence of the Asynchronous-Channel Tag List is indicated by the “A” bit in the second word of the Context Association Lists Section. When included, the Asynchronous-Channel Tag List has the same size as the Asynchronous-Channel Context Association List, given by the Asynchronous-Channel List Size field. Whether or not the Asynchronous-Channel Tag List is included in the Context Packet Stream, the equivalent information needs to be included in the Information Class documentation. See Appendix B.14 for an example using these lists.

Rule 7.1.5.25.4-1: When the paired Data Packet Stream uses Channel Tags, the Asynchronous-Channel Context Association List, when present, **shall** have exactly *N* entries, where *N* is the number of channels in the paired Data Packet Stream.

Rule 7.1.5.25.4-2: The Asynchronous-Channel Context Association List **shall not** contain any Stream IDs other than the Stream IDs required to associate metadata with each channel.

Rule 7.1.5.25.4-3: When the Asynchronous Channel Tag List is present it **shall** have the same size as the Asynchronous-Channel Context Association List.

Rule 7.1.5.25.4-4: When the Asynchronous Channel Tag List is present, the ordering of its Channel Tags **shall** match the ordering of Stream IDs in the Asynchronous-Channel Context Association List. Specifically, the Channel Tag for a given Channel **shall** be the k^{th} entry in the Asynchronous-Channel Tag List when the Stream ID of the IF Context Packet Stream for that Channel is the k^{th} entry in the Asynchronous-Channel Context 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 7.2-1 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 the same as for an IF Context Packet Class, except that the Context section is user-defined, as shown in Figure 7.2-1.

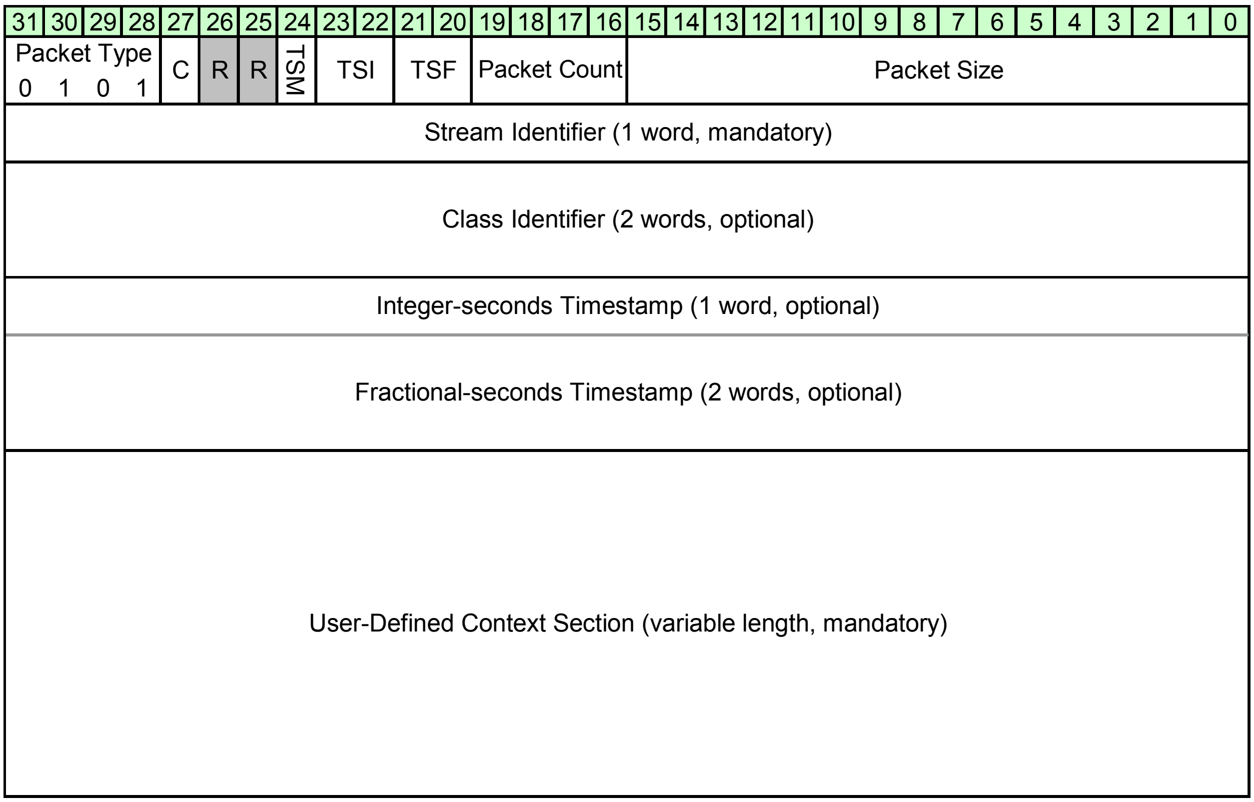


Figure 7.2-1: The Template for Extension Context Packet Classes

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

Rule 7.2-3: The Packet Type subfield of an Extension Context packet header **shall** contain the binary value "0101", as shown in Figure 7.2-1.

Rule 7.2-4: The C, TSM, TSI, TSF, Packet Count, and Packet Size subfields of an Extension Context packet header **shall** function as defined for the IF Context Packet Class.

Rule 7.2-5: The Stream Identifier, Class Identifier, and Timestamp fields of an Extension Context packet **shall** function as defined for the IF Context packet.

Recommendation 7.2-1: The Context section of an Extension Context Packet **should** be formatted to look as much like that of an IF Context Packet as possible. This facilitates the reuse of hardware and/or software.

DocumentationRequirement 7.2-1: The Context Packet Class documentation **shall** specify the format of the user-defined Context section, as well as the meaning of all the included fields.

8 Information Stream and Class Rules

This section sets forth rules governing Information Streams and Information Classes. Information Stream rules govern the allowed contents and structure of an Information Stream. An Information Class, as explained in Section 4, 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 included Packet Streams. It also specifies the purpose of the resulting Information Stream.

Section 8.1 provides rules and recommendations governing the allowed content and organization of a VRT-compliant Information Stream. Section 8.2 provides 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 8.3 presents recommended system-specific specifications. These include such items as the accuracy of various frequency and time fields, which are typically dependent upon the equipment emitting the Information Stream.

8.1 *Information Stream Rules*

An Information Stream can consist of only a single VRT Packet Stream, but in general it involves more than one. For example, an IF Data Packet Stream may be paired with an IF Context Packet Stream which provides the Context needed to fully understand the contents of the IF Data Packet Stream. A Data Packet Stream may be paired with at most one such Context Packet Stream, but other Context may be associated with the Data by means of Context Association Lists. It is possible for these associations to become rather complex. This section contains rules governing the inclusion of Packet Streams within an Information Stream and the way those Packet Streams may be associated with each other.

Rule 8.1 -1: When a VRT Information Stream contains multiple Packet Streams, every included Packet Stream **shall** be associated with at least one other Packet Stream in that Information Stream. This association **may** be a Data-Context pairing, or it may be via one of the Context Associations described in Section 4.1.3.2.8. The type of association chosen **shall** conform to the rules in Section 7.1.5.24 and in the following sections.

8.1.1 Data Packet Stream Rules

Rule 8.2-1: An Information Stream **shall** include at most one Data Packet Stream. The included Data Packet Stream **may** be an IF Data Packet Stream or an Extension Data Packet Stream.

Rule 8.1.1-2: When a Data Packet Stream is included in the Information Stream it **shall** be associated with the collection of included Context Packet Streams by pairing it with exactly one Context Packet Stream.

Rule 8.1.1-3: A Data Packet Stream **shall not** be paired with another Data Packet Stream.

Rule 8.1.1-4: A Data Packet Stream **shall not** be associated with an Information Stream as if it were Context. Correspondingly its Stream ID **shall not** appear in any Context Association List.

Permission 8.1.1-1: An Information Stream **may** contain no Data Packet Stream at all. This will occur when the Described Signal is an analog or other non-VRT data signal, or when a Context Packet Stream is created solely for the sake of conveying equipment status.

Observation 8.1.1-1: Though an analog signal or a non-VRT digital signal may be the Described Signal in a VRT Information Stream, these types of Described Signals are not considered part of the Information Stream. Thus the Information Stream can be considered compliant even when the Described Signal is not.

8.1.2 Context Packet Stream Rules

Rule 8.1.2-1: A Context Packet Stream **shall** be paired with at most one Data Packet Stream. This pairing **shall** be indicated on the link by the sharing of a Stream ID between the Data Packet Stream and the paired Context Packet Stream.

Rule 8.1.2-2: Once paired with a Data Packet Stream, a Context Packet Stream **shall not** be associated with other Context Packet Streams, either within the Information Stream or within other Information Streams, such that its Stream ID appears in a Context Association List of another Context packet.

Rule 8.1.2-3: A Context Packet Stream **shall not** be paired with another Context Packet Stream.

Observation 8.1.2-1: The pairing of a Context Packet Stream with a Data Packet Stream is always a one-to-one association. It is intended to create a simple aggregate that is often capable of conveying all required signal context without the complexity of Context Association Lists.

Additional Context Packet Streams may be appended to an Information Stream by associating them either directly or indirectly with the Context Packet Stream that is paired with the Data Packet Stream. This association is indicated in the Packet Streams by the contents of the Context Association Lists. This mechanism for conveying the associations between Packet Streams is described in Section 7.1.5.25.

Rule 8.1.2-4: Any Context Packet Streams included in the Information Stream in addition to the Context Packet Stream paired with the Data Packet Stream **shall** be associated with that Data Packet Stream via some chain of associations that can be represented via the Context Association Lists in the IF Context Packets. This chain of associations **shall** always be traceable to the paired Context Packet Stream.

Permission 8.1.2-1: A VRT Information Stream **may** contain an arbitrary number of Context Packet Streams, as long as none of the association rules in Sections 8.1.1, 8.1.2, and 7.1.5.25 are violated.

Rule 8.1.2-5: The Context Packet Stream associations within the Information Stream **shall** be reflected by the presence of the Stream Identifiers in the appropriate Context Association Lists (see Section 7.1.5.25) whenever those lists are present in the resulting emitted packets.

Permission 8.1.2-2: A Context Packet Stream not paired with the Data Packet Stream **may** be associated with multiple other Context Packet Streams in an Information Stream. Correspondingly its Stream ID **may** appear in multiple Context Association Lists within the Information Stream.

Permission 8.1.2-3: A Context Packet Stream not paired with the Data Packet Stream **may** be associated with Context Packet Streams in multiple other Information Streams. Correspondingly its Stream ID **may** appear in multiple Context Association Lists in multiple Information Streams.

Observation 8.1.2-2: Permissions 8.1.2-2 and 8.1.2-3 allow the sharing of Context both within an Information Stream and across multiple Information Streams. Only Context can be shared however, not Data.

Observation 8.1.2-3: Since Extension Context Packets do not contain Context Association Lists, they cannot be used to associate other Context Packet Streams with the Information Stream. Extension Context Packet Streams are always found at the end of a chain of associations. As a consequence, an Extension Context Packet Stream can only be paired with the Data Packet Stream if no additional Context Packet Streams are included in the Information Stream.

Permission 8.1.2-4: There are no restrictions on the rate or time at which Context packets within an Information Stream are emitted. When a Context packet is emitted, it **may** contain as many or as few of the optional fields as required.

8.2 *Information Class Specification*

As explained in Section 4, an Information Class is a specification for the creation of a VRT Information Stream. This specification has eight components, shown in Table 8.2-1.

Documentation Rule 8.2-1: A VRT Information Class **shall** specify the eight components, shown in Table 8.2-1, defining an Information Stream. These specifications shall conform to the requirements in Sections 8.2.1 through 8.2.7.

	Information Class Component	Suggested Method of Specification
1	Information Class Name and Code	Enter the name of the Information Class and the 16-bit Code assigned to it by the Company creating the Class.
2	Information Stream Purpose	Sentence or paragraph stating the purpose of the Information Stream being defined.
3	Packet Stream Names	A list of Packet Stream names, one for each Packet Stream being created
4	Packet Stream Purposes	A list of Packet Stream purposes, one for each Packet Stream being created
5	Packet Classes	A collection of Packet Classes from which the above Packet Streams are to be created.
6	Packet Stream Details	A list of any significant details related to each Packet Stream which are not specified in the Packet Class documentation
7	Reference Points	A description of all Context Reference Points. Include Reference point IDs when appropriate.
8	Packet Stream Associations	A diagram or outline, etc. Must be a complete and unambiguous specification of all the Packet Stream associations within the Information Stream. (See Appendices A and B for examples.)

Table 8.2-1: The Eight Components of an Information Class.

The table suggests a method of specification for each Information Class component. The actual requirements for each of these components are given in the following subsections.

8.2.1 Information Class Name

Every VRT Information Class 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).

Documentation Rule 8.2.1-1: Every VRT Information Class **shall** have a name.

Documentation Recommendation 8.2.1-1: The name given to an Information Class **should** reflect the purpose of the resulting Information Stream(s).

8.2.2 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.

Documentation Rule 8.2.2-1: A VRT Information Class **shall** specify the purpose of the resulting VRT Information Stream.

Documentation Recommendation 8.2.2-1: 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 8.2.2-1: 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 Stream in such a case might be stated as "To convey data and Context for one of *N* channels of data of type *X*."

8.2.3 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.

Documentation Rule 8.2.3-1: An Information Class **shall** specify a name for every Packet Stream within the resulting Information Stream.

Documentation Recommendation 8.2.3-1: The name given to a Packet Stream **should** reflect its purpose.

8.2.4 Packet Stream Purposes

An Information Class must include a statement of the purpose within the Information Stream of each Packet Stream it specifies. 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 generally more specific than the purpose stated within the corresponding Packet Class since the Packet Class may be used for multiple different Information Streams. Therefore, the purpose specified within the Packet Class generally only specifies the type of information conveyed rather than the particular information conveyed in the application. 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 the A1000 tuner made by the Acme Receiver Company." The Packet Stream purposes are stated as an aid to understanding the structure and uses of the resulting Information Streams.

Documentation Rule 8.2.4-1: An Information Class **shall** state the purpose of each Packet Stream included in the resulting Information Stream.

8.2.5 Packet Classes

Most of the detailed information about 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 of that type. Section 4 explains the uses of each type of Packet Class. Sections 6 and 7 provide rules governing their structure and function.

Rule 8.2.5-1: Every Packet Stream in a VRT Information Stream **shall** be generated from exactly one VRT compliant Packet Class. That class **may** be either an IF Data Packet Class, or an Extension Data Packet Class, or an IF Context Packet Class, or an Extension Context Packet Class.

Documentation Rule 8.2.5-1: A VRT Information Class **shall** include all the Packet Classes used to make Packet Streams within the resulting Information Stream.

Permission 8.2.5-1: 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.

8.2.5.1 IF Data Packet Classes

An IF Data Packet Class consists of four things:

1. A class name and code.
2. The specification of the purpose of the resulting Packet Stream(s).
3. The specification of the choice of allowed options for the Class.
4. The specification of the meaning of Channel Tags and/or Event Tags included in the Item Packing Fields, and of the meaning of any State and Event indicators used in the Trailer.

Documentation Rule 8.2.5.1-1: Every VRT IF Data Packet Class **shall** have a name and a Class code.

Documentation Recommendation 8.2.5.1-1: The name given to an IF Data Packet Class **should** reflect the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.1-2: VRT IF Data Packet Class documentation **shall** specify the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.1-3: VRT IF Data Packet Class documentation **shall** include the specification of the option chosen for each item shown in Table 8.2.5.1-1 for the IF Data Packet Class.

Documentation Recommendation 8.2.5.1-2: The specification of an IF Data Packet Class **should** be formatted as a table with entries in the same order as in Table 8.2.5.1-1.

Documentation Rule 8.2.5.1-4: VRT IF Data Packet Class documentation **shall** specify any Channel Tags included in the Item Packing Fields, along with their meaning. If a table such as Table 8.2.5.1-1 is used, this information **may** reside in the comment box next to the Channel-Tag size. If a large number of Channel Tags is used, then a separate list, or some defining rule from which that list can be created, **shall** be provided.

Documentation Rule 8.2.5.1-5: VRT IF Data Packet Class documentation **shall** specify the meaning of any Event Tags included in the Item Packing Fields. If a table such as Table 8.2.5.1-1 is used, this information **may** reside in the comment box next to the Event-Tag size.

Documentation Rule 8.2.5.1-6: VRT IF Data Packet Class documentation **shall** specify the meaning of any State and Event indicators included in the Trailer. If a table such as Table 8.2.5.1-1 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 & Code:		
Packet Stream Purpose:		
Packet Header		
Parameter	**Allowed Options**	**Comments**
Packet Type	- IF Data Stream Packet with Stream ID - IF Data Stream Packet w/o Stream ID	
Packet Size	Up to 65535 32-bit words	Specify range of sizes if applicable
Stream Identifier	Any 32-bit value, when used	Specify value if appropriate
Class ID	Used / Not used	Specify OUI, Information Class Code(s)*, Packet Class Code
Integer-seconds Timestamp	- UTC -Other - GPS - Not used	
Fractional-seconds Timestamp	- Sample Count - Free-running Counter - Real-time - Not used	Specify free-running counter modulus if 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 65,535	
Real/Complex Type	- Real - Complex Cartesian - Complex Polar	
Data Item Format	- Unsigned fixed-point - Two's-Complement Fixed Point - IEEE-754 Single-Precision Floating-point - IEEE-754 Double-Precision 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 - No Repeating	
Repeat Count	1 to 1024	
Packet Trailer		
Used/Not used		
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	Explain what is detected.
Spectral Inversion Indicator	Used/Not used	
Over-Range Indicator	Used/Not used	
Sample Loss Indicator	Used/Not used	
User-Defined Indicators(Bits 2..0)	Each of the indicators may be: - Used (indicate function) - Not used	Specify function if used
Associated Context Packet Count	Used / Not used 0 to 127 packets if used	Specify range of packets if used

Table 8.2.5.1-1: Options for an IF Data Packet Class.

* *If a Packet Class is used in multiple Information Classes, then this comment should list them all. In an actual Information Stream however, only the corresponding Information Class code will appear in the packets.*

8.2.5.2 IF Context Packet Classes

An IF Context Packet Class consists of four things:

1. A class name and code.
2. The specification of the purpose of the resulting Packet Stream(s).
3. The specification of the choice of allowed fields and options for the class.
4. The specification of the meaning of the enabled fields.

Documentation Rule 8.2.5.2-1: Every VRT IF Context Packet Class **shall** have a name.

Documentation Recommendation 8.2.5.2-1: The name given to an IF Context Packet Class **should** reflect the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.2-2: VRT IF Context Packet Class documentation **shall** specify the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.2-3: VRT IF Context Packet Class documentation **shall** specify the option chosen for each item shown in Table 8.2.5.2-1 for the IF Context Packet Class.

Documentation Recommendation 8.2.5.2-2: The IF Context Packet Class documentation **should** be formatted as a table with entries in the same order as in Table 8.2.5.2-1.

Documentation Rule 8.2.5.2-4: VRT IF Context Packet Class documentation **shall** specify the meaning of any fields that are used, as well as the allowed values in that field. If a table such as Table 8.2.5.2-1 is used, this information **may** reside in the comment box next to the option to which it relates.

Documentation Recommendation 8.2.5.2-3: For each Context Field that is included into only a subset of the transmitted Context Packets, the Packet Class documentation **should** specify the conditions under which the field is included.

Documentation Recommendation 8.2.5.2-4: The Packet Class documentation **should** specify the valid range for each Context Field used in a Context Packet Stream.

IF Context Packet Class Options Table		
Class Name & Code:		
Packet Stream Purpose:		
Packet Header		
Parameter	Options	Comments
Packet Type	IF Context Packet	
Stream Identifier	- Used (Actual 32-bit value is optional) - Not used	Specify value if appropriate
Class ID	- Used - Not used	Specify OUI, Packet Class Code, and Information Class Code if known
Integer-seconds Timestamp	- UTC - GPS - Other - Not used	
Fractional-seconds Timestamp	- Sample Count - Real-time - Free-running Counter - Not used	Specify free-running counter modulus if used
Timestamp Precision	- Sample-precision - Packet-precision	
Context Fields		
Parameter	Options	Comments
Context Field Change Indicator	Used/Not used	Give period of Context updates if applicable
Reference Point Identifier	Used/Not used	
Bandwidth	Used/Not used	Specify the useable spectrum definition
IF Reference Frequency	Used/Not used	
RF Reference Frequency	Used/Not used	
RF Reference Frequency Offset	Used/Not used	
IF Band Offset	Used/Not used	
Reference Level	Used/Not used	
Gain	Used/Not used	
Over-Range Count	Used/Not used	
Sample Rate	Used/Not used	
Timestamp Adjustment	Used/Not used	
Timestamp Calibration Time	Used/Not used	
Temperature	Used/Not used	
Device ID	Used/Not used	Specify Manufacturer OUI + Device code when used.
IF Data Packet Payload Format	Used/Not used	
Logical Events	Used/Not used	
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	
Spectral Inversion Indicator	Used/Not used	
Over-Range Indicator	Used/Not used	
Sample Loss Indicator	Used/Not used	
User-Defined Indicators (Bits 2..0)	Each of the indicators may be Always used / Sometimes used / Never used	Specify function if used
User-Defined Bits [7..0]	Each bit may be: Used/Not used	Specify function of used bits
GPS	Used/Not used	
Formatted GPS Geolocation	Used/Not used	
Formatted INS Geolocation	Used/Not used	
ECEF Ephemeris	Used/Not used	
Relative Ephemeris	Used/Not used	
GPS ASCII	Used/Not used	
Ephemeris Reference Identifier	Used/Not used	
Context Association Lists	Used/Not used	
Source List	Number of Stream IDs in the list (0 to 511)	
System List	Number of Stream IDs in the list (0 to 511)	
Vector Component List	Number of Stream IDs in the list (0 to 65.535)	

Asynchronous Channel List	Number of Stream IDs and Channel Tags in the lists (0 to 32,767)	

Table 8.2.5.2-1: Options for a Standard Context Packet Class.

8.2.5.3 Extension Data Packet Classes

Extension Data Packet Classes have additional documentation requirements compared to 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 four things:

1. The class name and code.
2. The specification of the purpose of the resulting Packet Stream(s).
3. The specification of the organization of the payload of the packet. This includes the location and format of each field in the payload.
4. The meaning of each field in the payload section of the packet.

Documentation Rule 8.2.5.3-1: Every VRT Extension Data Packet Class **shall** be assigned a name and class code.

Documentation Recommendation 8.2.5.3-1: The name given to an Extension Data Packet Class **should** reflect the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.3-2: Extension Data Packet Class documentation **shall** specify the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.3-3: Extension Data Packet Class documentation **shall** specify the meaning of any user-defined bits that are utilized in the header.

Documentation Rule 8.2.5.3-4: Data Packet Class documentation **shall** specify the location and format of each field in the payload.

Documentation Permission 8.2.5.3-1: Extension Data Packet Class documentation **may** be given in any format that clearly conveys the location of all the fields.

Documentation Permission 8.2.5.3-2: The specification of the field formats in Extension Data Packet Class documentation **may** be given in any format that clearly conveys these formats. Different methods **may** be used for each field.

Documentation Rule 8.2.5.3-5: Extension Data Packet Class documentation **shall** specify the meaning each field in the payload section of the packet.

Documentation Recommendation 8.2.5.3-1: An Extension Data Packet Class documentation **should** be formatted as much like an IF Data Packet Class as possible, to facilitate the reuse of hardware and/or software.

8.2.5.4 Extension Context Packet Classes

Extension Context Packet Classes have additional documentation requirements compared to 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 four things:

1. The class name.
2. The specification of the purpose of the resulting Packet Stream(s).

3. The specification of the organization of the payload of the packet. This includes the location and format of each field in the payload.
4. The meaning of each field in the payload section of the packet.

Documentation Rule 8.2.5.4-1: Every VRT Extension Context Packet Class **shall** be assigned a name and class code.

Documentation Recommendation 8.2.5.4-1: The name given to an Extension Context Packet Class **should** reflect the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.4-2: Extension Context Packet Class documentation **shall** specify the purpose of the resulting Packet Stream(s).

Documentation Rule 8.2.5.4-3: Extension Context Packet Class documentation **shall** include the specification meaning of any user-defined bits that are utilized in the header.

Documentation Rule 8.2.5.4-4: Extension Context Packet Class documentation **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.

Documentation Permission 8.2.5.4-1: Extension Context Packet Class documentation **may** be given in any format that clearly conveys the packet organization.

Documentation Permission 8.2.5.4-2: The specification of the field formats in Extension Context Packet Class documentation **may** be given in any format that clearly conveys these formats. Different methods **may** be used for each field.

Documentation Rule 8.2.5.4-5: Extension Context Packet Class documentation **shall** specify the meaning of each field in the payload section of the packet.

8.2.6 Packet Stream Details

Although a Packet Class fully defines the structure of packets to be used in a Packet Stream, the definition of the function of the Packet Stream is typically somewhat general. This facilitates reuse in multiple Information Streams. The more precise details need to be provided by the Information Class, since these details are typically specific to an application, as is the Information Class created for that application. Some examples of Packet Stream specifics are:

- Packet transmission rate
- Physical meaning, i.e. scale, of Data Items in Data packets (may also be included in Context packets)
- Conditions under which Context packets will be transmitted
- Conditions under which optional fields will be included in packets

In addition to these, any other Packet Stream details important to the application and not specified by the Packet Class need to be specified in this part of the Information Class.

Documentation Rule 8.2.6-1: In addition to the required documentation, a VRT Information Class **shall** specify the various Packet Stream details needed to complete the description of the Information Stream.

Observation 8.2.6-1: In some simple self-contained systems the Stream IDs can be specified in the Information Class. This should only be done when there is no possibility that combining the system with other systems will lead to the duplication of Stream IDs across the aggregate system.

8.2.7 Reference Points

As explained in Section 4.1.3.2.7, Context packets typically convey Context information related to some Reference Point upstream in the system. In order for this information to be properly understood, the exact point in the system must be specified.

Documentation Rule 8.2.7-1: For any Context Packet Stream conveying information related to one or more Reference Points, the location of those Reference Points in the system **shall** be specified in the Information Class.

8.2.8 Packet Stream Associations

This section sets forth rules governing the documentation of Packet Stream associations within an Information Stream. Every VRT compliant Information Class must provide a specification of the Packet Stream associations in the resulting Information Stream. The association specification may take, for example, the form of a diagram or an outline-like text description. Appendix A presents an example of each of these methods. Any method of specification may be used, as long as it is complete and unambiguous.

Documentation Rule 8.2.8-1: The Information Class for a VRT Information Stream **shall** specify all the associations between the Packet Streams that make up the resulting Information Stream.

Permission 8.2.8-1: The specification of Packet Stream associations within an Information Stream **may** take any form that is complete and unambiguous.

8.3 *Recommended System-specific Specifications*

This specification provides for a variety of equipment-centric Context to be communicated either in IF Context packets or in Extension Context packets. This Context may be of little value if the accuracy of the values conveyed is not documented. For example, when the RF Reference Frequency is sent in an IF Context packet the receiver of that packet may need to know the maximum error between that value and the actual center frequency.

Table 8.3-1 contains a list of recommended specifications related to the accuracy of certain fields conveyed in IF Context packets. The values provided will depend on the performance of the specific equipment used in the system. This information need not be provided if the corresponding field is not used.

Documentation Recommendation 8.3-1: For each IF Context Packet Stream in a VRT Information Stream, the Information Class **should** specify each item shown in Table 8.3-1 if the corresponding field is used. 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 **should** provide as a specification any response that accurately communicates that fact.

Parameter	Explanation
Timestamp 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.
RF Reference Frequency Accuracy	
RF Reference Frequency Offset Accuracy	
Bandwidth 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 8.3-1: Recommended Accuracy Specifications for IF Context Fields.
The specification of the accuracy is recommended when the corresponding field is used.

In addition to the standard Context that can be communicated in an IF Context Packet Stream a wide variety of other types of Context can be communicated in one or more Extension Context Packet Streams. It is recommended that the accuracy of these fields, when the concept applies, be specified in the VRT specification. Some examples of Extension Context for which an accuracy specification would be useful are:

- GPS location, velocity, etc.
- Signal direction of arrival.
- Signal time of arrival.
- Signal-to-Noise Ratio (SNR) estimates.

Finally there is a third category of recommended system-specific specifications. These include static system details and equipment performance specifications that are not communicated via Context Packets Streams in any way, but which affect system performance. Examples of these include:

- Receiver noise figure.
- Receiver tuning range.
- Equipment operating temperature range.
- Maximum required power.

The specification of this type of information as part of the VRT Information Class is suggested whenever the application is tailored around a specific set of equipment. The list of specifications such as these could be quite long, especially in a large system. However, since these specifications are a standard part of typical equipment documentation, this recommendation is usually easily accommodated by simply including the equipment datasheets into the Information Class when appropriate.

Appendix A An Information Stream Specification Example

This appendix illustrates how to specify the eight components of an Information Class for one example system using the VRT protocol. Figure A-1 shows the example system. 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 within the system.

The tuner emits 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 GPS data and system health, i.e. the power supply voltages of the modules and their temperatures. It formats this information into VRT packets and sends them to the Ethernet interface.

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

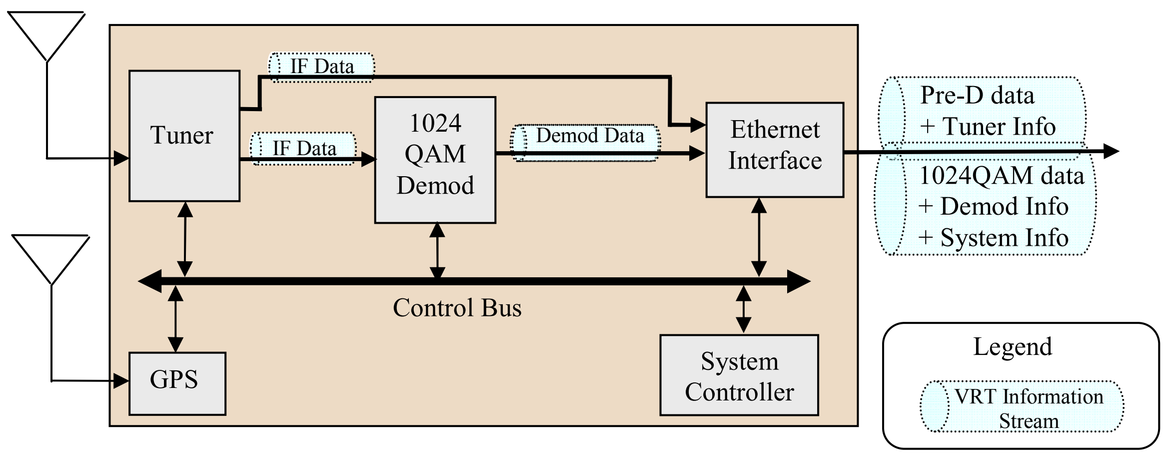


Figure A-1: An Example System Emitting Two VRT Information Streams over an Ethernet Link

The Ethernet interface receives both the IF Data Information Stream from the tuner and the demodulated data Information Stream from the demodulator. The Ethernet interface creates two new 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.

Table 8.1 may be adapted to form a checklist of the eight items required to specify each of the two Information Streams output from the above example system. The resulting tables, A-1 and A-2, list these components and contain most of the specifications:

	Information Class Component	Information Class Component Specification
1	Class Name/Code	"Pre-D" (Pre-detected signal), Class Code 7
2	Information Stream Purpose	"To convey digitized IF and related Context from the 1024QAM system"
3	Packet Stream Names	1. "Pre-D" 2. "Tuner Info"
4	Packet Stream Purposes	1. "To convey the pre-detected signal in the digitized IF from the 1024QAM system" 2. "To convey the Context related to the pre-D signal"
5	Packet Classes	1. "Pre-D" IF Data Packet Class (see below) 2. "Tuner Context" IF Context Packet Class (see below)
6	Packet Stream Details	1. "Pre-D" packet rate is 14,954 packets per second 2. A "Tuner Info" packet is transmitted whenever any Context change occurs, or once per second, whichever comes first
7	Context Reference Points	The tuner input is always the reference point.
8	Packet Stream Associations	Shown separately (see below)

Table A-1: The "Pre-D" Information Class Components.

All Information Class details are provided except for Packet Class details and Packet Stream Associations.

	Information Class Component	Information Class Component Specification
1	Class Name/Code	"1024QAM Data", Class Code 8
2	Information Stream Purpose	"To convey the demodulated 10-bit symbols and related Context from the 1024QAM system"
3	Packet Stream Names	1. "1024QAM Data" 2. "Demod Info" 3. "System Info"
4	Packet Stream Purposes	1. To convey the demodulated 10-bit symbols from the 1024QAM system 2. To convey the demodulator Context from the 1024QAM system 3. To convey device temperatures and voltages from the 1024QAM system
5	Packet Classes	1. "1024QAM Data" (see below) 2. "Demod Info" (see below) 3. "System Info" (see below)
6	Packet Stream Details	1. The "1024 QAM Data" packet rate is 100 packets per second 2. A "Demod Info" packet is transmitted whenever any of its fields change, or once per second, whichever comes first. 3. A "System Info" packet is transmitted once per second.
7	Reference Points	The tuner input is always the reference point.
8	Packet Stream Associations	Shown separately (See below)

Table A-2: The "1024QAM Data" Information Class Components.

All Information Class details are provided except for Packet Class details and Packet Stream Associations.

The above two tables contain most of the specifications required for the two Information Classes. All that remains is to specify the associations between the Packet Streams and the details of each Packet Class.

A.1 Specifying Packet Stream Associations

Figure A.1-1 graphically depicts the association between the Packet Streams in these two Information Streams. The Packet Stream names and types are indicated in the figure, as are their functions. 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 Data-Context pairing, which occurs when the Data Packet Stream and Context Packet Stream share a Stream ID. There are two of these arrows, one in each Information Stream. In each case it pairs the data with the Context related to the module that produced the data.

In the figure, the “System Info” Context Packet Stream is associated with the “Demod Info” Context Packet Stream via a “System Context” association. This indirectly associates it with the “1024QAM Data” Packet Stream. The “System Context” association arrow also corresponds to the presence of the “System Info” Stream ID in the System Context Association List in the “Demod Info” Packet Stream. (See Section 7.5.1.24 for Context Association List details.)

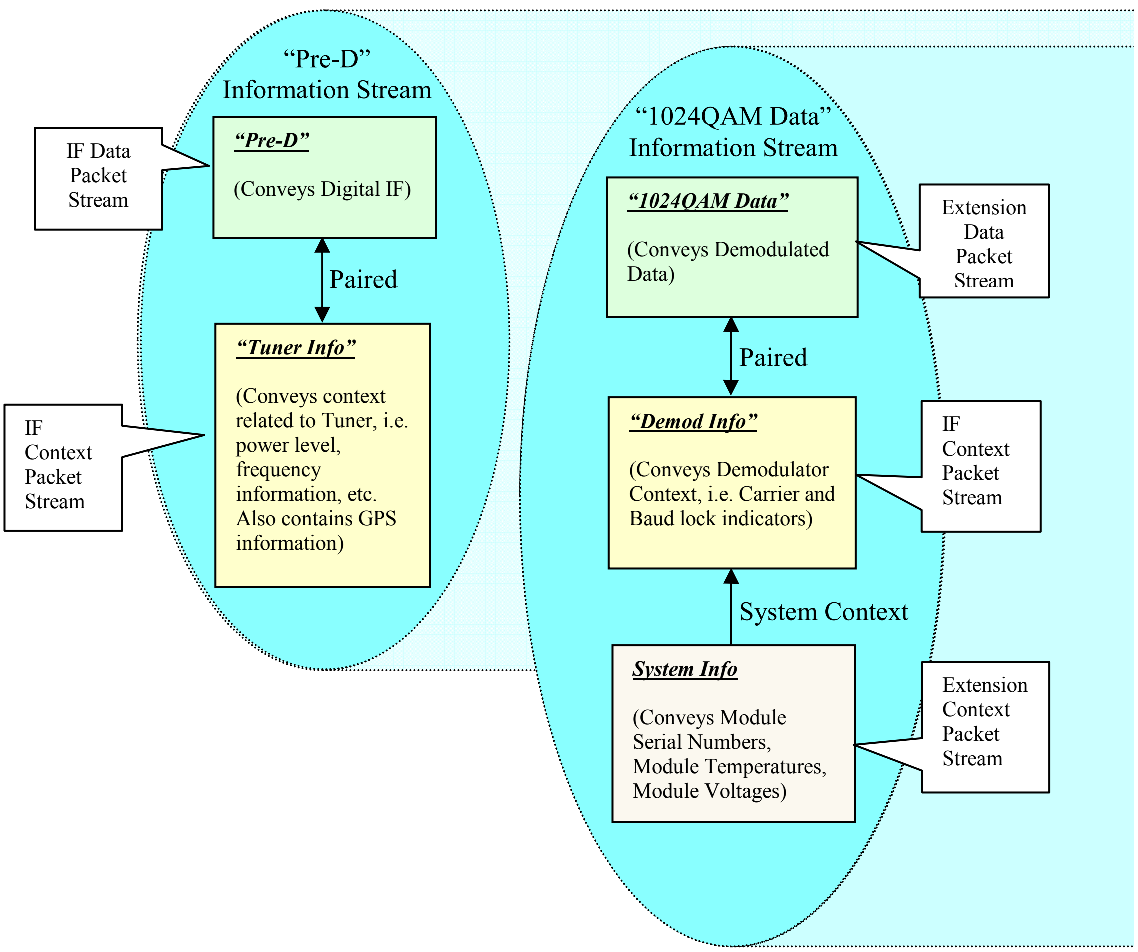


Figure A.1-1: Packet Stream Associations in the Example Information Stream. The colored tubes represent Information Streams, the boxes represent Packet Streams, and the arrows represent the associations between the Packet Streams.

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 associations.

1. Information Stream: "Pre-D"
    1.1 IF Data Packet Stream: "***Pre-D***"
    1.2 Paired IF Context Packet Stream: "***Tuner Info***"

2. Information Stream: "1024QAM Data"
    2.1 Extension Data Packet Stream: "***1024QAM Data***"
    2.2 Paired IF Context Packet Stream: "***Demod Info***"
        2.2.2 System Context:
            2.2.2.1 Extension Context Packet Stream***: "System Info"***

The outline format is more compact than a graphical representation. In some cases however the outline form could become confusing. For example if some Context Packet Stream is associated with multiple other Context Packet 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) ← (paired) → ("***Pre-D***" IF Data)

Information Stream: "1024QAM Data"
1. ("***Demod Info***" IF Context) ← (paired) → ("***1024QAM Data***" Extension Data)
2. (***"System Info"*** Extension Context) → (system) → ("***Demod Info***" IF Context)

This method works for any set of associations, but does not always lend itself to being easily pictured by the reader. So as with the other methods, it also 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 the application.

A.2 Documenting Packet Classes

The current example uses Packet Classes from all 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 each show a way to document one of the required Packet Classes.

A.2.1 *IF Data Packet Class Documentation*

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.

IF Data Packet Class		
Class Name: "Pre-D" Code: 23 (0x0017)		
Packet Stream Purpose: To convey the digitized IF from the tuner.		
Packet Header		
Parameter	**Selected Option**	**Comments**
Packet Type	IF Data packet with Stream ID	Conveys the Digitized IF.
Packet Size	2054 Words	Fixed Packet Size.
Stream Identifier	Yes	Actual code is selected at run-time
Class ID	Not Present	OUI is 0xE3199A. Info Class Code is "7" and Packet Class Code is "23," but these are not sent in the packets.
Integer-seconds Timestamp	UTC	Present in every packet
Fractional-seconds Timestamp	Real-Time	
Packet Payload		
Parameter	**Selected Option**	**Comments**
Packing Method	Link Efficient	2048-word payload holds 4681 samples plus two pad bits.
Item Packing Field Size	14 bits	
Data Item Size	14 bits	
Event-Tag Size	0 bits	No Event Tags
Channel-Tag Size	0 bits	No Channel Tags
Vector Size	0 bits	No vectors
Real/Complex Type	Real samples	
Data Item Format	Two's-Complement Fixed Point	
Sample-repeating /Channel Repeating	N/A	No repeating of any kind
Repeat Count	0	
Packet Trailer		
(Used)		
Parameter	**Selected 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	No	
Spectral Inversion Indicator	No	
Over-Range Indicator	No	
Sample Loss Indicator	No	
User-Defined Indicators(Bits 2..0)	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.

Table A.2.1-1: Documentation of the "Pre-D" IF 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.2.2 *Extension Data Packet Class Documentation*

The format of the 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 for this Packet Class. The only difference between this and an IF Data Packet Class is that the demodulated symbol stream does not conform to any of the Data Item types specifiable in an IF Data Packet Class, so the Data Item format is identified as "Nonstandard." The Table A.2.2-1 specifies the format of the "1024QAM Data" Packet Class.

Extension Data Packet Class		
Class Name: "1024QAM Data" Code: 13 (0x000D)		
Packet Stream Purpose: To convey the decoded 10-bit symbols from the demodulator.		
Packet Header		
Parameter	**Selected Option**	**Comments**
Packet Type	Extension Data packet with Stream ID	Same format as IF Data packets, but with custom Data Items.
Packet Size	326 Words	Fixed Packet Size
Stream Identifier	Yes	Actual code is selected at run-time
Class ID	Yes	OUI = 0xE3199A Info Class = 8 (0x0008) Packet Class = 13 (0x000D)
Integer-seconds Timestamp	UTC	Present in every packet
Fractional-seconds Timestamp	Real-Time	
Packet Payload		
Parameter	**Selected 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 bits	
Data Item Size	10 bits	
Event-Tag Size	0 bits	No Event Tags
Channel-Tag Size	0 bits	No Channel Tags
Vector Size	0 bits	No vectors
Real/Complex Type	Real samples	
Data Item Format	Nonstandard	10-bit symbols from the 1024 QAM constellation.
Sample-repeating /Channel Repeating	N/A	No repeating of any kind
Repeat Count	0	
Packet Trailer		
(Used)		
Parameter	**Selected 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	
Spectral Inversion Indicator	No	
Over-Range Indicator	No	
Sample Loss Indicator	No	
User-Defined Indicators(Bits 2..0)	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.

Table A.2.2-1: Documentation 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.2.3 *IF Context Packet Class Documentation*

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 Context updates related to the demodulator. Table A.2.3-1 documents the "Demod Info" Packet Class.

IF Context Packet Class		
Class Name: "Demod Info" Code: 14 (0x000E)		
Packet Stream Purpose: To convey Context related to demodulated 1024 QAM data.		
Packet Header		
Parameter	Options	Comments
Packet Type	IF Context Packet	
Stream Identifier	Yes, must be used for this packet type	Actual value is chosen at run-time
Class ID	Yes	OUI = 0xE3199A, Info Class = 8, Packet Class = 14
Integer-seconds Timestamp	UTC	Present in every packet
Fractional-seconds Timestamp	Real-time	
Timestamp Precision	Sample-precision	Context changes are indicated with near sample accuracy.
Context Fields		
Parameter	Options	Comments
Context Field Change Indicator	No	Receiver must keep track of changes.
Reference Point Identifier	No	
Bandwidth	No	Specify the useable spectrum definition
IF Reference Frequency	No	
RF Reference Frequency	No	
RF Reference Frequency Offset	No	
IF Band Offset	No	
Reference Level	No	
Gain	No	
Over-Range Count	No	
Sample Rate	Yes	Always indicates the 10-bit symbol rate of 102.4 KHz
Timestamp Adjustment	No	
Timestamp Calibration Time	No	
Temperature	No	
Device ID	Yes	Vendor OUI: 0xE3199A. Vendor Device code: 0x19, which specifies the entire system in Figure A-1.
IF Data Packet Payload Format	Yes	This field is inserted in a packet about once per second.
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
Spectral Inversion Indicator	No	
Over-Range Indicator	No	
Sample Loss Indicator	No	
User-Defined Indicators (Bits 7..0)	Bits 0, 1 used. Others unused and undefined.	Bit 1 = Carrier lock Bit 0 = Baud lock
GPS	No	No GPS feature used in these packets
Context Association Lists	Yes	
Source List	No	
System List	Yes, one entry	Links in "System Maintenance" Context
Vector Component List	No	
Asynchronous Channel List	No	

Table A.2.3-1: Documentation of the "Demod Info" IF Context Packet Class. This class is primarily for updating Context related to the demodulator. It also associates the "System Info" Context Packet Stream with the Information Stream via the System-Context Association List.

Table A.2.3-2 documents 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 Packet Stream coming from the tuner. It also contains GPS information.

IF Context Packet Class		
Class Name: "Tuner Info" Code: 15 (0x000F)		
Packet Stream Purpose: To convey Context related to the digitized IF output of the tuner.		
Packet Header		
Parameter	Options	Comments
Packet Type	IF Context Packet	
Stream Identifier	Yes, must be used for this packet type	Actual value is chosen at run-time
Class ID	Yes	OUI = 0xE3199A, Info Class = 7, Packet Class = 14
Integer-seconds Timestamp	UTC	Present in every packet
Fractional-seconds Timestamp	Real-time	
Timestamp Precision	Sample-precision	Context changes are indicated with near sample accuracy.
Context Fields		
Parameter	Options	Comments
Context Field Change Indicator	No	Give period of Context updates if applicable
Reference Point Identifier	No	Reference for this Context is always the tuner input. No identifier needed
Bandwidth	Yes	Always 30 MHz (-3 dB points)
IF Reference Frequency	Yes	
RF Reference Frequency	Yes	
RF Reference Frequency Offset	No	There is no offset
IF Band Offset	No	There is no offset
Reference Level	Yes	At the tuner input
Gain	No	
Over-Range Count	No	
Sample Rate	Yes	Always 70 MHz IF Data sample rate
Timestamp Adjustment	No	
Timestamp Calibration Time	No	
Temperature	No	
Device ID	Yes	Vendor OUI: 0xE3199A. Vendor Device code: 0x19, which specifies the entire system in Figure A-1.
IF Data Packet Payload Format	Yes	Indicates the format of the "Pre-D" packets.
Logical Events	Yes	
Calibrated Time Indicator	Yes	"1" indicates GPS lock
Valid Data Indicator	Yes	"1" indicates tuner lock AND no over-range in the current packet.
Reference Lock Indicator	Yes	
AGC/MGC Indicator	Yes	AGC mode always
Detected Signal Indicator	No	N/A
Spectral Inversion Indicator	Yes	Always indicates no inversion
Over-Range Indicator	No	
Sample Loss Indicator	No	
User-Defined Indicators (Bits 7..0)	No	Unused and undefined
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 Identifier	No	
Context Association Lists	No	

Table A.2.3-2: Documentation 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.2.4 *Extension Context Packet Class Specification*

The "***System Info***" Extension Context Packet Class is an entirely custom Context Packet Class. The fields it contains are not like those in the IF Context Packet Class, so a new packet format is required. Figure A.2.4-1 shows the format for this example Packet Class.

<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="4">Packet Type
0 1 0 1</td><td>C</td><td colspan="2">R R</td><td>T
S
M</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="4">Packet Count</td><td colspan="16">Packet Size</td></tr>
<tr><td colspan="32">Stream Identifier</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="24">OUI = 0xE3195A</td></tr>
<tr><td colspan="16">Information Class Code = 8</td><td colspan="16">Packet Class Code = 28</td></tr>
<tr><td colspan="8">Tuner Temperature</td><td colspan="8">Demodulator Temperature</td><td colspan="8">Ethernet Temperature</td><td colspan="8">System Controller Temperature</td></tr>
<tr><td colspan="8">Power Supply Temperature</td><td colspan="8">Reserved</td><td colspan="8">Exit Air Temperature</td><td colspan="8">Inlet Air Temperature</td></tr>
<tr><td colspan="16">Tuner +12 Voltage</td><td colspan="16">Tuner +5 Voltage</td></tr>
<tr><td colspan="16">Controller +3.3 Voltage</td><td colspan="16">Tuner -12 Voltage</td></tr>
<tr><td colspan="16">Demod +3.3 Voltage</td><td colspan="16">GPS +3.3 Voltage</td></tr>
<tr><td colspan="16">Ethernet +3.3 Voltage</td><td colspan="16">Ethernet +1.2 Voltage</td></tr>
</table>

Figure A.2.4-1: Format of the Extension "System Info" Context Packet Class. The OUI corresponds to the Company originating the class. Together the OUI and Packet Class Code 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 two's-complement numbers ranging from -128 Celsius to +127 Celsius.
2. All voltages are two's-complement numbers with ten bits to the right of the radix point and six to the left. This gives a range of -32 to approximately 31.999 volts and about one millivolt resolution.

With the addition of these details the documentation of this Packet Class is complete.

A.2.5 *Other Recommended System Specifications*

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

Parameter	Specification
Timestamp 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 Recommended System Specifications.
An example specification is included for each item.

Appendix B Context Field Examples

The examples in this section illustrate the use of several fields in Context Packet Classes. Each example contains a block diagram of the system being described. The blocks in these diagrams that correspond to VRT Packet Stream generators contain symbols that are a visual shorthand for VRT Packet Stream generators. This visual shorthand is explained using the following examples.

Figure B-1 shows one of the simplest VRT applications where a process, in this case an analog-to-digital converter module, generates a VRT Data Packet Stream whose payload contains the ADC samples. In this example analog input to the ADC is assigned the Stream ID 200 and the digital output of the ADC is assigned the Stream ID 300. To signify that the ADC generates a Data Packet Stream with Stream ID 300, the ADC block contains a data stream icon, a rectangle in the lower right corner, containing the number 300. In addition, the block diagram is augmented with a representation of the contents of the ADC Data Packet Stream which is connected to the VRT signal path with a dashed line. As shown in the figure, the ADC Data Packet Stream includes the Stream ID field with value 300. The analog input signal, however, does not have a Data Packet Stream, so its Stream ID is specified in the Information Class or other system documentation.

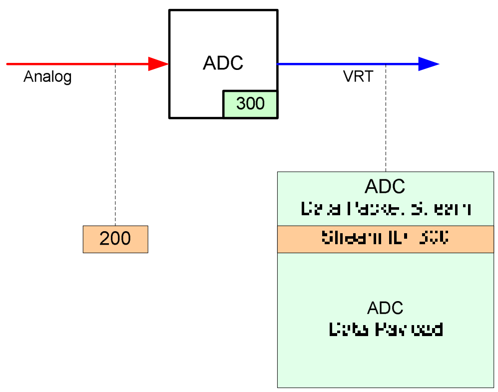


Figure B-1: Data Packet Stream Only Example

In the next example, shown in Figure B-2, the ADC module not only generates a Data Packet Stream but also a Context Packet Stream. Because the Context Packet Stream is paired with the Data Packet Stream, they share the same Stream ID 300.

As before, the ADC block contains a data stream icon (the rectangle in the lower right corner) to signify it generates a Data Packet Stream. In addition, the ADC block has a Context stream icon, a rectangle in the upper right corner, to signify that is also generates a Context Packet Stream. Both the Context and data stream icons are labeled with the Stream ID 300 to indicate that they are paired Packet Streams. Again, the block diagram is augmented with a representation of the contents of both the Data and Context Packet Streams. In this case the paired Context Packet Stream contains the Reference Level, Sample Rate, and Payload Format Context fields. In addition, the Context Packet Stream contains a Reference Point ID field which contains 200, the Stream ID of the ADC input. Therefore, for the ADC Context Packet Stream, the Described Signal is the ADC output and the Reference Point is the ADC input. This means that the Reference Level field describes the level of a signal at the ADC input that would produce a Unit-Scale Sinusoid at the ADC output.

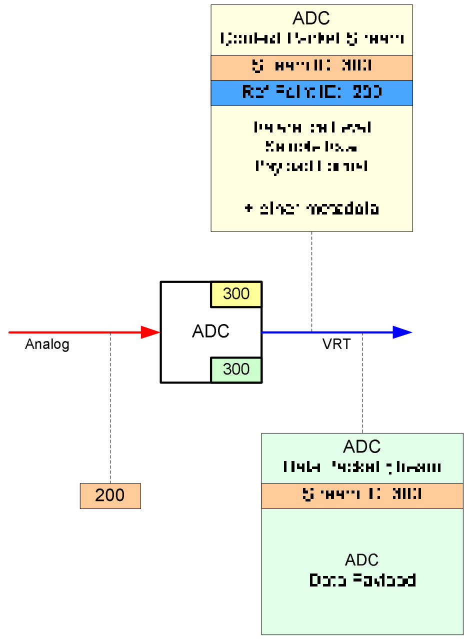


Figure B-2: Paired Data and Context Packet Stream Example

Now consider the downconverter shown in Figure B-3. The downconverter converts the analog RF signal from an antenna to a lower intermediate frequency (IF). The RFsignal is assigned Stream ID 100, and the IF signal is assigned Stream ID 200. The downconverter module has a digital interface over which it sends VRT Context Packet Streams conveying metadata about the analog IF signal. As in the previous example, the presence of the Context Packet Stream is indicated by the Context stream icon in the upper right corner of the downconverter block with the Stream ID 200. The absence of a Data Packet Stream is inferred from the absence of the data stream icon in the lower right corner.

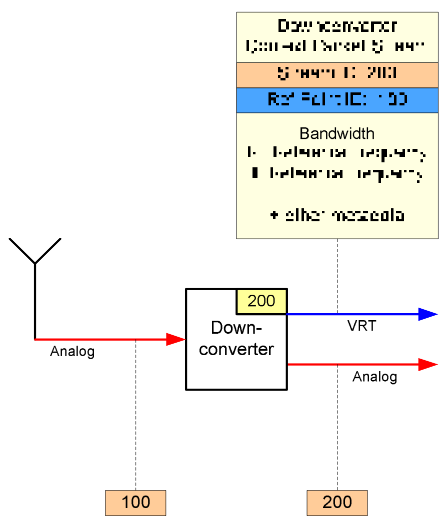


Figure B-3: Context Packet Stream Only Example

The downconverter and ADC modules from the above examples can be cascaded into the simple VRT system shown in Figure B-4. In this example the diagram indicates that the downconverter's Context Packet Stream with

Stream ID 200 is present on the interface at the output of both the downconverter and the ADC. The Data and Context Packet Streams of the ADC are only present at the output interface of the ADC.

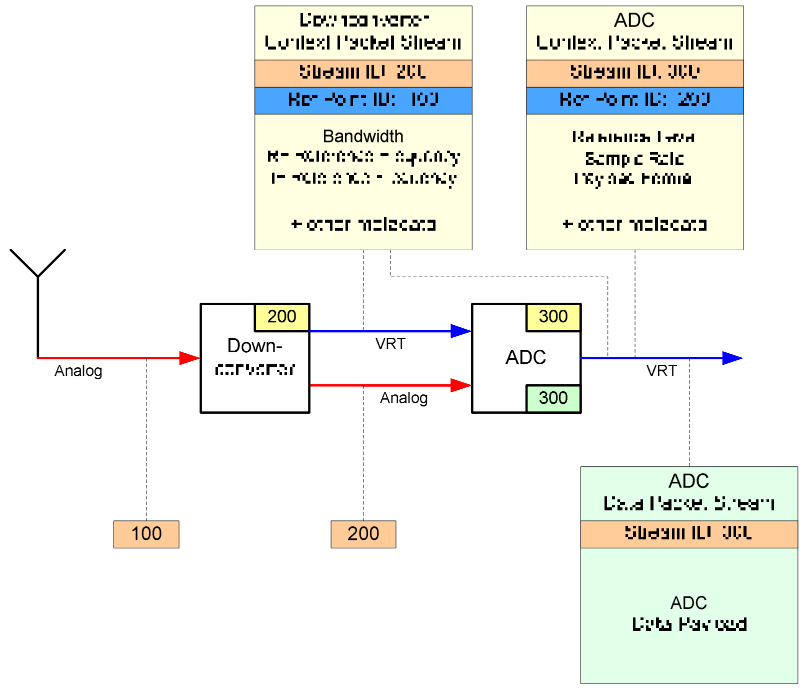


Figure B-4: Packet Streams of Cascaded Components Example

If the downconverter and ADC were considered components encapsulated in a single receiver module then the same system could be described with the block diagram shown in Figure B-5, where the receiver is shown generating a Context Packet Stream with Stream ID 200 as well as Data and Context Streams that are paired with Stream ID 300. The Data and Context stream icons could be attached to the individual components or to the encompassing receiver block. In this figure the stream icons are shown in the corners of the receiver block

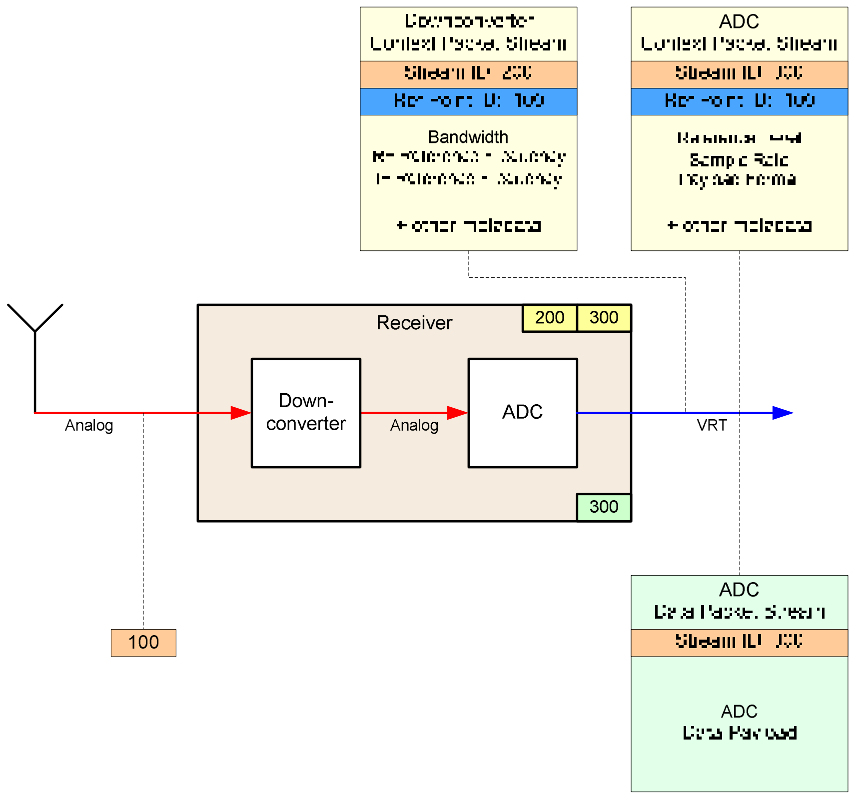


Figure B-5: Receiver Packet Streams Example

In addition, the provider of the receiver module may desire to provide only a single Data Packet Stream and paired Context Packet Stream as shown in Figure B-6. In this case the Context Packet Stream contains metadata about the entire receiver including metadata about the downconverter and ADC. The paired Packet Streams are shown to have Stream ID 200.

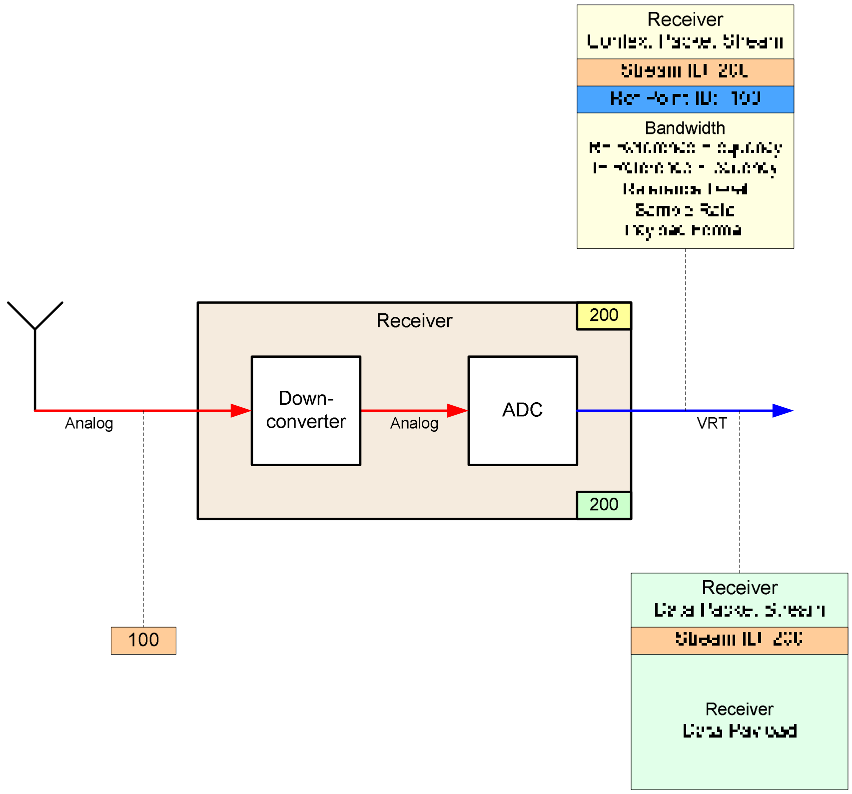


Figure B-6: Consolidated Receiver Packet Streams Example

B.1 *Reference Point Identifier Example*

Figure B.1-1 shows a block diagram of a VRT system that uses the Reference Point Identifier field. In this example there are three processes: a microwave tuner, a digital receiver, and a DDC. The signal at the antenna is assigned Stream ID 100 and the analog output of the microwave tuner is assigned Stream ID 200. The microwave tuner also has a digital interface over which it sends Context packets with Stream ID 200. The digital receiver and DDC blocks each generate paired Data and Context Packet Streams with Stream Identifiers 300 and 400, respectively.

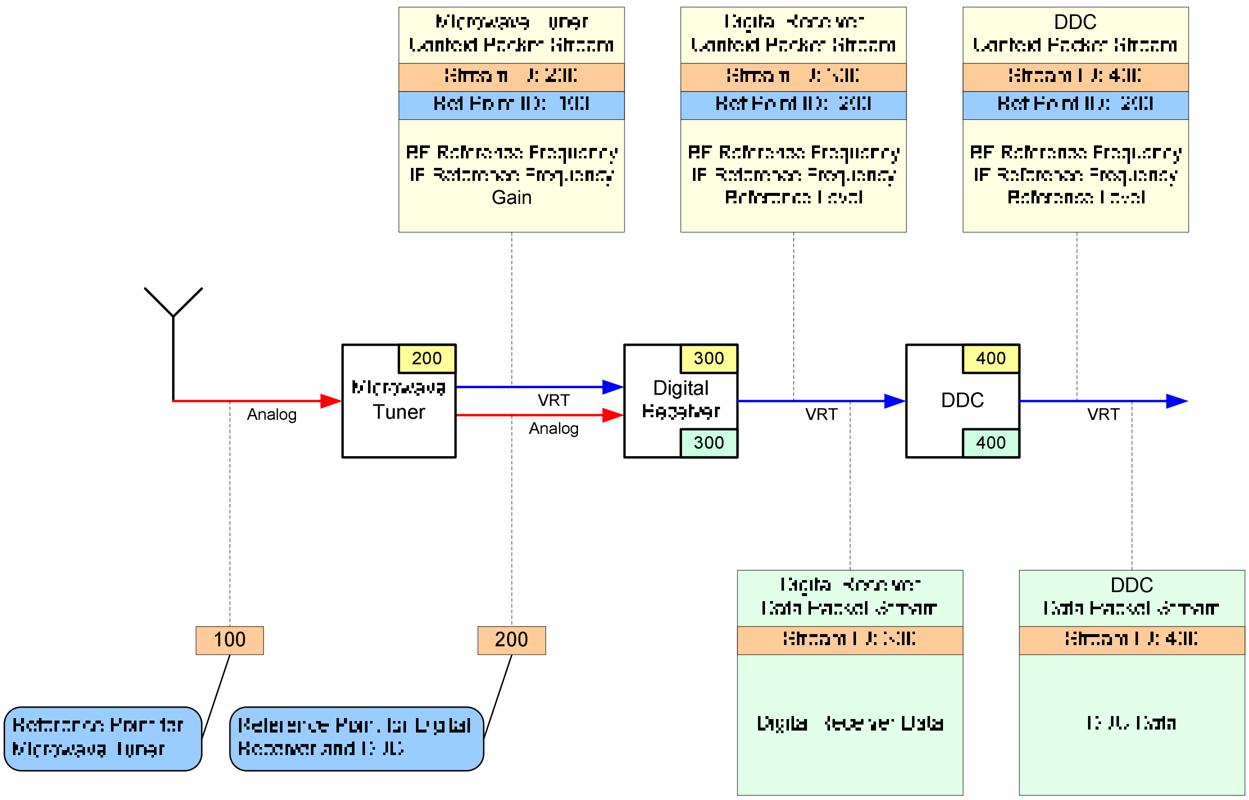


Figure B.1-1: Reference Point Identifier Example

The Reference Point for the microwave tuner is the antenna output.

The Reference Point for the digital receiver and DDC is the microwave tuner output.

Each of the Context Packet Streams contains a Reference Point ID field that specifies the Reference Point for that stream's Context. In this example the microwave tuner has a Reference Point ID of 100, which specifies the antenna output as its Reference Point. Both the Receiver and DDC Context Packet Streams include a Reference Point IDs of 200, which specifies that the Reference Point for both Receiver and DDC Context is at the microwave tuner output.

With these Reference Points established, the RF Reference Frequency field in the microwave tuner's Context Packet Stream indicates the frequency *at the antenna output* that was translated to its IF Reference Frequency. Likewise, the RF Reference Frequency fields in the digital receiver and DDC Context Packet Streams indicate the frequency *at the output to the microwave tuner* that was translated to their respective IF Reference Frequencies. Similarly, the Reference Level field in the DDC's Context Packet Stream indicates the power of a single tone *at the output of the microwave tuner* that creates a Unit-Scale Sinusoid in the payload of the DDC's Data packet.

B.2 *Spectral Fields Example*

This section presents an example using the Bandwidth, IF Reference Frequency, RF Reference Frequency, and Spectral Inversion Indicator fields. This example also illustrates how Context information can relate to analog signals. Subsequent examples illustrate how these same fields relate to digital signals.

The block diagram for this example contains a single process, the analog tuner shown in Figure B.2-1. The tuner translates spectral energy from a high frequency RF band to a lower frequency IF band.

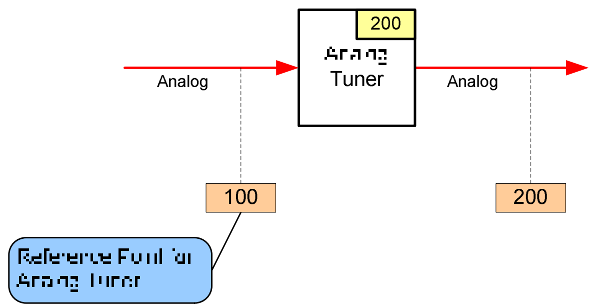


Figure B.2-1: Block Diagram for Spectral Fields Example

The analog input and output of the tuner are identified with Stream IDs of 100 and 200, respectively, and the tuner generates an IF Context Packet Stream with Stream ID 200. The Context packet contains the Reference Point Identifier field which has value 100 to indicate that the Reference Point is at the input to the tuner. The Context packet also contains Bandwidth, IF Reference Frequency, and RF Reference Frequency fields. The Band Frequency Offset and RF Reference Frequency Offset fields are not present in the Context Packet Stream (examples using these two fields are given in subsequent sections).

Figure B.2-2 shows the RF spectrum at the tuner input (the Reference Point) and the IF spectrum at the tuner output (the Described Signal). In this example the Bandwidth field contains the 3dB bandwidth at the tuner output. The IF center frequency, which is the nominal center of the output band the tuner, is marked with a wide bar in the figure. The IF Reference Frequency field in the Context packet specifies this value. The frequency at the tuner input that translates to the IF Reference Frequency is specified by the RF Reference Frequency field. Assuming the spectrum is not inverted from the RF to the IF band, the Spectral Inversion Indicator bit in the State and Event Indicator field is zero.

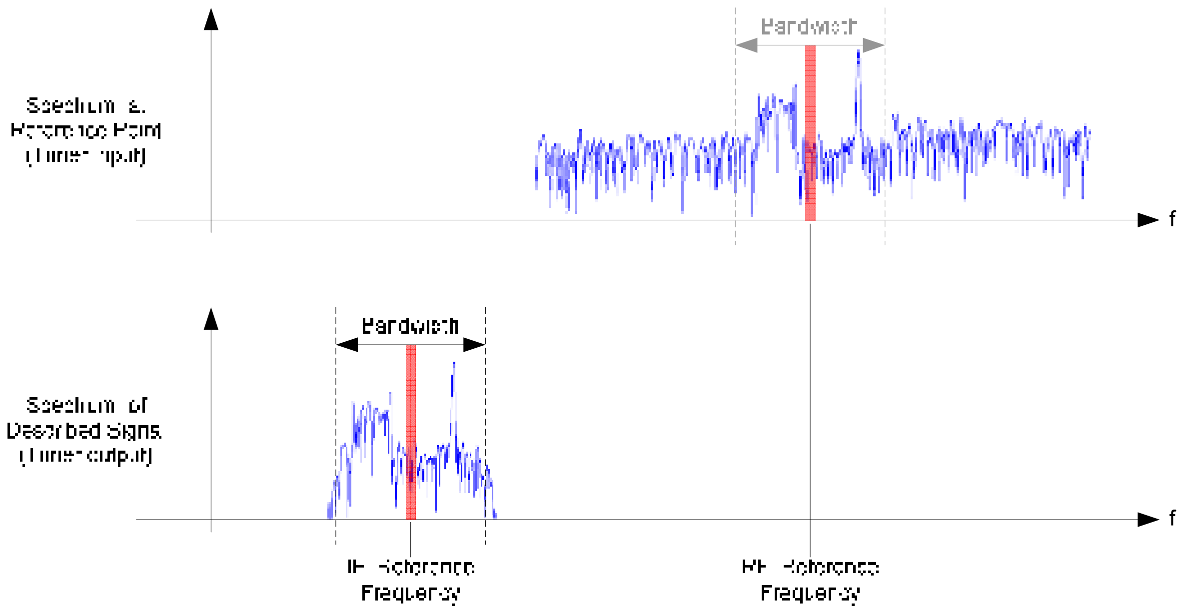


Figure B.2-2: The Spectrum of the Analog Signals at the Input and Output of the Tuner

The Bandwidth field gives the bandwidth of the output signal. The IF Reference Frequency marks the center of the output band and the RF Reference Frequency marks the frequency that was translated to the IF Reference Frequency. The bars that mark the Reference Frequencies are shown as thick red lines so that they are easily visible. In practice, they each specify a precise frequency, not a broad range of frequencies.

If the tuner were tuned to 2 GHz and downconverted a 30 MHz band to a 70 MHz IF without inverting its spectrum then the spectral fields contains the values shown in Figure B.2-3.

Analog Tuner Context Packet Stream	
Stream ID	200
Reference Point ID	100
Bandwidth	30 MHz
IF Reference Frequency	70 MHz
RF Reference Frequency	2 GHz
Spectral Inversion Indicator	0

Figure B.2-3: Context Packet Stream Contents for Spectral Fields Example.

B.3 *RF Reference Frequency Offset Example*

The example in this section expands upon the example given in Appendix B.2 to illustrate how the RF Reference Frequency Offset field is used to efficiently convey frequency translation information in channelized systems. This field is typically used with channelizers where a large number of narrowband signals are created from portions of a wideband input signal. Figure B.3-1 shows such a system where a digital receiver is followed by a bank of 64 DDCs. In this example the digital receiver is dynamically tuned across a spectral region of interest. The DDCs are configured to examine portions of the receiver's output band in greater detail. The tuning frequency of each DDC channel remains fixed as the receiver's tuning frequency is changed.

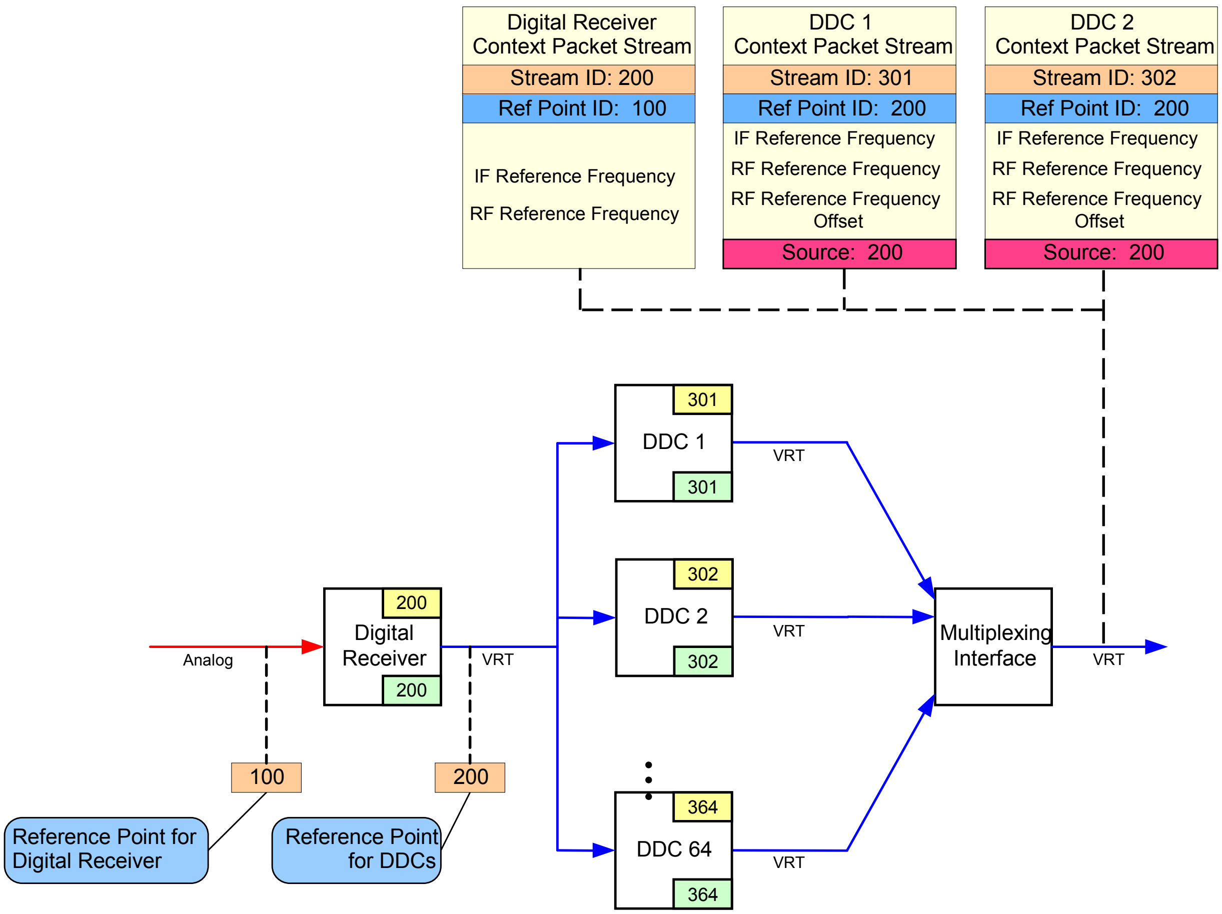


Figure B.3-1: Block Diagram for RF Reference Frequency Offset Example

The RF Reference Frequency Offset field is typically used with channelizers

All three Context Packet Streams generated in this system are output from the system as part of the output Information Stream. The Context Packet Stream for the digital receiver contains a Reference Point ID that specifies the input to the receiver as the Reference Point. The Receiver's Context Packet Stream also contains entries for IF Reference Frequency and RF Reference Frequency explained further in Appendix B.2.

The DDC Context Packet Streams each contain a Reference Point ID and a Source Context Association List. Each Reference Point ID containt Stream ID 200, which specifies the output of the receiver as its Reference Point. In addition, the entry in each of the Source Context Association Lists contains Stream ID 200, which denotes that the receiver is the input source for each of the DDCs. Each DDC's Context Packet Stream also contains entries for IF Reference Frequency, RF Reference Frequency, and RF Reference Frequency Offset. *

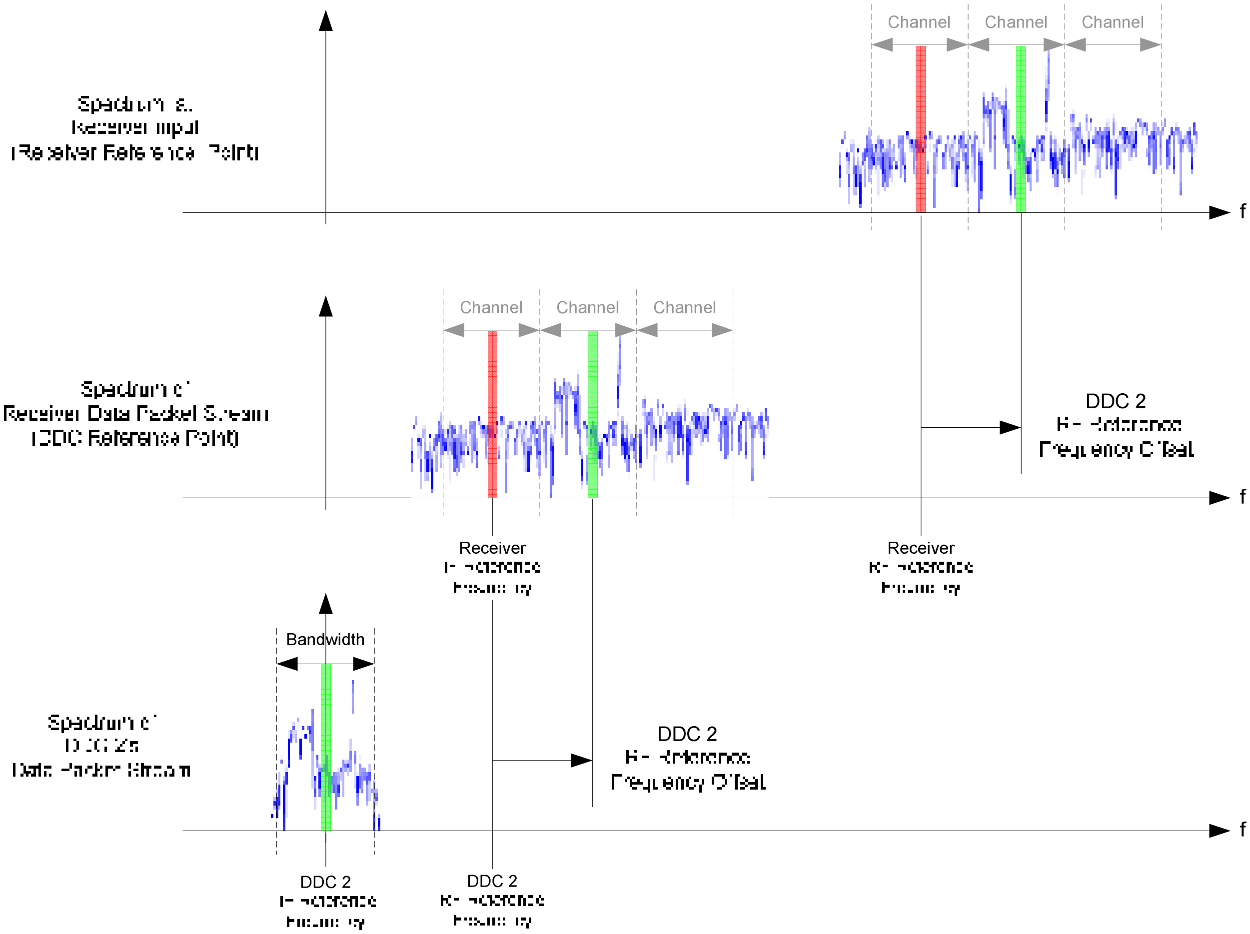


Figure B.3-2: Original and translated spectra for a single DDC
The leftmost and rightmost (red and green, respectively) bars indicate which frequencies get translated to the IF Reference Frequency for DDC1 and DDC2, respectively.

Figure B.3-2 illustrates how the DDC's RF Reference Frequency Offset field is used. The original spectrum at the input to the digital receiver is shown in the top axis. The bands for several DDC channels are shown overlaid upon the RF spectrum. In addition, two bars are shown. The leftmost (red) bar is shown at the receiver's RF Reference Frequency. This is the frequency that gets converted to the receiver's IF Reference Frequency. The rightmost (green) bar is shown at the frequency that gets translated to DDC2's IF Reference Frequency. The calculation to determine this original frequency is discussed shortly.

The middle axis shows the spectrum of the receiver's Data Packet Stream. The same channel bands and colored bars are shown with this translated spectrum. The bottom axis shows the spectrum of DDC2's Data Packet Stream. The original frequency marked by the rightmost (green) bar is translated to baseband and the spectrum for all other channels, including the one marked by the leftmost (red) bar, is filtered out.

---

* *From a system-level perspective, the RF frequency is typically the frequency at the input to the downconverter and the IF frequency is at the output of the downconverter. This matches the definitions of the RF Reference Frequency and IF Reference Frequency for the digital receiver. However, when the DDC's Reference Point is located at the output of the receiver, the DDC's RF Reference Frequency is at the system's IF frequency, and the DDC's IF Reference Frequency is at baseband.*

As stated in Section 7.1.5.6, the sum of the RF Reference Frequency Offset and the RF Reference Frequency gets translated to the IF Reference Frequency. This rule can be applied to both the Receiver and DDC blocks in this example. The following equation shows the translated frequency relationship of the Receiver. Note that because the RF Reference Frequency Offset of the Receiver is unspecified in the Context Packet Stream, it is assumed to be zero. The left-hand side of this equation gives the frequency at the Receiver's Reference Point (the Receiver's input) that is translated to the Receiver's IF Reference Frequency. In Figure B.3-2 this is the leftmost (red) bar in the top axis. The right-hand side of this equation is shown by the leftmost (red) bar in the middle axis.

$$\underbrace{\begin{array}{c}\text{Receiver}\\ \text{RF Reference}\\ \text{Frequency}\end{array} + \begin{array}{c}\text{Receiver}\\ \text{RF Reference}\\ \text{Frequency Offset}\end{array}}_{\begin{array}{c}\text{At the Receiver's Reference Point}\\ \text{(Receiver's input)}\end{array}} \overset{\text{Frequency Translation}}{\longleftrightarrow} \begin{array}{c}\text{Receiver}\\ \text{IF Reference}\\ \text{Frequency}\end{array}$$

The same relationship holds for DDC2. However, in this case the RF Reference Frequency Offset is non-zero. The left-hand side of this equation shows the frequency that is translated to the DDC's Reference Point (the Receiver's output) that is translated to the DDC's IF Reference Frequency. In Figure B.3-2 this is the rightmost (green) bar in the middle axis. The right-hand side of this equation is shown by the (green) bar in the bottom axis.

$$\underbrace{\begin{array}{c}\text{DDC 2}\\ \text{RF Reference}\\ \text{Frequency}\end{array} + \begin{array}{c}\text{DDC 2}\\ \text{RF Reference}\\ \text{Frequency Offset}\end{array}}_{\begin{array}{c}\text{At DDC2's Reference Point}\\ \text{(Receiver's output)}\end{array}} \overset{\text{Frequency Translation}}{\longleftrightarrow} \begin{array}{c}\text{DDC 2}\\ \text{IF Reference}\\ \text{Frequency}\end{array}$$

The two relationships above can be combined to find the frequency at the Receiver's input that gets translated to the DDC's IF Reference Frequency. Knowing that the Receiver's IF Reference Frequency is equal to the DDC's RF Reference Frequency, the upper equation can be substituted into the lower equation to yield the relationship below. Here the left-hand side of the relationship is shown by the rightmost (green) bar in the top axis and the right-hand side of this equation is again the (green) bar in the bottom axis.

$$\underbrace{\begin{array}{c}\text{Receiver}\\ \text{RF Reference}\\ \text{Frequency}\end{array} + \begin{array}{c}\text{Receiver}\\ \text{RF Reference}\\ \text{Frequency Offset}\end{array} + \begin{array}{c}\text{DDC 2}\\ \text{RF Reference}\\ \text{Frequency Offset}\end{array}}_{\begin{array}{c}\text{At the Receiver's Reference Point}\\ \text{(Receiver's input)}\end{array}} \overset{\text{Frequency Translation}}{\longleftrightarrow} \begin{array}{c}\text{DDC 2}\\ \text{IF Reference}\\ \text{Frequency}\end{array}$$

Consider this system when the receiver translates a 32 MHz band centered at 1 GHz down to an IF frequency of 70 MHz and the bank of 64 DDCs breaks this band into 500 kHz-wide channels with a channel spacing that is also 500 kHz. If DDC1 converted the IF band centered at 70 MHz down to baseband and DDC2 converted the IF band centered at 70.5 MHz down to baseband then the following parameters are sent in their respective Context Packet Streams.

Digital Receiver Context Packet Stream	
Stream ID	200
Reference Point ID	100
Bandwidth	32 MHz
IF Reference Frequency	70 MHz
RF Reference Frequency	1 GHz

DDC 1 Context Packet Stream	
Stream ID	301
Reference Point ID	200
Bandwidth	500 kHz
IF Reference Frequency	0 Hz
RF Reference Frequency	70 MHz
RF Reference Frequency Offset	0 MHz
Source Context Assoc. List	200

DDC 2 Context Packet Stream	
Stream ID	302
Reference Point ID	200
Bandwidth	500 kHz
IF Reference Frequency	0 Hz
RF Reference Frequency	70 MHz
RF Reference Frequency Offset	500 kHz
Source Context Assoc. List	200

Figure B.3-3: Context Packet Stream Contents for the RF Reference Frequency Offset Example

Given these Context Packet Streams a VRT processor could calculate that, of the spectrum at the receiver output, the band centered at 70.5 MHz was translated to 0 Hz at DDC2's output and that, of the spectrum at the receiver input, the band centered at 1000.5 MHz was translated to 0 Hz at DDC2's output.

To see why the additional computation required by the RF Reference Frequency Offset is useful, note that if the receiver sweeps its tuned frequency to 1032 MHz, then 1064 MHz, and so on, only a single digital receiver Context packet needs to be sent per frequency step. The 64 Context packets for the 64 DDCs do not need to be sent at each frequency step. This feature can significantly reduce the chance of link congestion in some systems.

B.4 *IF Band Offset Example*

The Bandwidth field is used to describe the amount of usable signal spectrum at the output of a process. For many systems the center of this signal band is at the IF Reference Frequency. For those systems where the IF Reference Frequency is not at the center of the signal band, the IF Band Offset field is used to specify the difference between the true center of the band and the IF Reference Frequency. Assume that it is necessary to have the IF Reference Frequency correspond to the lower band edge.for the digital receiver shown in Figure B.4-1.

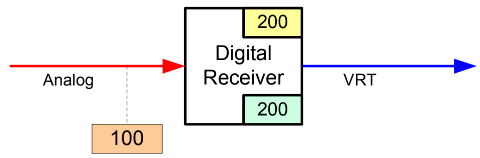


Figure B.4-1: Block Diagram for IF Band Offset Example

Figure B.4-2 shows the signal spectrum at the input and output of the receiver. The IF Reference Frequency is at the lower band edge and the IF Band Offset specifies the amount to add to the IF Reference Frequency value to get the true center of the band described by the Bandwidth field.

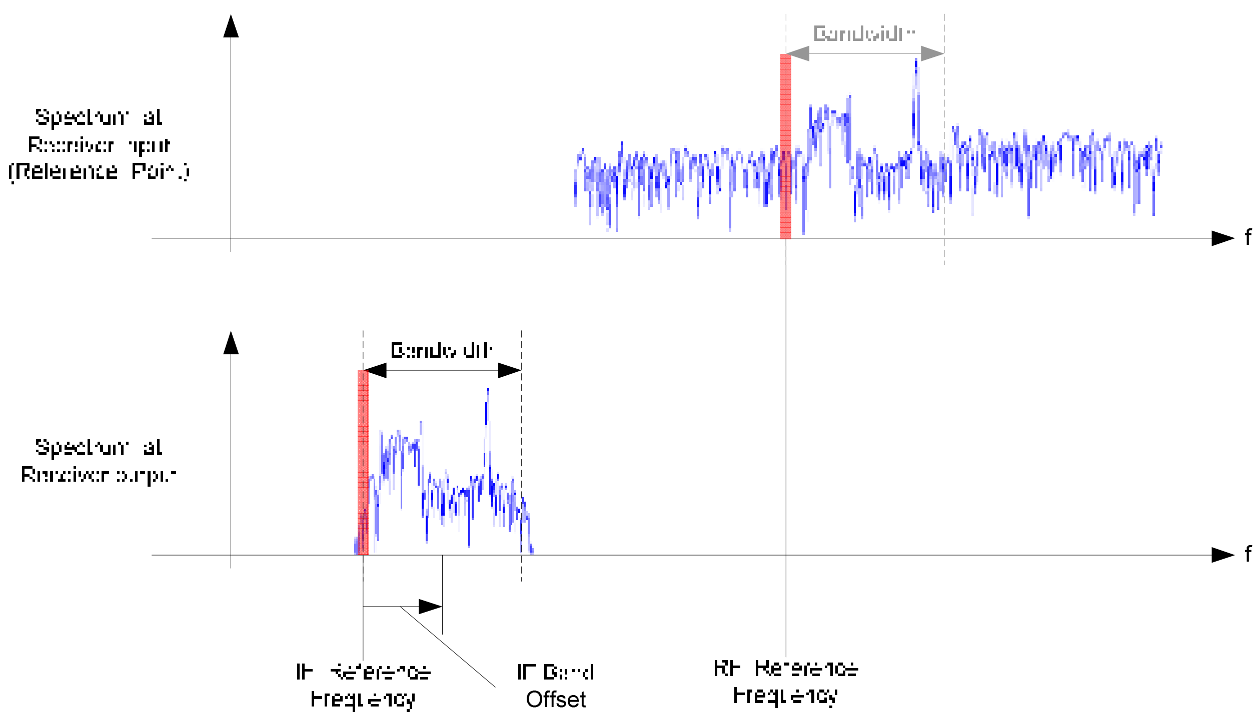


Figure B.4-2: Original and translated spectra for IF Band Offset example

The IF Reference Frequency marks the lower band edge. The sum of the IF Reference Frequency and IF Band Offset gives the band center frequency.

The following numerical example is the same as that given in Appendix B.2, but with the IF Reference Frequency at the lower band edge instead of the band center.

If the tuner were tuned to 2 GHz and downconverted a 30 MHz band to a 70 MHz IF without inverting its spectrum then the Context packet contains the values shown in Figure B.4-3

Digital Receiver Context Packet Stream	
Stream ID	200
Reference Point ID	100
Bandwidth	30 MHz
IF Reference Frequency	55 MHz
RF Reference Frequency	1.985 GHz
IF Band Offset	15 MHz
Spectral Inversion Indicator	0

Figure B.4-3: Context Packet Stream Contents for IF Band Offset Example

B.5 *Frequency Translation Example*

In many applications, equipment from different vendors is connected together to perform a particular task. For some systems the Context from each individual component may not be useful, but the combined settings of the entire system are of interest to the end user. Therefore the system designer may choose to aggregate the Context from each component and generate a summary Context Packet Stream for the entire system to be sent along with the system's final Data Packet Stream.

Aggregating Context from a cascade of components is generally a straight-forward procedure. For the system delay parameter, summing the individual Timestamp Adjustment fields usually suffices (or equivalently adjusting the Timestamp by the same amount). For the system level parameter, the gain of each component can be added to the Reference Level of the digitizer. The system bandwidth is typically the bandwidth of the final band-limiting process. The determination of the original RF frequency, however, is complicated when the input and output center frequencies of neighboring process are mismatched or when spectral inversion is encountered.

Consider the VRT-enabled components shown in Figure B.5-1 which were combined to form a VRT system. The SHF tuner downconverts a 500 MHz band from a center frequency of 10 GHz to an IF center frequency of 1200 MHz. The UHF tuner converts a 30 MHz portion of this band centered at 1400 MHz down to an IF center frequency of 20 MHz. There the signal is digitized and a DDC is used to downconvert a 1 MHz band from a center frequency of 30 MHz to baseband.

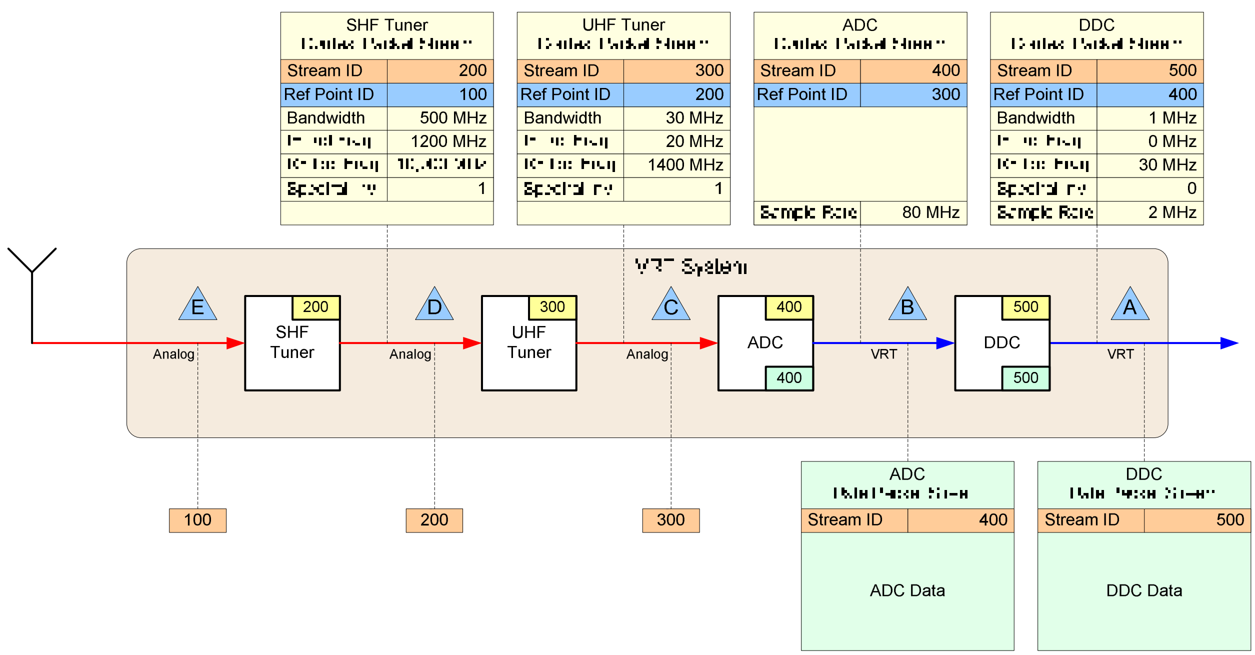


Figure B.5-1: VRT System Components for Frequency Translation Example

VRT Context Packet Streams are generated by each of these components (or their controllers) and VRT Data Packet Streams are generated by the ADC and DDC. The end user of the system, however, would like to be presented with only the DDC Data Packet Stream and a Context Packet Stream that describes the settings of the combined system as shown in Figure B.5-2. In other words, the end user would like the Data and Context for only the DDC process, but with the Reference Point for the DDC Context moved up to the antenna.

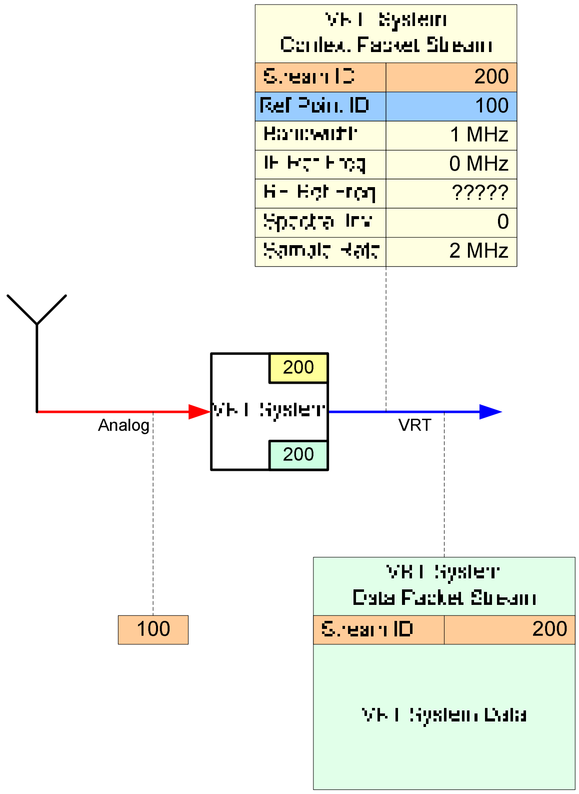


Figure B.5-2: Composite VRT System for Frequency Translation Example

As shown in Figure B.5-2, several Context parameters for the aggregate VRT system are easy to determine. The Bandwidth, IF Reference Frequency, and Sample Rate fields for the system are the same as those for the final DDC process. There are two spectral inversions (in the SHF and UHF tuners) between the original DDC input Reference Point and the desired antenna Reference Point. These cancel each other out so that the spectral inversion indicator for the system is the same as that for the DDC. The final parameter to determine is the RF Reference Frequency, which requires calculation.

This calculation is made simpler by first determining the *translation frequency* for each process in the signal path. The translation frequency is the difference between the frequency of the original RF signal and the frequency of the *upright* spectral image after frequency conversion. In terms of VRT parameters shown in Figure B.5-3, the translation frequency is the difference between the RF Reference Frequency and the IF Reference Frequency if no spectral inversion occurs, and the difference between the RF Reference Frequency and the negative IF Reference Frequency if spectral inversion does occur during the frequency conversion.

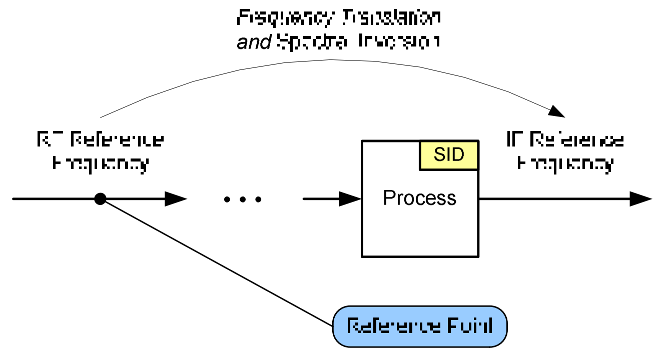


Figure B.5-3: VRT Parameters and Frequency Conversion

The translation frequency is shown graphically for the spectrally inverting and non-inverting cases in Figure B.5-4.

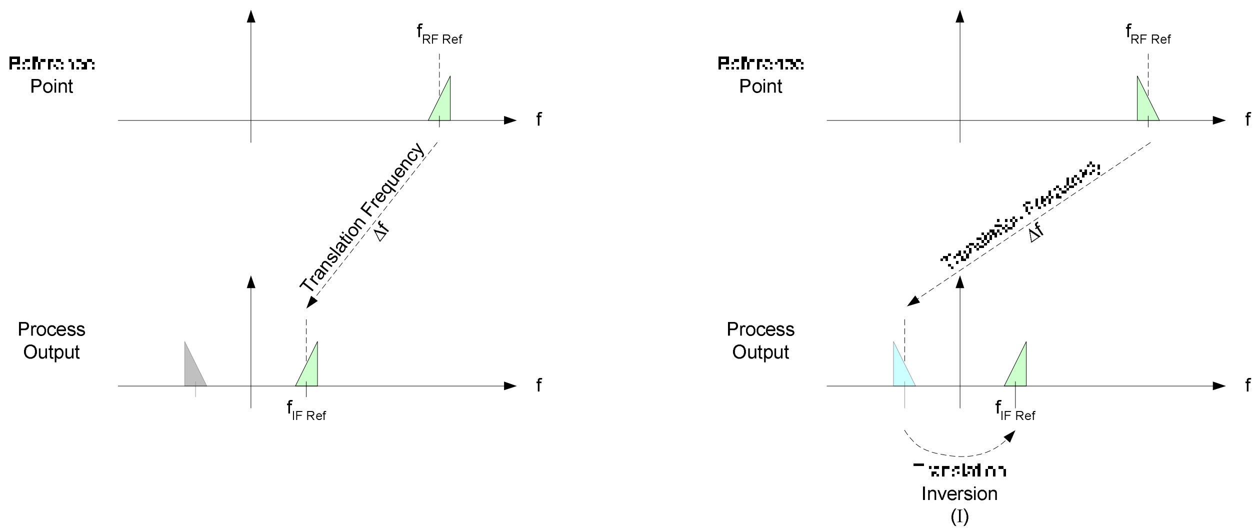


Figure B.5-4: The Translation Frequency for Spectral Inversion and Non-inversion

The translation frequency is the difference between the original spectra at the Reference Point and the upright spectra after frequency conversion. For the non-inverting downconversion shown on the left, the translation frequency is simply the difference between the RF and IF Reference Frequencies. For the inverting downconversion, it is the difference between the RF Reference Frequency and the negative IF Reference Frequency.

The equation relating these VRT fields is

$$f_{\text{RF Ref}} + \Delta f = I \cdot f_{\text{IF Ref}} \tag{1}$$

where Δf is the translation frequency and I is related to the Spectral Inversion Indicator. I is +1 for non-inverting translations and -1 for inverting translations. Note that if the RF Frequency Offset field were also utilized in this system then $f_{\text{RF Ref}}$ is replaced with $f_{\text{RF Ref}} + f_{\text{RF Ref Offset}}$ in the above equation.

Once the translation frequencies for each process is known, it is straight-forward to determine the relationship between the IF Reference Frequency and the RF Reference Frequency for any Reference Point in the signal path. For the example shown in Figure B.5-1 the translation frequencies are:

$$\Delta f_{\text{SHF}} = I_{\text{SHF}} \cdot f_{\text{SHF IF Ref}} - f_{\text{SHF RF Ref}} = (-1) \cdot 1200MHz - 10GHz = -11{,}200MHz$$
$$\Delta f_{\text{UHF}} = I_{\text{UHF}} \cdot f_{\text{UHF IF Ref}} - f_{\text{UHF RF Ref}} = (-1) \cdot 20MHz - 1400MHz = -1{,}420MHz$$
$$\Delta f_{\text{ADC}} = 0Hz$$
$$\Delta f_{\text{DDC}} = I_{\text{DDC}} \cdot f_{\text{DDC IF Ref}} - f_{\text{DDC RF Ref}} = (+1) \cdot 0MHz - 30MHz = -30MHz$$

Figure B.5-1 has five triangle symbols at the inputs and outputs of the system components. The relationships between these five points can be expressed using equation (1) as:

$$f_B = I_{DDC} f_A - \Delta f_{DDC}$$
$$f_C = I_{ADC} f_B - \Delta f_{ADC}$$
$$f_D = I_{UHF} f_C - \Delta f_{UHF}$$
$$f_E = I_{SHF} f_D - \Delta f_{SHF}$$

Using substitution, the frequency relationship between points A and E is:

$$f_E = I_{SHF}\left(I_{UHF}\left(I_{ADC}\left(I_{DDC} f_A - \Delta f_{DDC}\right) - \Delta f_{ADC}\right) - \Delta f_{UHF}\right) - \Delta f_{SHF}$$
$$f_E = I_{SHF} I_{UHF} I_{ADC} I_{DDC} f_A - I_{SHF} I_{UHF} I_{ADC} \Delta f_{DDC} - I_{SHF} I_{UHF} \Delta f_{ADC} - I_{SHF} \Delta f_{UHF} - \Delta f_{SHF}$$

Points A and E are the IF and RF Reference Frequency points for the aggregate system. Therefore the RF Reference Frequency for the system is calculated as:

$$f_{\text{System RF Ref}} = I_{SHF} I_{UHF} I_{DDC} f_{\text{System IF Ref}} - I_{SHF} I_{UHF} \Delta f_{DDC} - I_{SHF} \Delta f_{UHF} - \Delta f_{SHF}$$
$$f_{\text{System RF Ref}} = 0Hz + 30MHz - 1{,}420MHz + 11{,}200MHz$$
$$f_{\text{System RF Ref}} = 9{,}810MHz$$

The translation frequency for the entire system can be found as:

$$f_{\text{System RF Ref}} = I_{\text{System}} f_{\text{System IF Ref}} - \Delta f_{\text{System}} = I_{SHF} I_{UHF} I_{DDC} f_{\text{System IF Ref}} - I_{SHF} I_{UHF} \Delta f_{DDC} - I_{SHF} \Delta f_{UHF} - \Delta f_{SHF}$$
$$I_{\text{System}} = I_{SHF} I_{UHF} I_{DDC} = +1$$
$$\Delta f_{\text{System}} = I_{SHF} I_{UHF} \Delta f_{DDC} + I_{SHF} \Delta f_{UHF} + \Delta f_{SHF} = -9810MHz$$

The frequency translations and spectral inversions are also shown graphically in Figure B.5-5.

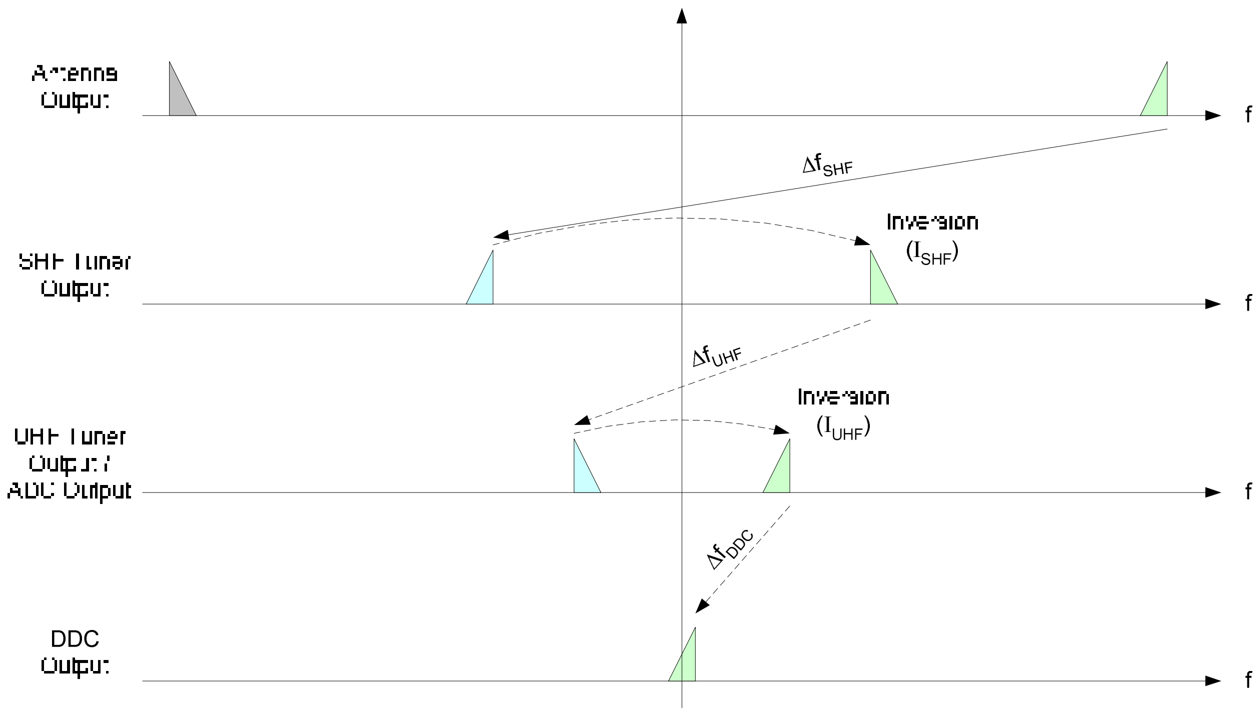


Figure B.5-5: Frequency Translations within the System

The frequency translation for the entire system simplifies to that shown in Figure B.5-6.

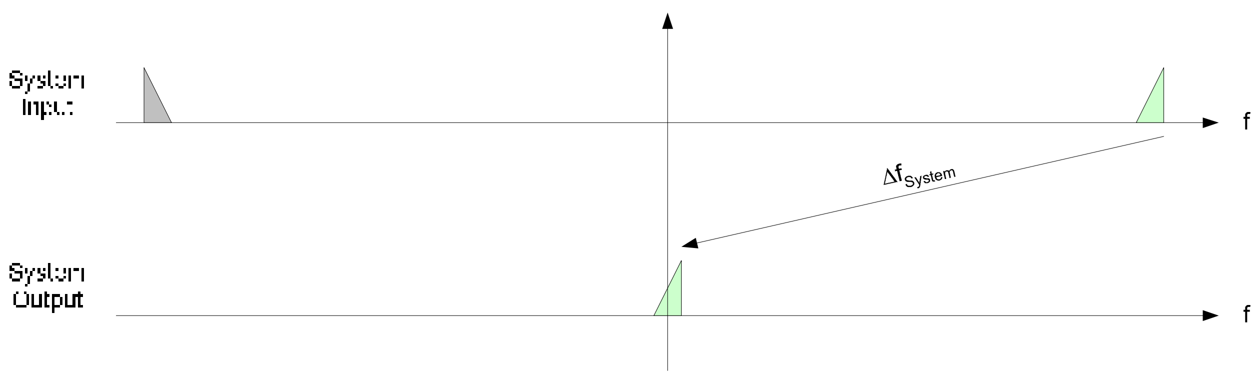


Figure B.5-6: Frequency Translation for Cascaded System

The RF and IF Reference Frequencies within the system are shown in Figure B.5-7

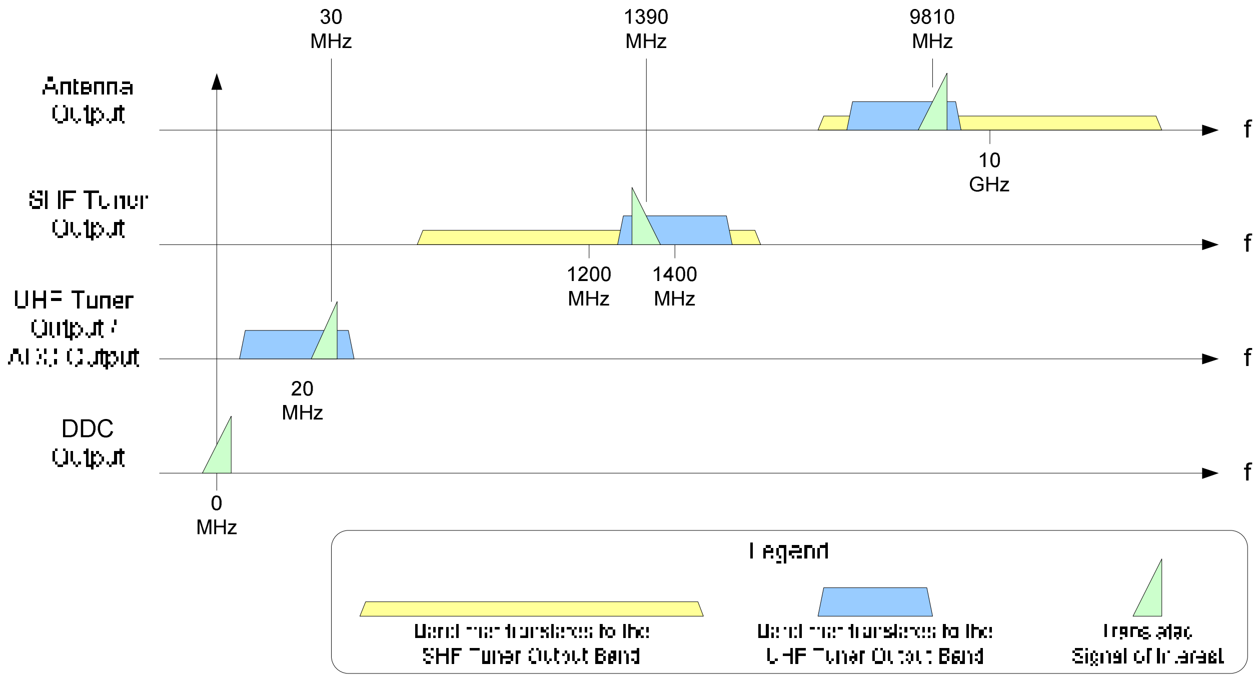


Figure B.5-7: RF and IF Reference Frequencies within the System

B.6 *Reference Level Example*

This section contains an example illustrating the use of the Reference Level field. Figure B.6-1 shows a trivial example containing only a single ADC. In this example it is assumed that a 2 Vp-p tone applied to Reference Point, which is the ADC input, generates a Unit-Scale Sinusoid* at its digital output. A 2 Vp-p tone into a 50Ω load corresponds to an input power of +10 dBm. Therefore the Reference Level of the ADC Data Packet Stream is +10 dBm.

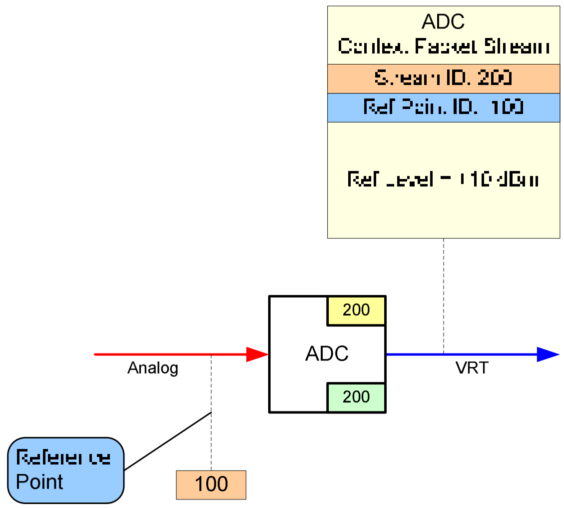


Figure B.6-1: Reference Level of ADC

Figure B.6-2 shows a Unit-Scale Sinusoid generated when a +10 dBm signal is applied to the ADC input. Assuming the ADC has an 8-bit, signed, two's-complement output the integer representation of the Unit-Scale Sinusoid spans the range of -2^7 to $+2^7$ * (granted that the actual ADC saturates at $+2^7-1$), which is the same as saying the Normalized Interpretation of the Unit-Scale Sinusoid spans the range from -1 to +1.

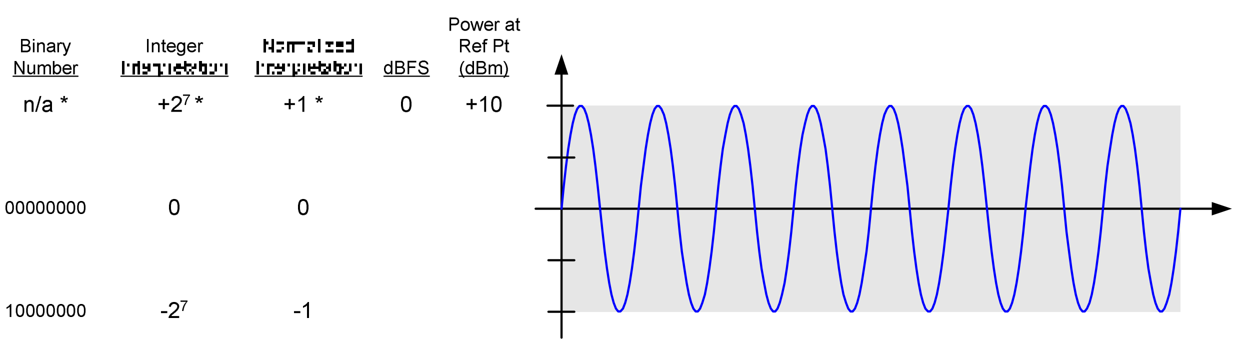


Figure B.6-2: Unit-Scale Sinusoid at the ADC Output

Now consider a digital receiver shown in Figure B.6-3 consisting of a downconverter, which has a gain of 30 dB, and the same ADC used in the example above. If the Reference Point of the receiver were at its input then a Unit-

---

* *By definition, the integer representation of a Unit-Scale Sinusoid spans the range from -2^{N-1} to $+2^{N-1}$ for N-bit signed fixed-point data. Although the most positive value of the Unit-Scale Sinusoid, $+2^{N-1}$, cannot be represented by an N-bit signed fixed-point number, Unit-Scale Sinusoids are still useful in describing the levels of signals represented by fixed-point numbers.*

Scale Sinusoid is generated when a -20 dBm tone is applied to its input. Therefore the Reference Level of the receiver is -20 dBm. Note that the Reference Level is defined for a single input tone at a frequency assumed to be at the center of the passband as defined by the system designer.

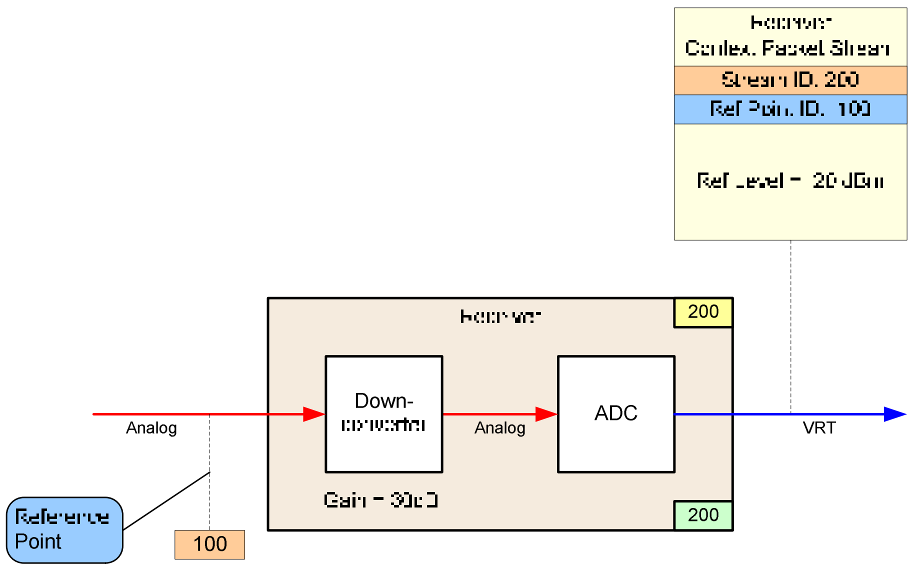


Figure B.6-3: Reference Level of Receiver

Finally, consider the example shown in Figure B.6-4 where the ADC is followed by a hypothetical process which performs an arithmetic right shift (preserving the sign bit) on the ADC data. Both the ADC data and the right-shifted data are 8-bit 2's complement numbers, though the integer and Normalized Interpretation of the right-shifted data will be half the value of the ADC data (or nearly half the value).

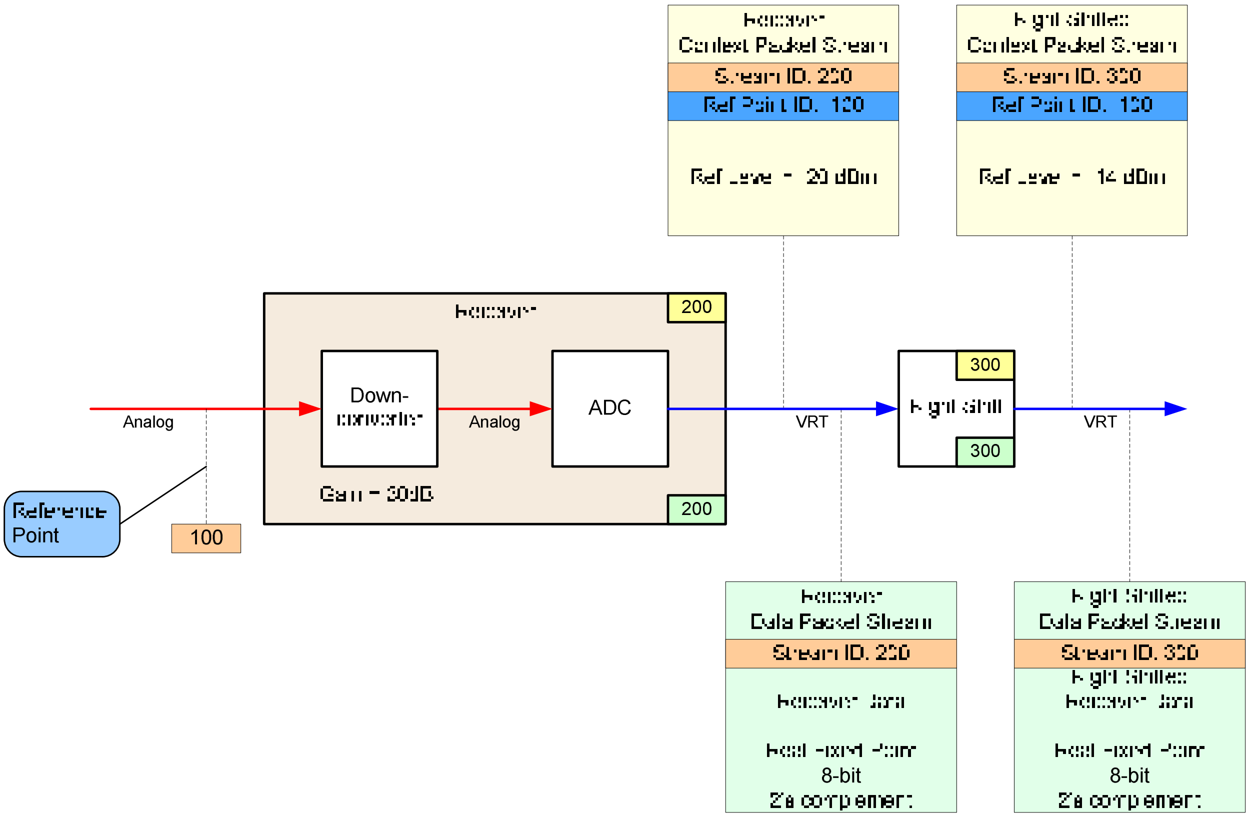


Figure B.6-4: Reference Level Changes with Gain and Attenuation

The Reference Level for the right-shifted Data Packet Stream is 6 dB higher than that of the receiver Data Packet Stream because it takes a signal 6 dB higher to generate a Unit-Scale Sinusoid at the output of the right-shift process. Therefore the Reference Level is -14 dB.

Note that signal distortion due to compression or saturation is not considered when specifying the Reference Level. The system is assumed to be completely linear when calculating the Reference Level for any Data Packet Stream and any Reference Point.

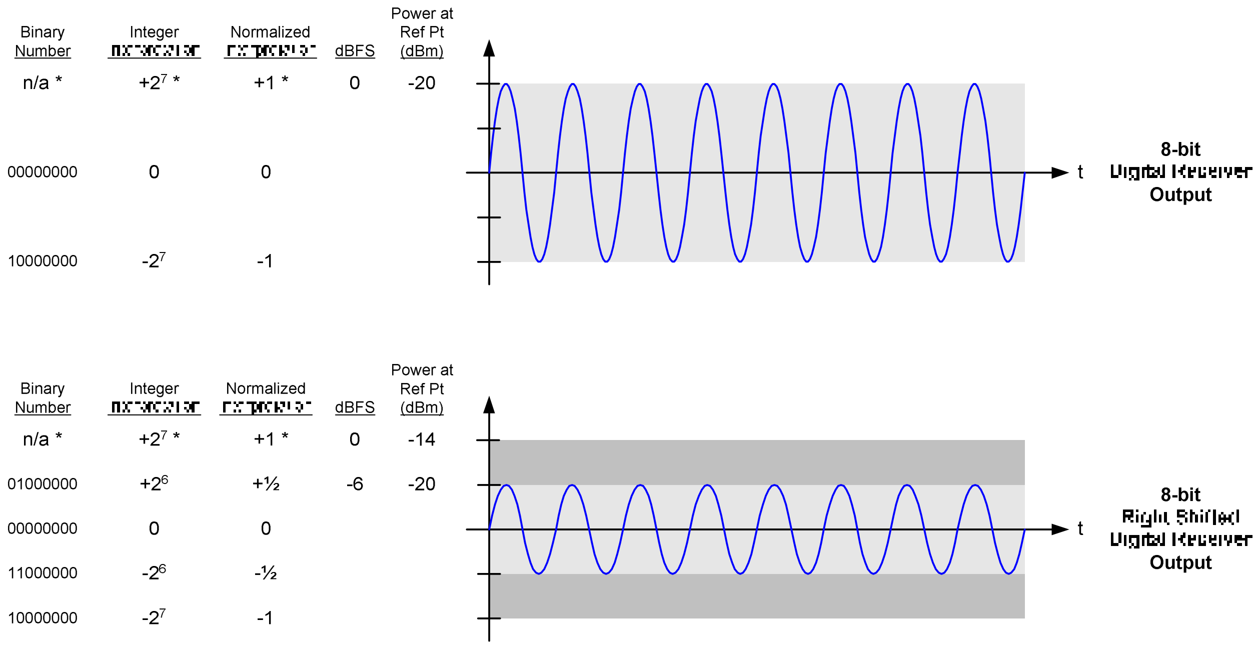


Figure B.6-5: Gain and Attenuation Change the Reference Level

A -20 dBm signal at the Reference Point (the Receiver input) produces a Unit-Scale Sinusoid at the output of the Digital Receiver. However, this -20 dBm signal only produces a half-scale signal at the output of the right shift process, as shown in the lower portion of this figure. It takes a -14 dBm signal (not shown) at the Reference Point to generate a Unit-Scale Sinusoid at the output of the right-shift process. Therefore the Reference Level of the right-shifted data is -14 dBm.

* *By definition, the integer representation of a Unit-Scale Sinusoid spans the range from* -2^{N-1} *to* $+2^{N-1}$ *for N-bit signed fixed-point data. Although the most positive value of the Unit-Scale Sinusoid,* $+2^{N-1}$*, cannot be represented by an N-bit signed fixed-point number, Unit-Scale Sinusoids are still useful in describing the levels of signals represented by fixed-point numbers.*

B.7 *Gain Example*

The Gain and Reference Level fields can both be used to send metadata regarding the level of signals in VRT systems. However there are instances where the use of the Gain field may be more appropriate that the Reference Level field, and vice versa. For example, a device that is able to generate VRT Context Packets (or have Context Packets made for it) with an analog input and analog output is only able to use the Gain field because Reference Level has no meaning for analog signals. Devices with digital input and digital output may describe the Reference Level field of the output only if the Reference Level of the input is known. Otherwise the Gain field must be used to convey the relative signal levels from input to output.

This section contains examples illustrating the use of the Gain field. It is based on the Reference Level example described in B.6. Figure B.7-1 shows the example system, which matches that in Figure B.6-4 except that the receiver block is now described as separate downconverter and digitizer blocks and that each block specifies its own input at its Reference Point instead of having a single Reference Point at the input to the system.

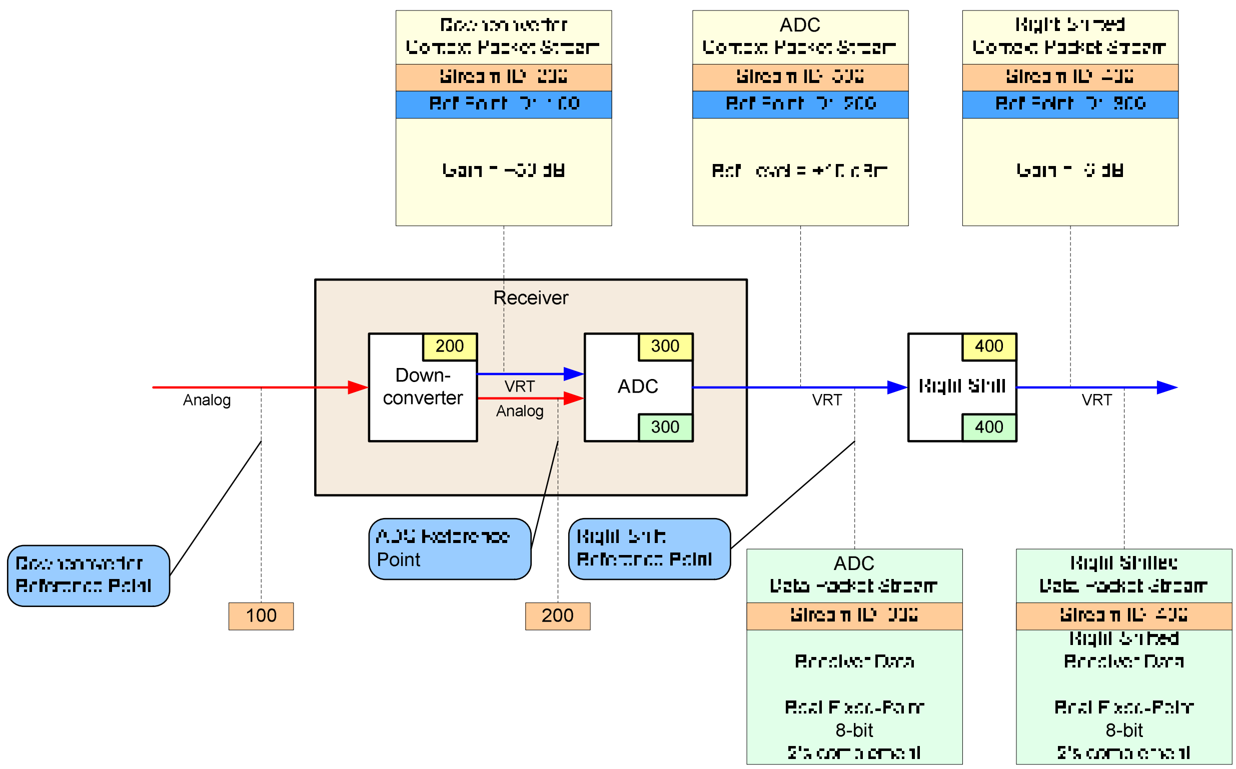


Figure B.7-1: Block Diagram for Gain Example

In this example the downconverter's Context Packet Stream specifies its gain as being +30 dB, which simply means that the power of a signal at the output is 30 dB higher than at the input of the downconverter, ignoring any non-linear effects such as saturation. The Reference Level field is only used in conjunction with processes that generate a Data Packet Stream, so the Reference Level field cannot be used with the downconverter whose signal output is analog.

The Gain field is defined for processes with an analog Reference Point and analog output, and for processes with a digital Reference Point and a digital output, but not for processes with a mixture of analog Reference Points and digital outputs. Therefore the Gain field has no meaning for the ADC block. In this example the ADC instead specifies a Reference Level that relates the signal amplitude at the ADC input to the level of the output samples. As in the previous section, a 10 dBm tone generates a Unit-Scale Sinusoid at the output of the ADC block.

With the Gain field from the downconverter and the Reference Level fields from the ADC it can be calculated that if the Reference Point for the ADC were moved to the input of the receiver then the Reference Level of the new ADC Data Packet Stream is -20 dBm just as in the previous section.

The Context Packet Stream for the right-shifted data contains the Gain field that gives the gain from the receiver output (the Reference Point for the right shift process) to the right shift output (the Described Signal). The Reference Point and Described Signal are both digital signals, so the Gain field will contain the digital gain between these points. Digital gain differs from analog gain in that for the bit widths at the Reference Point and Described Signal must be considered for fixed point signals.

The example shown in B.7-1 demonstrates the simple case where the bit width of the Reference Point and Described Signal are both 8 bits wide. If a Unit-Scale Sinusoid were generated at the Reference Point, the Described Signal would be a sinusoid that was half the amplitude of a Unit-Scale Sinusoid at the right shift output, as shown in Figure B.6-5. Therefore the digital gain between these points is -6 dB, and the Context Packet Stream would convey a Gain field of -6 dB.

The Impact of Changes in Bit Widths on Digital Gain

Bit width changes in the signal path of a fixed point signal may or may not impact the Gain of a digital process. A designer may change bit widths in any manner, be it zero padding, sign extension, truncation, rounding,etc. but should provide Packet Class Documention or Context Packets that accurately indicate the affect of the bit width change on signal levels with the Reference Level or Gain fields.

For example, consider the two examples shown in Figure B.7-2 where the bit width changes from 8 bits at the Reference Point to 16 bits at the Described Signal.

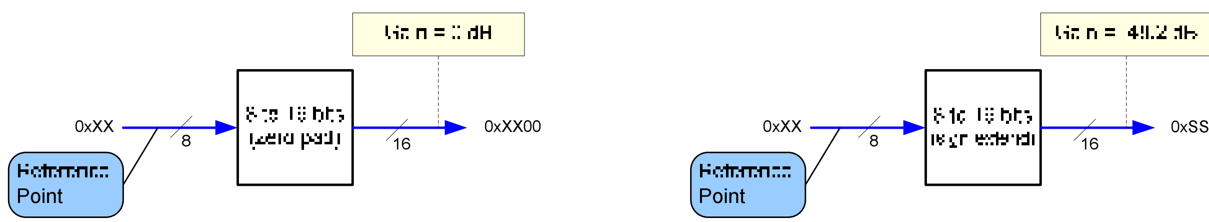


Figure B.7-2: Changes in Bit Width affect Digital Gain

In the first case, shown at the left of Figure B.7-2, the bits of the input sample are placed in the msbs of the output sample and the 8 lsbs of the output sample are zero padded. In this case, an 8-bit Unit-Scale Sinusoid at the input would generate a signal at the output with the same amplitude as a 16-bit Unit-Scale Sinusoid. When a Unit-Scale Sinusoid at the Reference Point (using the Reference Point bit width) generates a Unit-Scale Sinusoid Described Signal (using the Described Signal bit width), the digital gain is 0 dB by Rule 7.1.5.10-1.

In the second case, shown at the right of Figure B.7-2, the bits of each input sample are placed in the lsbs of an output sample, and the msbs of the output sample are set equal to the sign of the input sample, effectively sign extending the input sample to 16 bits. In this case, an 8-bit Unit-Scale Sinusoid at the input would generate a signal at the output that was 8-bits below a 16-bit Unit-Scale Sinusoid. Therefore, by Rule 7.1.5.10-1 the digital gain would be -48.2 dB.

In all cases, the digital gain for fixed point signals can be calculated as:

$$Gain = \frac{\text{Power of output signal relative to a Unit-Scale Sinusoid at the Described Signal (using the bit width of the Described Signal)}}{\text{Power of input signal relative to a Unit-Scale Sinusoid at the Reference Point (using the bit width of the Reference Point)}}$$

B.8 *Timestamp Adjustment Example*

Section 4.2 introduced the concept of Reference-Point Time as the time of an event or signal as seen at a specified Reference Point. Sometimes Reference-Point Time can be conveyed in the Timestamp directly, but it is often better to convey it as the sum of the Timestamp and the Timestamp Adjustment. The example in this section illustrates the use of the Timestamp Adjustment field to make corrections to the value contained in the Timestamp field in order to convey the proper Reference-Point Time.

Before presenting a specific example demonstrating the use of the Timestamp Adjustment field it is necessary to distinguish between the timing of a signal as seen at one or more Reference Points, and the timing of signal processing. The first of these we refer to as *signal time*, and the second we refer to as *actual time.* This distinction arises due to two things. First, because it is possible to digitize signals, store them, and process them later, the actual time that a signal is processed may be quite different than the time it passed some Reference Point, such as the antenna. The second reason the distinction arrises is that signal processing may procede at a rate that is either faster or slower than the sampling rate of the signal.

Actual time, or processing time, is time in the real world measured with respect to an unstoppable timing reference. This is the understanding of time that most people are familiar with. However, when dealing with sampled data where an event that occurred at time T_0 is sampled and processed at a later time $T_0+\tau$, it is useful to have a separate time base that relates to the timing of the original event but does not consider the delay in processing, τ. Therefore signal time relates the instant of an event with an external time reference at the time of sampling. Signal time is stoppable in the sense that once a signal has been sampled it can be stored indefinitely and processed at a later time. Time with respect to the processing time base continues all the while but time with respect to the signal time base is halted during the storage. However, note that the sampled signal must be timestamped and the timestamp must also be stored in order to extract the original timing of the sampled event, i.e the signal time.

An important point about signal time and actual time is that they can differ *only* for digital processes. They are always the same for processes with analog inputs and analog outputs because in the continuous-time analog domain signals always propagate according to the laws of physics in actual time. By contrast, a signal can be digitized, timestamped, and sent over a coaxial cable to another device for processing and the amount of propagation delay from the digitizer output to the next device's input does not affect its signal time because the original timing can be extracted from the timestamp of the sampled data. However, propagation from an analog device to another analog device does affect its signal time because it has not yet been digitized and a timestamp has not been attached to the signal.

Another important point is that even though signal time of a timestamped, digitized signal is frozen during storage and propagation from one process to another, signal time for this signal does not generally match the original time at the sampling device. One must still consider algorithmic delays after digitization such as group delay[*] through a digital filter. Figure B.8-1 shows an example system with several analog and digital components. Consider an event detected by a processor attached to the end of this signal path that needs to determine the timing of an event at the antenna input that caused a related event seen in the DDC's digital output signal. To do this the processor could take the timestamp of the signal and adjust it for the group delay of the DDC, the sampling delay of the ADC, the group delay of the cabling between the antenna and ADC, and any group delay in the antenna module itself. It does not need to adjust for the buffer or propagation delays after digitization because timestamping occurred at the point of digitization.

[*] *Note that though group delay varies with frequency, the Timestamp Adjustment field can only convey a single delay value. It is therefore most useful in those applications where the group delay can be considered constant over a particular band of interest.*

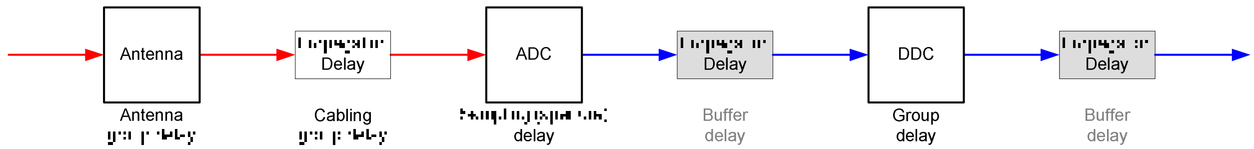


Figure B.8-1: Actual Time and Signal Time
Group delay affects signal time for both analog and digital processes.
Propagation delay affects signal time only for analog processes.

Next consider the example shown in Figure B.8-1. In this example the Timestamps need to express the signal time of the information in the Data Packet Stream as seen at the Reference Point, which is the antenna. So the Reference-Point Time is signal time, which is the typical case. We specify the group delay through the antenna and cabling as 10 ns, the sampling delay of the ADC module is 5 ns, and the group delay through the DDC filter is 1 µs. At the output of the ADC and DDC modules the sampled data is buffered into VRT Data packets, but the buffering delay and propagation delay from one module to another does not affect signal time so values for these delays are not necessary. One VRT implementation of this system may look like that shown in Figure B.8-2 where a single system Reference Point is defined at the antenna input (with Stream ID 100).

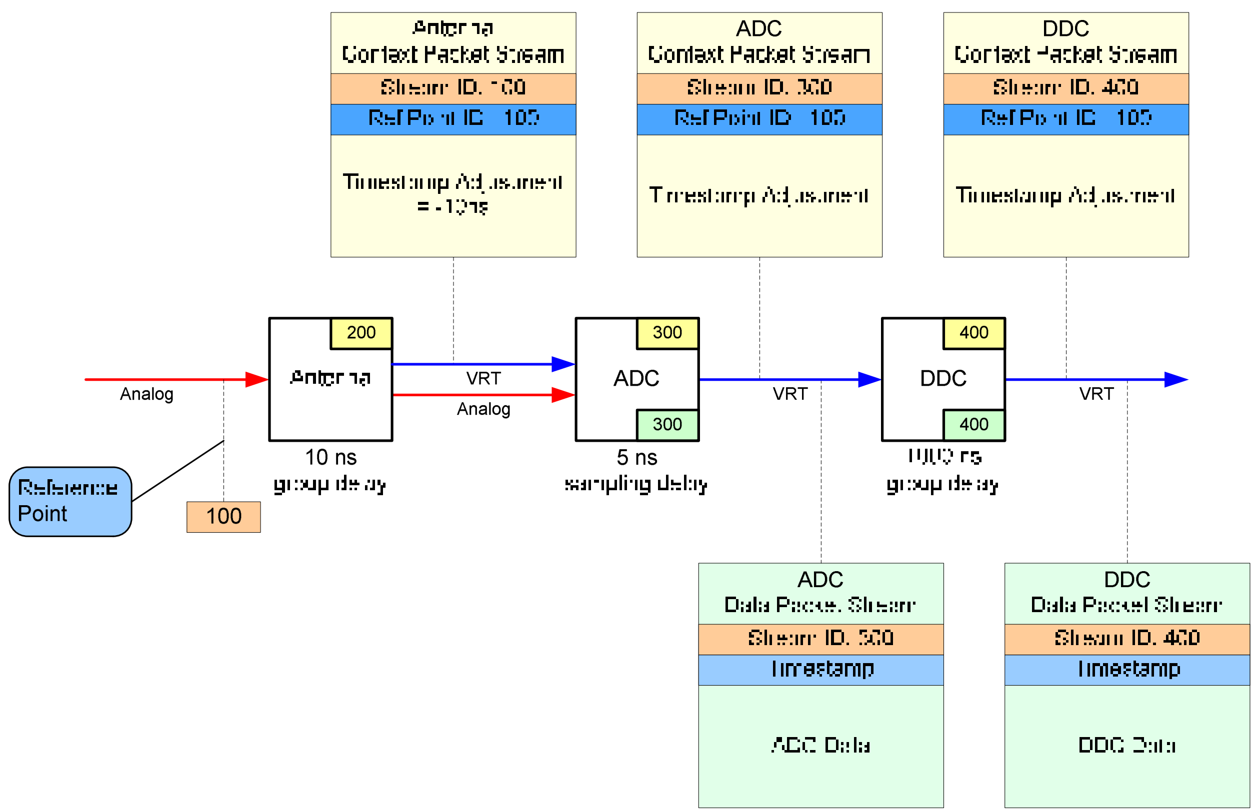


Figure B.8-2: Timestamp Adjustment Example with Single Reference Point at the Antenna Input

The value placed in the Timestamp Adjustment field for the antenna's Context Packet Stream is set to -10ns to convey the group delay of the antenna module. Negative Timestamp Adjustments indicate that events at the antenna's Reference Point, which is the antenna input, occurred *earlier* than the corresponding events reflected in the output signal of the antenna.

The values placed in the Timestamp Adjustment field of the ADC's and DDC's Context Packet Streams depend on the value of the Timestamps of the paired Data Packet Streams, which are based on the timing of the first samples in their respective packets. This example illustrates a few of the possibilities for both Timestamp and Timestamp Adjustment fields.

Figure B.8-3 shows an example of the analog signal at the antenna input and the digitized signals at the output of the ADC and the DDC. All three signals are shown in signal time. The top trace shows a hypothetical impulse-like spike occurring at the antenna input. The middle trace shows that it takes this spike 15ns to propagate from the antenna input and appear at the ADC output. The bottom trace shows the response of the DDC to the spike. The output of the DDC peaks 1µs after the spike appears (in signal time) at the DDC input.

In this figure the instant of digitization of the impulse is labeled as T_0. It can be seen that the impulse occurred at the antenna input a time δ_1 before T_0. δ_1 is the sum of the group delay through the antenna, the group delay through the antenna cabling and the sampling delay of the ADC, which is 15ns. The group delay of the DDC filter is shown as δ_2 = 1µs.

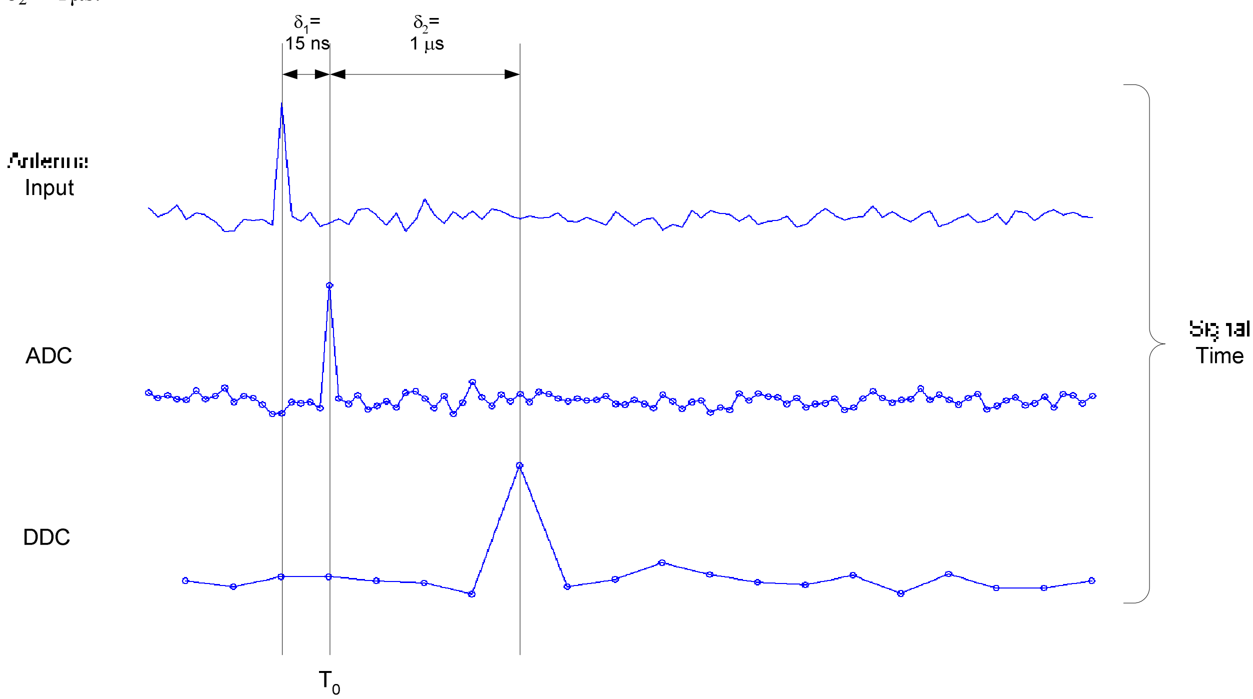


Figure B.8-3 : Signals with respect to Signal Time

First we consider the case where the first samples of the ADC and DDC data packets are timed as shown in Figure B.8-4. The top three traces show the three signals in signal time identical to Figure B.8-3. The shaded boxes indicate which samples are included in a single IF Data packet for the ADC and DDC. Here the first sample of the DDC data packet occurs a time δ_2 after the first sample in the ADC packet occurs, effectively hiding the impact of the DDC group delay in the samples themselves.

For this example the sum of the Timestamp and Timestamp Adjustment is equal to T_0-δ_1 for both the ADC and DDC data packets. This is because the Reference Point for the ADC and DDC is the antenna input, and the information in the signal at the antenna input that occurred at time T_0-δ_1 appears in the first sample of the ADC and DDC data packets. The split between the Timestamp and Timestamp Adjustment fields is left up to the system designer. One option is to have the Timestamp set to T_0 and the Timestamp Adjustment set to -δ_1. Another option is for the Timestamp to be T_0-δ_1 and the Timestamp Adjustment set to 0. This second option does not work for cases where the Timestamp is sent in a sample count format but the delay adjustment is not an integer number of samples.

The bottom three traces in Figure B.8-4 show the samples and packets in actual time, which includes the buffering and propagation delays of the ADC and DDC. These traces are shown only to show the difference between signal time and actual time. The buffering and propagation delays of digital signals do not need to be considered when determining the Timestamps and Timestamp Adjustments.

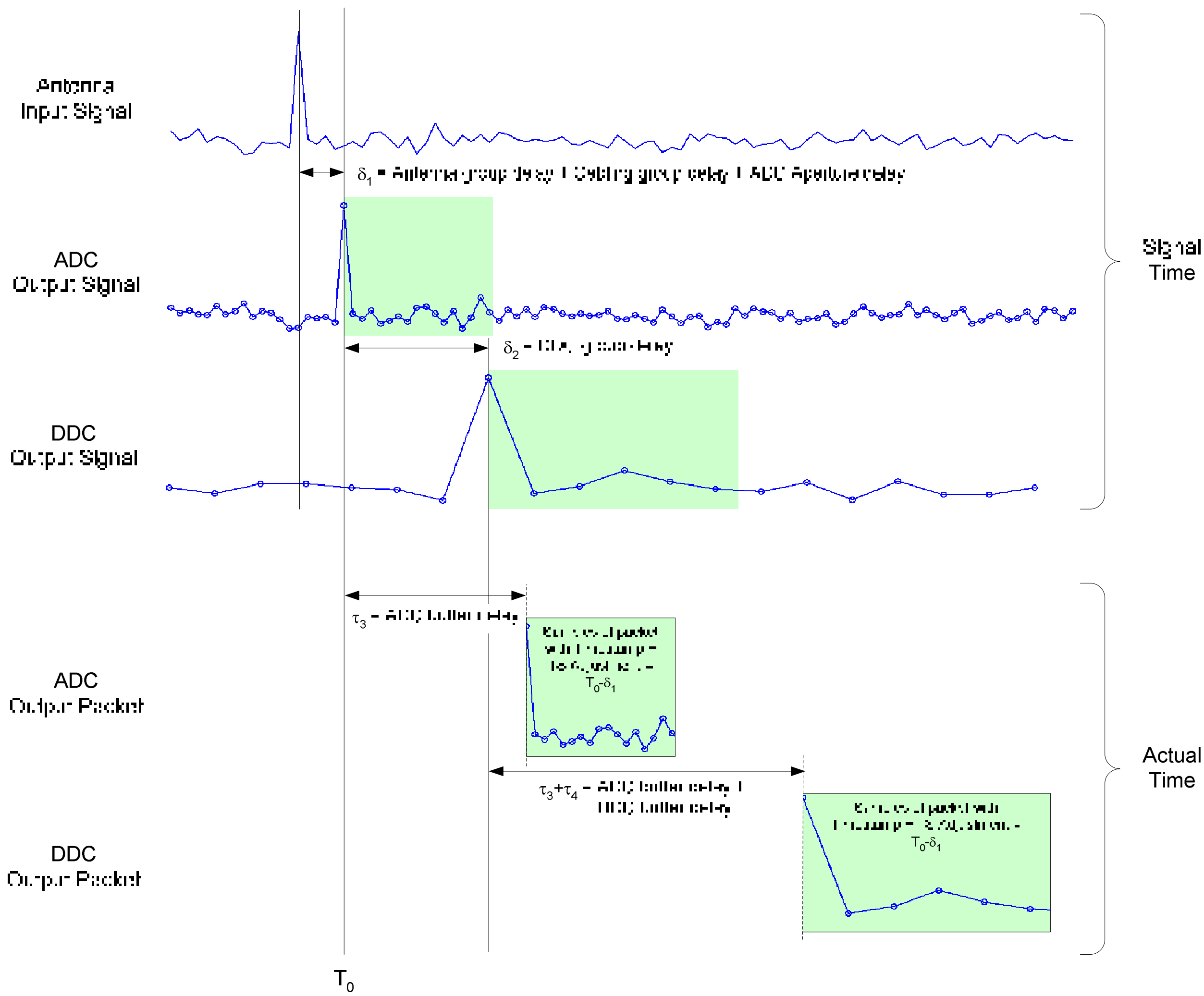


Figure B.8-4 : Signals with respect to Signal Time and Actual Time

The shaded boxes indicate the samples included in a particular IF Data packet.

Figure B.8-5 shows another option for the same example system. Here the first sample of the ADC and DDC data packets both start at time T_0 in signal time. This is the same as the previous example for the ADC packet so the Timestamp plus Timestamp Adjustment for the ADC data packet is still T_0-δ_1. However, with this packet formation the Timestamp plus Timestamp Adjustment for the DDC data packet must now account for the DDC's group delay and be set to T_0-δ_1-δ_2.

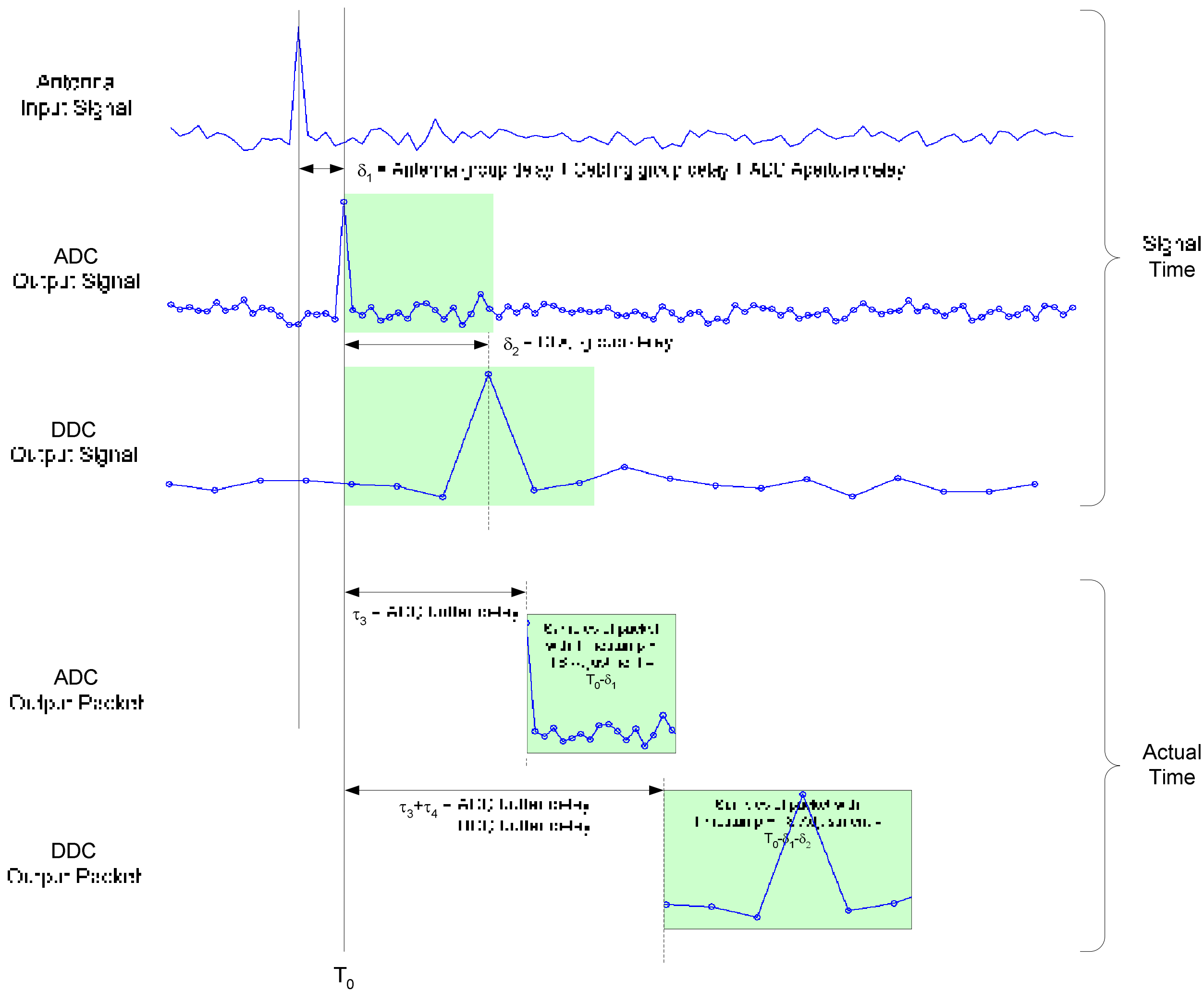


Figure B.8-5 : Signals with respect to Signal Time and Actual Time

The shaded boxes indicate the samples included in a particular IF Data packet.

B.9 *Data Packet Payload Format Examples*

This section presents several examples of the Data Packet Payload Format field in an IF Context packet and shows the corresponding packing of the Data Packet payload. There are a very large number of permutations of payload formats. The aim of this section is to show the use of each subfield within the Data Packet Payload Format rather than show every permutation of subfields.

Figure B.9-1 shows the organization of the Data Packet Payload Format field, described in Section 7.1.5.18, which is a useful reference when analyzing the format codes given in this section.

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	Pack	Real/ Cmplx		Data Item Format					Rpt	Event- Tag Size			Channel- Tag Size				Reserved				Item Packing Field Size						Data Item Size					
2	Repeat Count																Vector Size															

Figure B.9-1: Data Packet Payload Format Field

Figure B.9-2 shows a specific Data Packet Payload Field for a real, 32-bit, signed, fixed-point payload. This field contains the hexadecimal words 0x000007DF and 0x00000000. This matches the first example shown in Figure B.9-3(a).

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	Pack	Real/ Cmplx		Data Item Format					Rpt	Event- Tag Size			Channel- Tag Size				Reserved				Item Packing Field Size						Data Item Size					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1
2	Repeat Count																Vector Size															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure B.9-2: Data Packet Payload Format Field for Real 32-bit, Signed, Fixed-point Payload

The figures in the remainder of this section have two parts. The (a) part at the top of each figure contains valid values of the Data Packet Payload Format field for a variety of Data Item Formats. The (b) part at the bottom of each figure shows the first few Item Packing fields in the resulting Data packet payload.

Figure B.9-3 through Figure B.9-8 show the payload packing for the common 32-, 16-, and 8-bit data item sizes and both real and complex formats.

Data Packet Payload Format	Real/Complex	Format
000007DF 00000000	Real	Signed Fixed Pt.
100007DF 00000000	Real	Unsigned Fixed Pt.
0E0007DF 00000000	Real	Single-Precison Floating Pt.

(a)

<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">Data Item 1 (Sample 1)</td></tr>
<tr><td colspan="32">Data Item 2 (Sample 2)</td></tr>
<tr><td colspan="32">Data Item 3 (Sample 3)</td></tr>
<tr><td colspan="32">Data Item 4 (Sample 4)</td></tr>
</table>

(b)

Figure B.9-3: 32-bit, Real Format

Data Packet Payload Format	Real/Complex	Format
200007DF 00000000	Complex Cartesian	Signed Fixed Pt.
2E0007DF 00000000	Complex Cartesian	Single-Precison Floating Pt.
400007DF 00000000	Complex Polar	Signed Fixed Pt.
4E0007DF 00000000	Complex Polar	Single-Precison Floating Pt.

(a)

<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">Data Item 1 (Sample 1 Real/Ampl)</td></tr>
<tr><td colspan="32">Data Item 2 (Sample 1 Imag/Phase)</td></tr>
<tr><td colspan="32">Data Item 3 (Sample 2 Real/Ampl)</td></tr>
<tr><td colspan="32">Data Item 4 (Sample 2 Imag/Phase)</td></tr>
</table>

(b)

Figure B.9-4: 32-bit, Complex Format

Data Packet Payload Format	Real/Complex	Format
000003CF 00000000	Real	Signed Fixed Pt.
100003CF 00000000	Real	Unsigned Fixed Pt.

(a)

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
Data Item 1 (Sample 1)																Data Item 2 (Sample 2)															
Data Item 3 (Sample 3)																Data Item 4 (Sample 4)															
Data Item 5 (Sample 5)																Data Item 6 (Sample 6)															
Data Item 7 (Sample 7)																Data Item 8 (Sample 8)															

(b)

Figure B.9-5: 16-bit, Real Format

Data Packet Payload Format	Real/Complex	Format
200003CF 00000000	Complex Cartesian	Signed Fixed Pt.
400003CF 00000000	Complex Polar	Signed Fixed Pt.

(a)

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
Data Item 1 (Sample 1 Real/Ampl)																Data Item 2 (Sample 1 Imag/Phase)															
Data Item 3 (Sample 2 Real/Ampl)																Data Item 4 (Sample 2 Imag/Phase)															
Data Item 5 (Sample 3 Real/Ampl)																Data Item 6 (Sample 3 Imag/Phase)															
Data Item 7 (Sample 4 Real/Ampl)																Data Item 8 (Sample 4 Imag/Phase)															

(b)

Figure B.9-6: 16-bit, Complex Format

Data Packet Payload Format	Real/Complex	Format
000001C7 00000000	Real	Signed Fixed Pt.
100001C7 00000000	Real	Unsigned Fixed Pt.

(a)

31..24	23..16	15..8	7..0
Data Item 1 (Sample 1)	Data Item 2 (Sample 2)	Data Item 3 (Sample 3)	Data Item 4 (Sample 4)
Data Item 5 (Sample 5)	Data Item 6 (Sample 6)	Data Item 7 (Sample 7)	Data Item 8 (Sample 8)
Data Item 9 (Sample 9)	Data Item 10 (Sample 10)	Data Item 11 (Sample 11)	Data Item 12 (Sample 12)
Data Item 13 (Sample 13)	Data Item 14 (Sample 14)	Data Item 15 (Sample 15)	Data Item 16 (Sample 16)

(b)

Figure B.9-7: 8-bit, Real Format

Data Packet Payload Format	Real/Complex	Format
200001C7 00000000	Complex Cartesian	Signed Fixed Pt.
400001C7 00000000	Complex Polar	Signed Fixed Pt.

(a)

31..24	23..16	15..8	7..0
Data Item 1 (Sample 1 Real/Ampl)	Data Item 2 (Sample 1 Imag/Phase)	Data Item 3 (Sample 2 Real/Ampl)	Data Item 4 (Sample 2 Imag/Phase)
Data Item 5 (Sample 3 Real/Ampl)	Data Item 6 (Sample 3 Imag/Phase)	Data Item 7 (Sample 4 Real/Ampl)	Data Item 8 (Sample 4 Imag/Phase)
Data Item 9 (Sample 5 Real/Ampl)	Data Item 10 (Sample 5 Imag/Phase)	Data Item 11 (Sample 6 Real/Ampl)	Data Item 12 (Sample 6 Imag/Phase)
Data Item 13 (Sample 7 Real/Ampl)	Data Item14 (Sample 7 Imag/Phase)	Data Item 15 (Sample 8 Real/Ampl)	Data Item 16 (Sample 8 Imag/Phase)

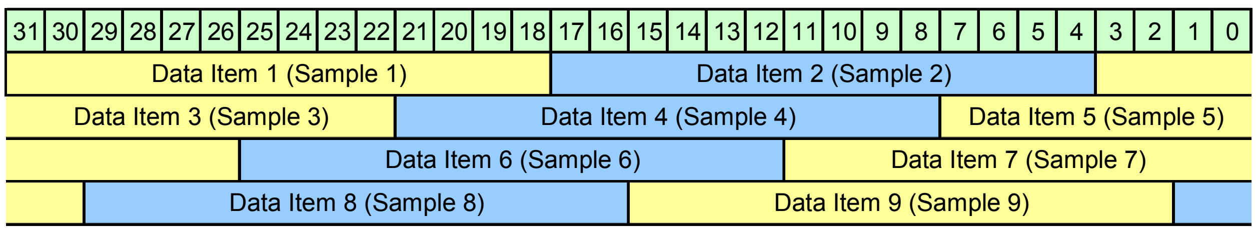
(b)

Figure B.9-8: 8-bit, Complex Format

The next three examples illustrate several methods of link-efficient and processing-efficient payload packing. Figure B.9-9 shows 14-bit real data with link-efficient packing, Figure B.9-10 shows 14-bit data with processing-efficient packing, and Figure B.9-11 shows 14-bit data placed in 16-bit packing fields. This last example has the same payload packing regardless of whether link- or processing-efficient packing was specified. Note that all packing fields are left-justified so unused bits appear at the lsbs of the 32-bit word or the lsbs of the packing field.

Data Packet Payload Format	Real/Complex	Format
8000034D 00000000	Real	Signed Fixed Pt.
9000034D 00000000	Real	Unsigned Fixed Pt.

(a)

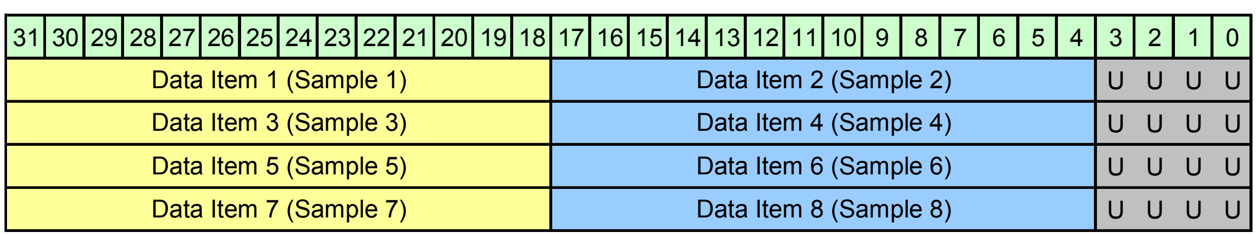


(b)

Figure B.9-9: Link-efficient Packing

Data Packet Payload Format	Real/Complex	Format
0000034D 00000000	Real	Signed Fixed Pt.
1000034D 00000000	Real	Unsigned Fixed Pt.

(a)



(b)

Figure B.9-10: Processing-efficient Packing

Data Packet Payload Format	Real/Complex	Format
000003CD 00000000	Real	Signed Fixed Pt.
100003CD 00000000	Real	Unsigned Fixed Pt.

(a)

<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="14">Data Item 1 (Sample 1)</td><td>U</td><td>U</td><td colspan="14">Data Item 2 (Sample 2)</td><td>U</td><td>U</td></tr>
<tr><td colspan="14">Data Item 3 (Sample 3)</td><td>U</td><td>U</td><td colspan="14">Data Item 4 (Sample 4)</td><td>U</td><td>U</td></tr>
<tr><td colspan="14">Data Item 5 (Sample 5)</td><td>U</td><td>U</td><td colspan="14">Data Item 6 (Sample 6)</td><td>U</td><td>U</td></tr>
<tr><td colspan="14">Data Item 7 (Sample 7)</td><td>U</td><td>U</td><td colspan="14">Data Item 8 (Sample 8)</td><td>U</td><td>U</td></tr>
</table>

(b)

Figure B.9-11: 14-bit Data Items in 16-bit Packing Fields

Figure B.9-12 illustrates the use of Event Tags. In this example 8-bit data is packed into 10-bit packing fields. The two remaining bits are declared as Event Tags, denoted with 'E.' This example also illustrates processing-efficient packing for 10-bit packing fields, where 3 packing fields can fit in a single 32-bit word.

Data Packet Payload Format	Real/Complex	Format
00200247 00000000	Real	Signed Fixed Pt.
10200247 00000000	Real	Unsigned Fixed Pt.

(a)

<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="8">Data Item 1
(Sample 1)</td><td>E</td><td>E</td><td colspan="8">Data Item 2
(Sample 2)</td><td>E</td><td>E</td><td colspan="8">Data Item 3
(Sample 3)</td><td>E</td><td>E</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 4
(Sample 4)</td><td>E</td><td>E</td><td colspan="8">Data Item 5
(Sample 5)</td><td>E</td><td>E</td><td colspan="8">Data Item 6
(Sample 6)</td><td>E</td><td>E</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 7
(Sample 7)</td><td>E</td><td>E</td><td colspan="8">Data Item 8
(Sample 8)</td><td>E</td><td>E</td><td colspan="8">Data Item 9
(Sample 9)</td><td>E</td><td>E</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 10
(Sample 10)</td><td>E</td><td>E</td><td colspan="8">Data Item 11
(Sample 11)</td><td>E</td><td>E</td><td colspan="8">Data Item 12
(Sample 12)</td><td>E</td><td>E</td><td>U</td><td>U</td></tr>
</table>

(b)

Figure B.9-12: Event Tags

Figure B.9-13 demonstrates the use of sample vectors. In this example 8-bit data is simultaneously collected from four sources. The first samples from each source are grouped at the beginning of the payload, followed by the group of second samples, and so on. This forms a 4-dimensional sample vector.

Data Packet Payload Format	Real/Complex	Format
000001C7 00000003	Real	Signed Fixed Pt.
100001C7 00000003	Real	Unsigned Fixed Pt.

(a)

<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="8">Data Item 1
(Component 1, Sample 1)</td><td colspan="8">Data Item 2
(Component 2, Sample 1)</td><td colspan="8">Data Item 3
(Component 3, Sample 1)</td><td colspan="8">Data Item 4
(Component 4, Sample 1)</td></tr>
<tr><td colspan="8">Data Item 5
(Component 1, Sample 2)</td><td colspan="8">Data Item 6
(Component 2, Sample 2)</td><td colspan="8">Data Item 7
(Component 3, Sample 2)</td><td colspan="8">Data Item 8
(Component 4, Sample 2)</td></tr>
<tr><td colspan="8">Data Item 9
(Component 1, Sample 3)</td><td colspan="8">Data Item 10
(Component 2, Sample 3)</td><td colspan="8">Data Item 11
(Component 3, Sample 3)</td><td colspan="8">Data Item 12
(Component 4, Sample 3)</td></tr>
<tr><td colspan="8">Data Item 13
(Component 1, Sample 4)</td><td colspan="8">Data Item 14
(Component 2, Sample 4)</td><td colspan="8">Data Item 15
(Component 3, Sample 4)</td><td colspan="8">Data Item 16
(Component 4, Sample 4)</td></tr>
</table>

(b)

Figure B.9-13: Sample Vector

Figure B.9-14 shows the use of Channel Tags to label the samples from four asynchronous channels packed into a single Data Packet Stream. This example is similar to the sample vector in example shown in Figure B.9-13 except that each 8-bit sample is placed in a 10-bit packing field with the remaining two bits used as Channel Tags.

Data Packet Payload Format	Real/Complex	Format
00020247 00000000	Real	Signed Fixed Pt.
10020247 00000000	Real	Unsigned Fixed Pt.

(a)

<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="8">Data Item 1
(Component 1, Sample 1)</td><td>0</td><td>0</td><td colspan="8">Data Item 2
(Component 2, Sample 1)</td><td>0</td><td>1</td><td colspan="8">Data Item 3
(Component 3, Sample 1)</td><td>1</td><td>0</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 4
(Component 1, Sample 2)</td><td>0</td><td>0</td><td colspan="8">Data Item 5
(Component 3, Sample 2)</td><td>1</td><td>0</td><td colspan="8">Data Item 6
(Component 1, Sample 3)</td><td>0</td><td>0</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 7
(Sample 1, Component 2)</td><td>0</td><td>1</td><td colspan="8">Data Item 8
(Component 1, Sample 4)</td><td>0</td><td>0</td><td colspan="8">Data Item 9
(Component 4, Sample 1)</td><td>1</td><td>1</td><td>U</td><td>U</td></tr>
<tr><td colspan="8">Data Item 10
(Component 3, Sample 3)</td><td>1</td><td>0</td><td colspan="8">Data Item 11
(Component 2, Sample 3)</td><td>0</td><td>1</td><td colspan="8">Data Item 12
(Component 1, Sample 5)</td><td>0</td><td>0</td><td>U</td><td>U</td></tr>
</table>

(b)

Figure B.9-14: Channel Tags for Asynchronous Channels

Figure B.9-15 demonstrates Channel Repeating with a 4-dimensional sample vector where several consecutive samples from the one source are grouped and packed in the payload before the same samples from the next source are grouped and packet in the payload. In this example 4 consecutive samples from each source are repeated.

Data Packet Payload Format	Real/Complex	Format
000001C7 00030003	Real	Signed Fixed Pt.
100001C7 00030003	Real	Unsigned Fixed Pt.

(a)

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
Data Item 1 (Component 1, Sample 1)	Data Item 2 (Component 1, Sample 2)	Data Item 3 (Component 1, Sample 3)	Data Item 4 (Component 1, Sample 4)
Data Item 5 (Component 2, Sample 1)	Data Item 6 (Component 2, Sample 2)	Data Item 7 (Component 2, Sample 3)	Data Item 8 (Component 2, Sample 4)
Data Item 9 (Component 3, Sample 1)	Data Item 10 (Component 3, Sample 2)	Data Item 11 (Component 3, Sample 3)	Data Item 12 (Component 3, Sample 4)
Data Item 13 (Component 4, Sample 1)	Data Item 14 (Component 4, Sample 2)	Data Item 15 (Component 4, Sample 3)	Data Item 16 (Component 4, Sample 4)
Data Item 17 (Component 1, Sample 5)	Data Item 18 (Component 1, Sample 6)	Data Item 19 (Component 1, Sample 7)	Data Item 20 (Component 1, Sample 8)
Data Item 21 (Component 2, Sample 5)	Data Item 22 (Component 2, Sample 6)	Data Item 23 (Component 2, Sample 7)	Data Item 24 (Component 2, Sample 8)

(b)

Figure B.9-15: Channel Repeating

Figure B.9-16 shows Sample Component Repeating for complex data formats where the real, or amplitude, components for several consecutive samples are packed in the payload before the imaginary, or phase, components are packed for the same samples. This example demonstrates a sample component repeat count of four.

Data Packet Payload Format	Real/Complex	Format
208001C7 00030000	Complex Cartesian	Signed Fixed Pt.
408001C7 00030000	Complex Polar	Signed Fixed Pt.

(a)

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
Data Item 1 (Sample 1 Real/Ampl)	Data Item 2 (Sample 2 Real/Ampl)	Data Item 3 (Sample 3 Real/Ampl)	Data Item 4 (Sample 4 Real/Ampl)
Data Item 5 (Sample 1 Imag/Phase)	Data Item 6 (Sample 2 Imag/Phase)	Data Item 7 (Sample 3 Imag/Phase)	Data Item 8 (Sample 4 Imag/Phase)
Data Item 9 (Sample 5 Real/Ampl)	Data Item 10 (Sample 6 Real/Ampl)	Data Item 11 (Sample 7 Real/Ampl)	Data Item 12 (Sample 8 Real/Ampl)
Data Item 13 (Sample 5 Imag/Phase)	Data Item 14 (Sample 6 Imag/Phase)	Data Item 15 (Sample 7 Imag/Phase)	Data Item 16 (Sample 8 Imag/Phase)

(b)

Figure B.9-16: Sample Component Repeating for Complex Data Formats

B.10 *ECEF and Relative Ephemeris Example*

This section presents an example where the Relative Ephemeris from one IF Context packet is combined with the ECEF Ephemeris from another IF Context packet. For simplicity this example is limited to two dimensions.

Consider the aircraft platform shown in Figure B.10-1. The location of an antenna on a wing of this aircraft is known in the Relative Ephemeris Coordinate System (RECS). The origin of the RECS is chosen to be at the location of a GPS antenna near the rear of the aircraft. The x′ axis is directed along the heading of the aircraft. The y′ axis is at a right angle to the x′ axis in the plane of the aircraft. The antenna in question is located at the point P′=(x_1′, y_1′) in the RECS. The antenna itself is directed at an angle α_1′ from the x′ axis. In this example, non-primed variables (such as x and α) refer to axes or coordinates in the ECEF Coordinate System (ECS) and primed variables (such as x′and α′') refer to axes or coordinates in the RECS.

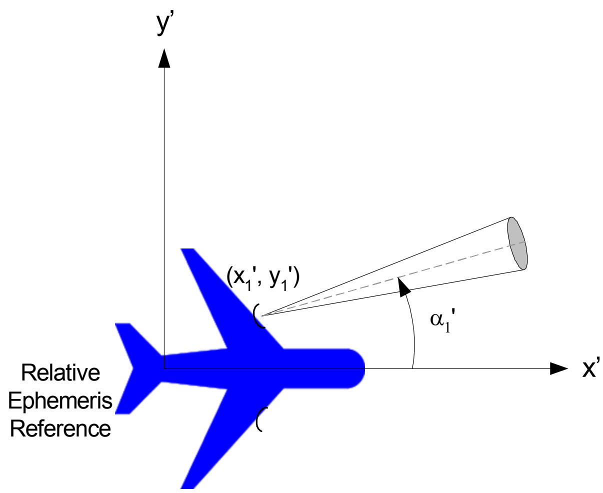


Figure B.10-1: Location and Attitude of Antenna in the Relative Ephemeris Coordinate System

The antenna is at coordinates (x_1′, y_1′) with respect to the aircraft GPS.

The GPS antenna is a convenient location for the RECS origin because the location of the GPS antenna can be determined in the ECS. Consider the arrangement shown in Figure B.10-2 where the origin of the ECS is at the center of the earth. Because this is a two-dimensional example, the aircraft is assumed to be in the plane of the earth's equator. The GPS antenna is located at the point P_0=(x_0, y_0) in the ECS and the aircraft heading is at an angle of θ with respect to the prime meridian. With this information the location of the antenna on the aircraft wing can be calculated with the proper transformation equations. For this two-dimensional example the transformation equations can be expressed as:

$$\begin{array}{ccccc} P & = & T(P_0) & R(\theta) & T(-P_0) & P' \end{array}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

where T() is a translation matrix and R() is a rotation matrix. This equation rotates the point P′=(x′, y′) an angle of θ about the point P_0=(x_0, y_0), where P and P′ are the location of the antenna in question in the ECS and RECS, respectively, and P_0 is the location of the GPS in the ECS. Also, the direction of the wing antenna in the ECS is:

$$\alpha = \theta + \alpha'$$

The above equations relate the location and attitude of an item in the RECS to the location and attitude of an item in the ECS. To solve for the ECS coordinates of the first wing antenna, x_1', y_1', and α_1' are substituted for x', y', and α' in the above equations. The same transformation equations could be used for other antennas on the same aircraft by substituting in their coordinates, x_k', y_k', and α_k'.

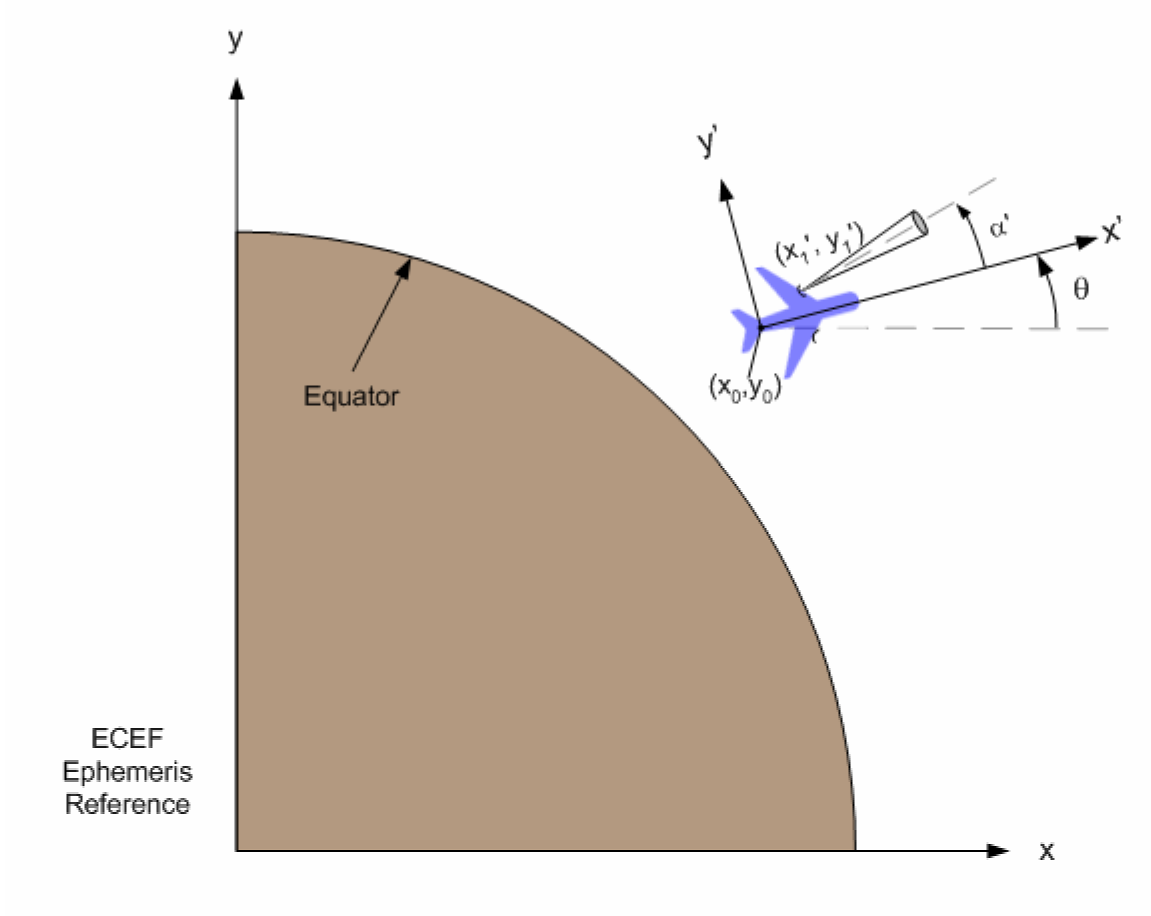


Figure B.10-2: Location and Attitude of Antenna in the ECEF Coordinate System

The GPS of the aircraft is at (x_0, y_0) and the aircraft has a heading of θ.

This aircraft platform carries the VRT system shown in Figure B.10-3. Here the coordinates of the antenna in the RECS are given in the IF Context packet of the antenna. The coordinates for the GPS and heading of the aircraft in the ECS are given in the IF Context packet for the GPS. To link the two coordinate systems together, the Antenna IF Context Packet Stream also contains the Ephemeris Reference ID field that contains the Stream ID of the GPS. The Information Class documentation should include the transformation between the RECS and ECS when it is necessary for the user's application.

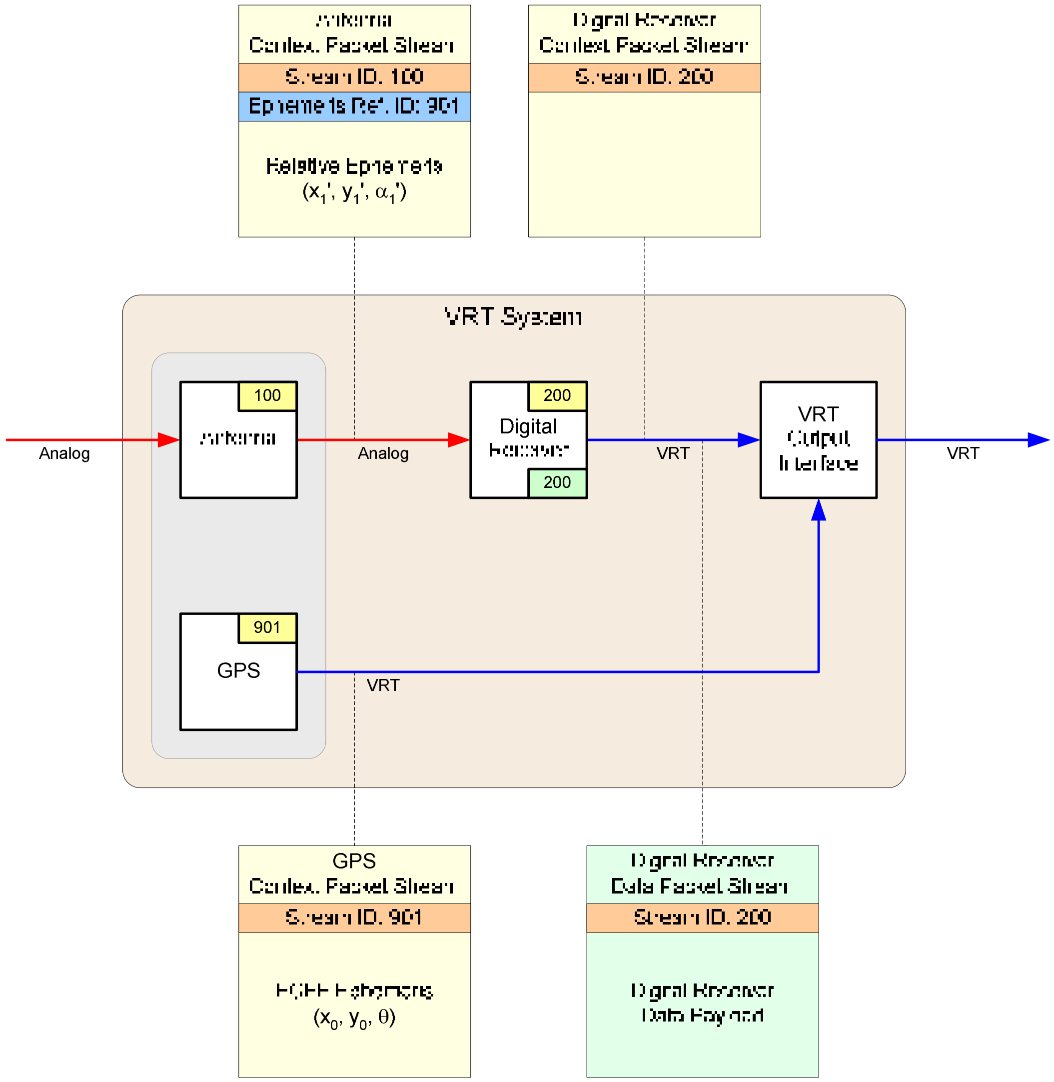


Figure B.10-3: Block Diagram for ECEF and Relative Ephemeris Example

The Ephemeris Reference ID field of the Antenna contains a value of 901, the Stream ID of the GPS.

B.11 *Source Context Association List Example*

The example in this section illustrates the use of the Source Context Association List also referred to as the Source List. Figure B.11-1 shows a system with three cascaded VRT processes: a digital receiver, a DDC, and a demodulator. The receiver is the source of the data for the DDC and the DDC is the input source for the demodulator.

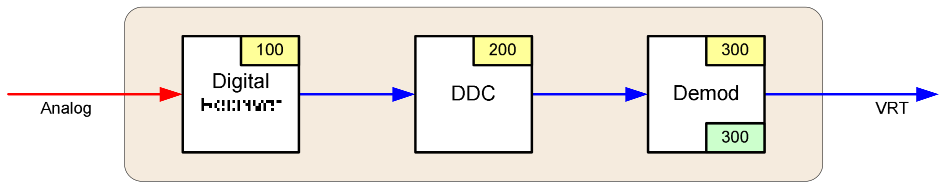


Figure B.11-1: Block Diagram for Source List Example

Clearly the settings of the receiver, DDC, and demodulator affect the interpretation of the data at the output of the demodulator. Therefore, it is reasonable for an Information Stream for this system to contain the demodulator's Data Packet Stream and the Context Packet Streams of the upstream processes.

Figure B.11-2 shows the content of the Data and Context packets in this Information Stream. The Stream ID for the demodulator Context Packet Stream, 300, matches the Stream ID for the demodulator Data Packet Stream. Therefore the demodulator Context Packet Stream is paired to the demodulator Data Packet Stream.

The demodulator Context Packet Stream also contains an entry of 200, the Stream ID of the DDC Context Packet Stream, in its Source List shown in blue. In this way the Context Packet Stream of the DDC is directly associated with the demodulator's Context Packet Stream and indirectly associated with the demodulator's Data Packet Stream. Also, the receiver's Context Packet Stream is indirectly associated with the demodulator's Data Packet Stream through the Source List in the DDC's Context packets. The collection of the Data Packet Stream and associated Context Packet Streams comprises the Information Stream.

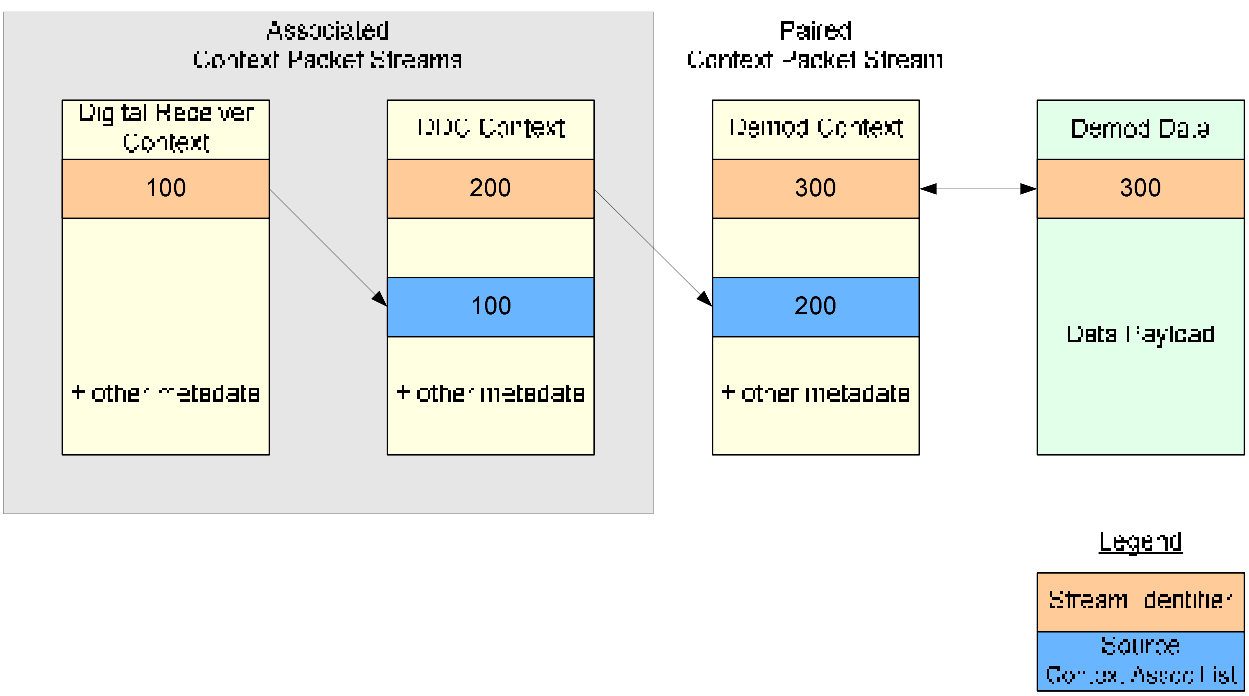


Figure B.11-2: Representation of Source Context Associations

The associations in the demodulator Information Stream can also be represented with the association diagram shown in Figure B.11-3. Here the contents of each Context Packet Stream are not shown, only the associations between different Packet Streams.

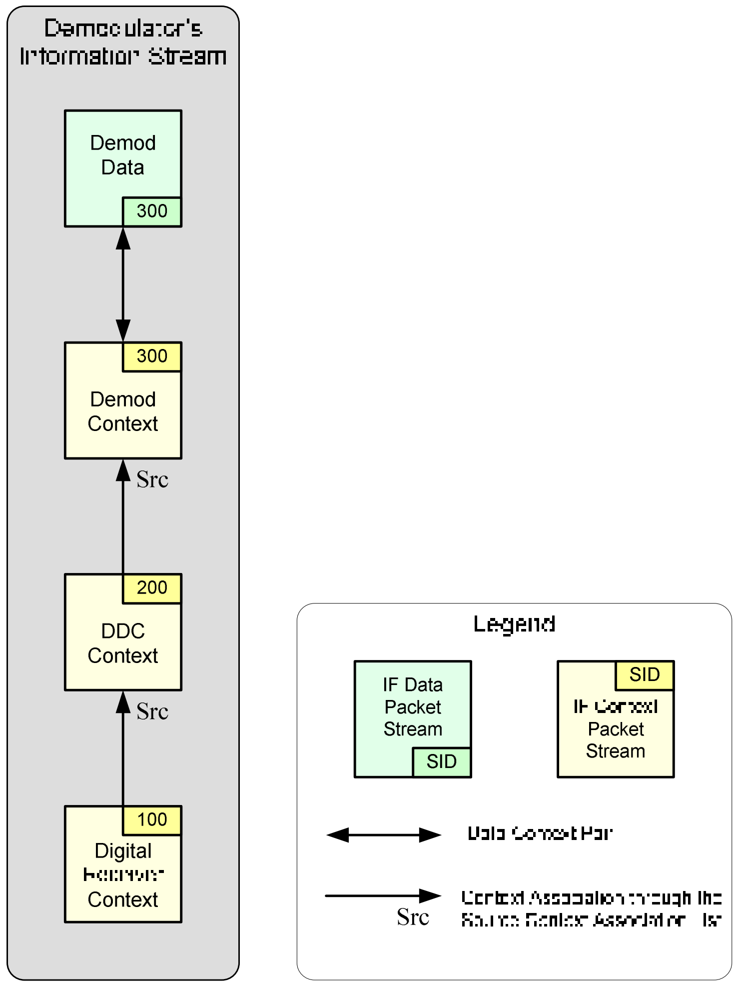


Figure B.11-3: Association Diagram for the Source Context Association List Example

The demodulator Data Packet Stream forms the head of this Information Class. The demodulator Context Packet Stream is paired with the Data Packet Stream as indicated by the double-headed arrow and the matching Stream Identifiers of 300. The DDC's Context is associated to this Information Stream by being included in the Source List of the demodulator's Context packets. This is represented by the single-headed arrow with the 'Src' for Source. Similarly, the receiver's Context is associated with the Information Stream by being included in the Source List of the DDC's Context packets.

B.12 *System Context Association List Example*

The example in this section illustrates the use of the System Context Association List also referred to as the System List. The Source and Vector Component Lists are used to associate Context that is directly related to the signal path. The System List is used to associate Context that is typically not in the signal path and can't be associated using the other Context Association Lists.

Figure B.12-1 shows a system where GPS equipment is co-located with a digital receiver and a temperature sensor that measures the heat in a DDC card. The GPS is related to the receiver and the temperature sensor is related to the DDC through the System Lists. Context packets from all of the components in this system pass through all downstream processes so that the Context Packet Streams all appear on the output link of the demodulator.

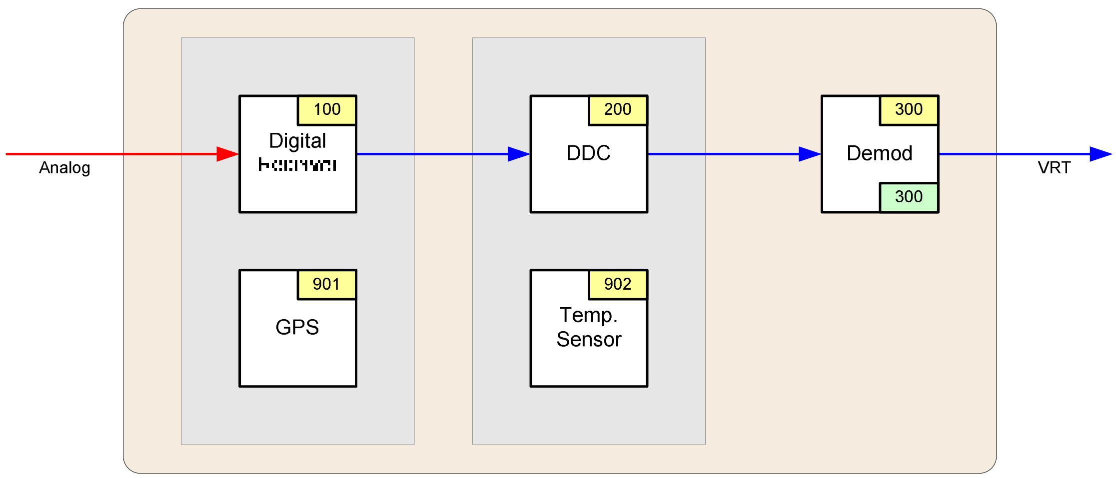


Figure B.12-1: Block Diagram for System List Example

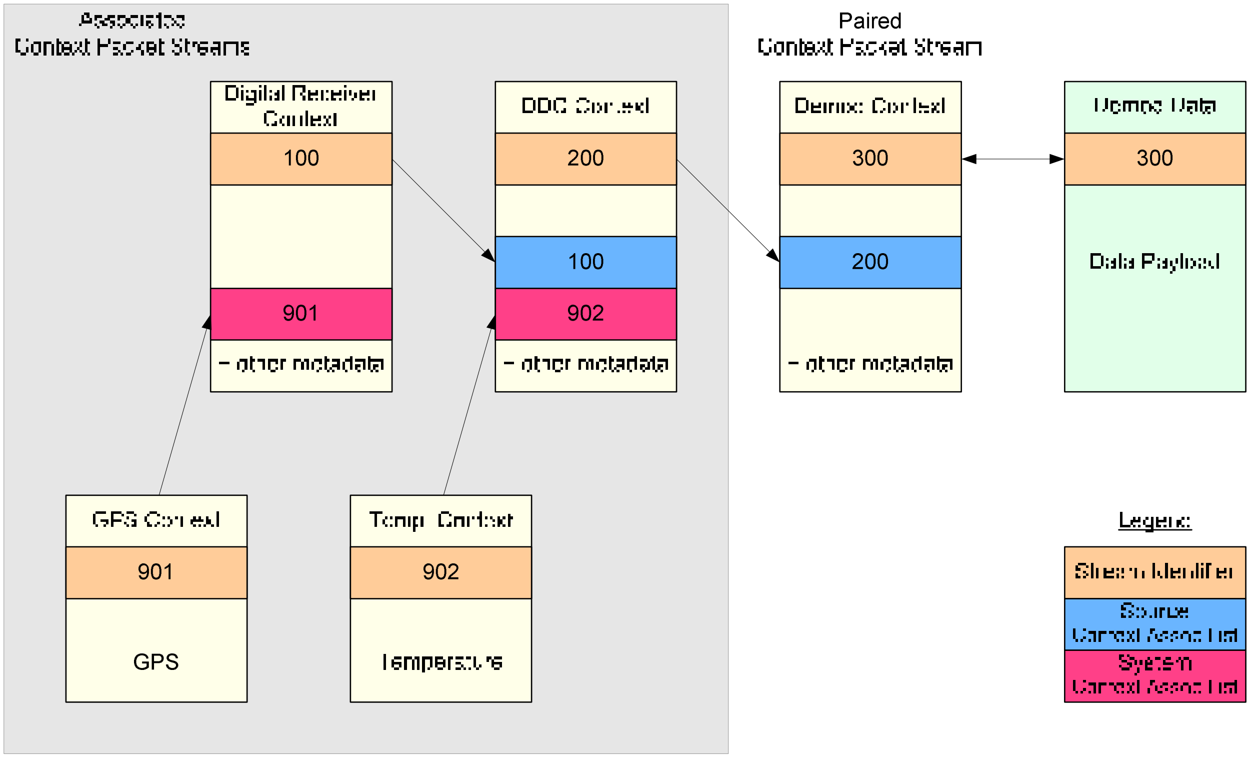


Figure B.12-2: Representation of System Context Associations

Figure B.12-2 shows the content of the Data and Context Packet Streams in the demodulator's Information Stream. The receiver's Context Packet Stream contains a System List that includes the Stream ID of the GPS's Context Packet Stream, 901. The Context of the temperature sensor is associated with the DDC in the same fashion. The Source List is also used as described in B.11.

The association diagram for this system is shown in Figure B.12-3. The GPS and temperature Context Packet Streams are associated with the Information Stream through the System Context Association List represented by the single-headed arrow marked with a 'Sys'. The Source List is used to associate Context for the processes in the signal path.

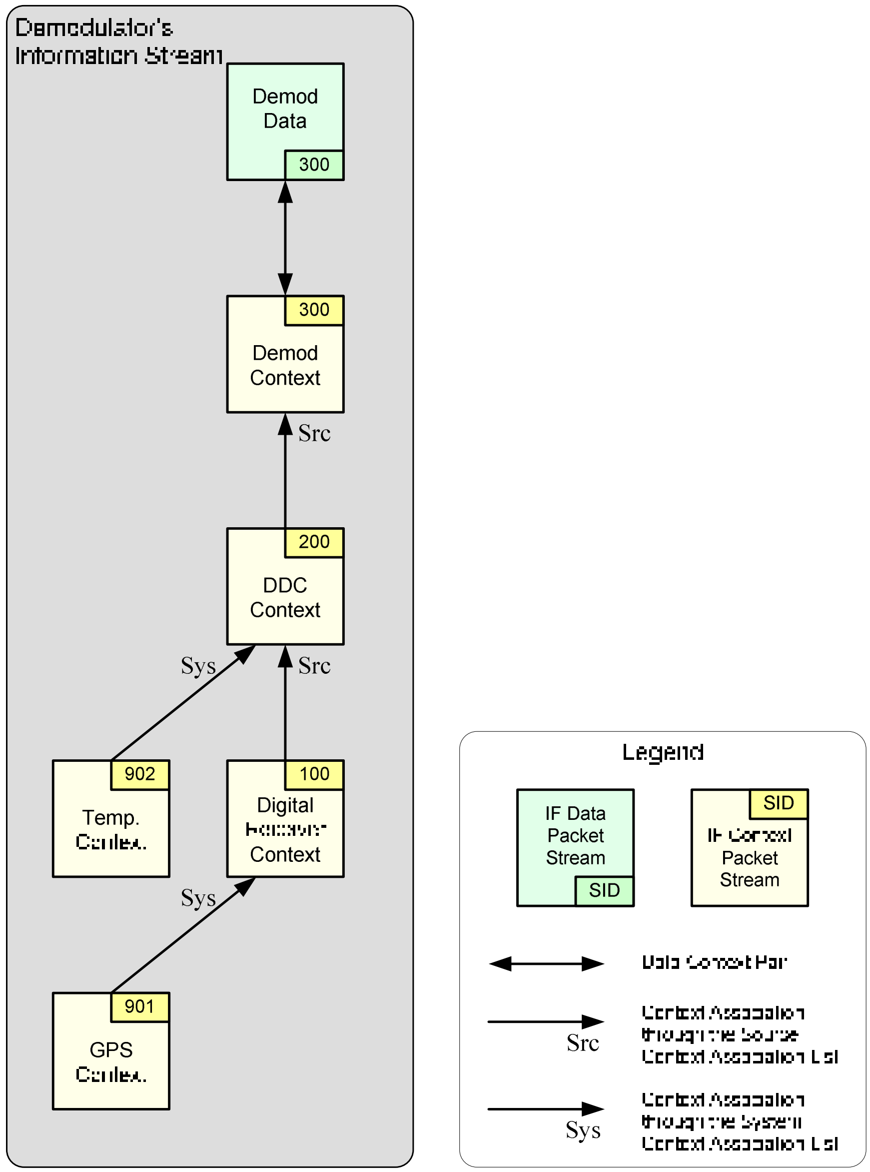


Figure B.12-3: Association Diagram of System List Example

Note that separate Information Streams for the digital receiver and DDC may also be described but are not described in this example.

B.13 *Vector-component Context Association List Example*

The example in this section illustrates the use of the Vector-component Context Association List also referred to as the Vector Component List. Figure B.13-1 shows a system with a digital receiver followed by a 64-channel channelizer.

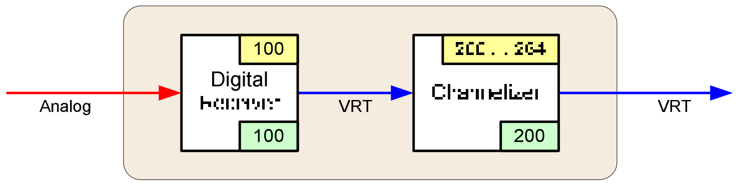


Figure B.13-1: Block Diagram for Vector Component Example

The channelizer creates an IF Data Packet Stream where the output of each of the 64 channels is interleaved following the rules in Section 6.1.5.3. This vector Data Packet Stream has a Stream ID of 200. The channelizer also generates a paired IF Context Packet Stream with the same Stream ID of 200. This Context Packet Stream can contain Context information that is common to all of the 64 channels such as Reference Level or Bandwidth, if applicable.

More importantly, for this example the channelizer Context Packet Stream contains the Vector-component Context Association List. This list contains the ordered Stream IDs for the Context Packet Streams for each of the 64-channels. This associates a unique Context Packet Stream to each of the 64 vector components of the Data Packet Stream. The Context Packet Streams for the component channels have Stream IDs from 201 to 264. Each contains Context information specific to that channel such as RF Reference Frequency Offset. Because the receiver is the input source for each of the channels of the channelizer, each channel's Context Packet Stream contains a Source List carrying the stream ID of the receiver's Context Packet Stream. The contents of the Data and Context Packet Streams for this system are shown in
Figure B.13-2.

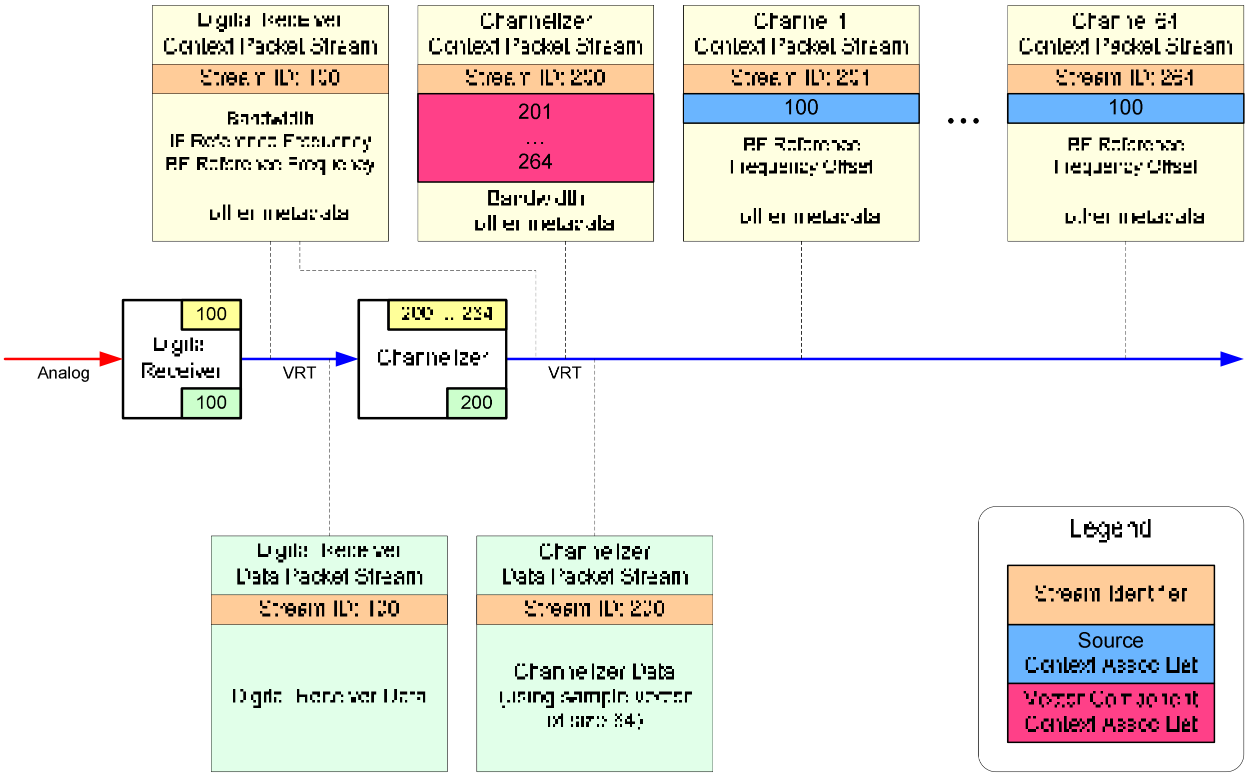


Figure B.13-2: Context Packet Contents for the Vector Component List Example

The association diagram for the digital receiver and channelizer Information Streams are shown in Figure B.13-3. Each vector component of the channelizer is associated to the common channelizer through the Vector-component Context Association List, represented by the single-headed arrow marked with a 'Vec'. Each of the component channels also associates with the receiver through the Source List, marked with a 'Src.'

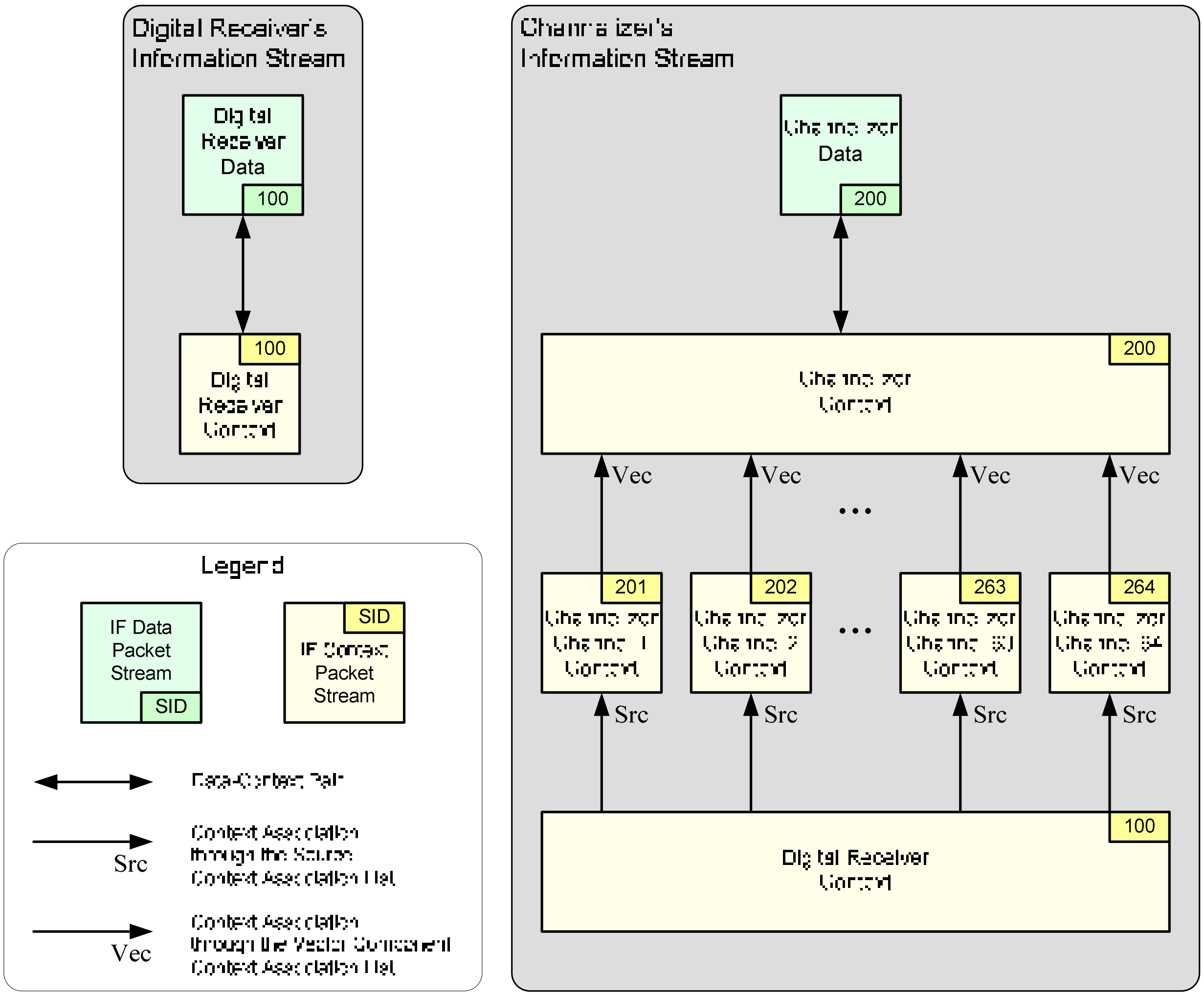


Figure B.13-3: Association Diagrams for the Vector Component List Example

Note that the digital receiver's Context Packet Stream appears in both Information Streams

B.14 *Asynchronous-Channel Context Association List Example*

The example in this section illustrates the use of the Asynchronous-Channel Context Association List, also referred to as the Asynchronous-Channel List. Figure B.14-1 shows a system with a digital receiver followed by a bank of DDCs. The DDCs in this system are individually configurable so each may have a unique bandwidth, sample rate, and tuning frequency.

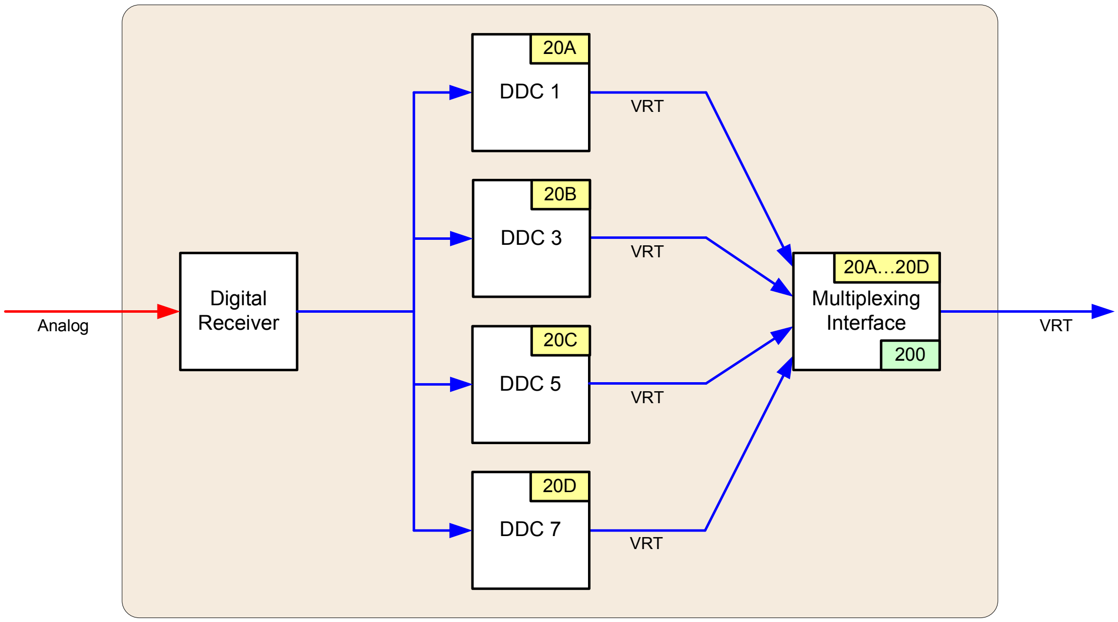


Figure B.14-1: Block Diagram for the Asynchronous-Channel List Example

The odd-numbered channels multiplexed into a single data packet stream with channel tags appended to each data item to identify the data source.

The samples from the first four odd-numbered DDCs are multiplexed over a single physical interface. This example demonstrates the case where multiple data channels are interleaved into a single Data Packet Stream[*]. Because they have different sample rates they cannot be packed into a Sample Vector as they were in the example of Appendix B.13. Instead, the samples from each channel are asynchronously multiplexed into the Data Packet Stream and Channel Tag bits are added to each Data Item to identify the sample's source.

Figure B.14-2 shows that the settings for each of the DDCs are sent over the interface in Context Packet Streams with Stream IDs 20A, 20B, 20C, and 20D for DDC1, 3, 5, and 7, respectively. The asynchronous data samples are sent in a Data Packet Stream with Stream ID 200. The paired Context Packet Stream, also with Stream ID 200, acts as an intermediary to associate the Context for the individual DDC channels with the Data Packet Stream.

---

[*] *Another method of doing this is to send four separate Data Packet Streams and their paired Context Packet Streams. In this example, however, it is assumed that this is not a viable option because it requires an excessive amount of memory to buffer data from multiple DDC channels while a Data Packet from another DDC channel is being sent.*

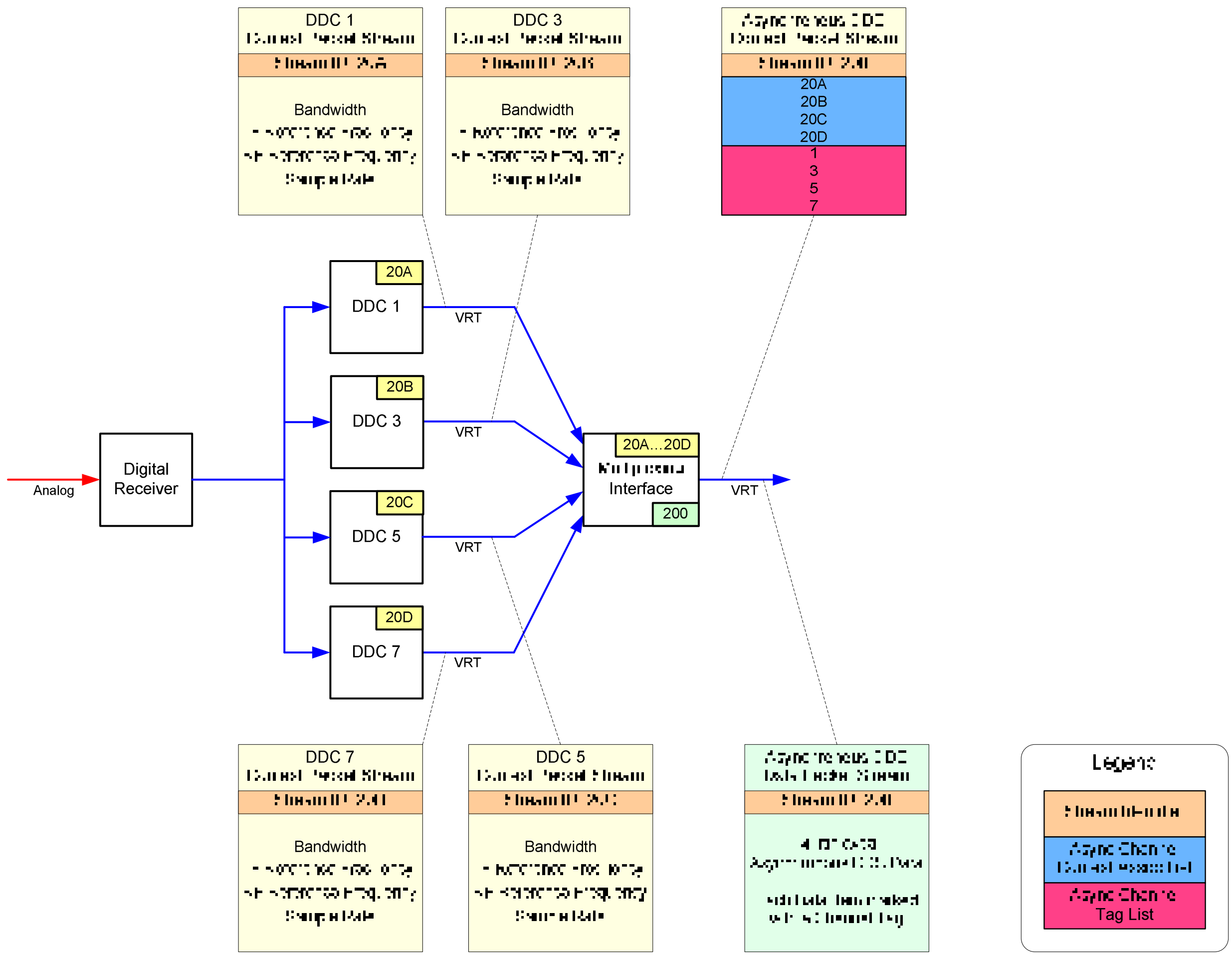


Figure B.14-2: Context Packet Contents for the Asynchronous Channel List Example.

The Context Packet Stream paired with the Asynchronous Data Packet Stream contains ordered lists of the Channel Tags used in the Data Packet Streams and the Stream IDs of the individual DDC Context Packet Streams for those channels.

In this example, the Asynchronous DDC Context Packet Stream contains an Asynchronous-Channel Context Association List and Asynchronous-Channel Tag List as shown in Figure B.14-2 and Figure B.14-3. These lists have the same length and the order of the entries in each list must match so that the data from each channel is associated with the correct individual DDC Context Packet Stream.

For example, the third entry in the Asynchronous-Channel Tag List is Channel Tag 5. Therefore the third entry in the Asynchronous-Channel Context Association List must be the Stream ID for the Context Packet Stream for DDC 5, which is 20C. The same holds for the other entries in each list.

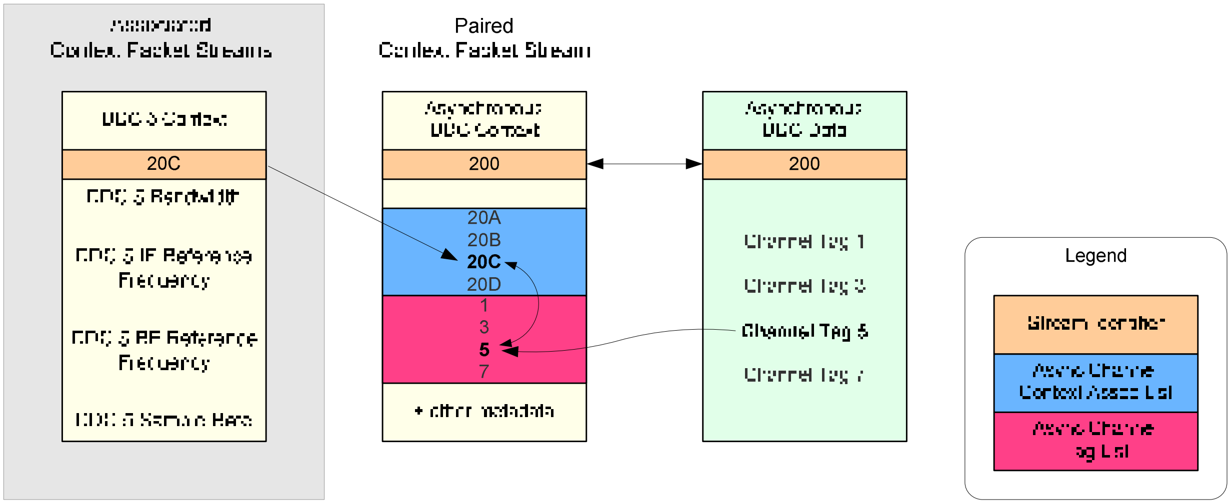


Figure B.14-3: Representation of an Asynchronous-Channel Context Association
The paired Context Packet Stream contains lists which hold the Stream IDs and Channel Tags for the DDC channels.

The association diagram for the Asynchronous DDC Information Stream is shown in Figure B.14-4. Each individual DDC channel is associated to the common DDC Context Packet Stream through the Asynchronous-Channel Context Association List and Asynchronous-Channel Tag List, represented by the single-headed arrow marked with a 'Chan'

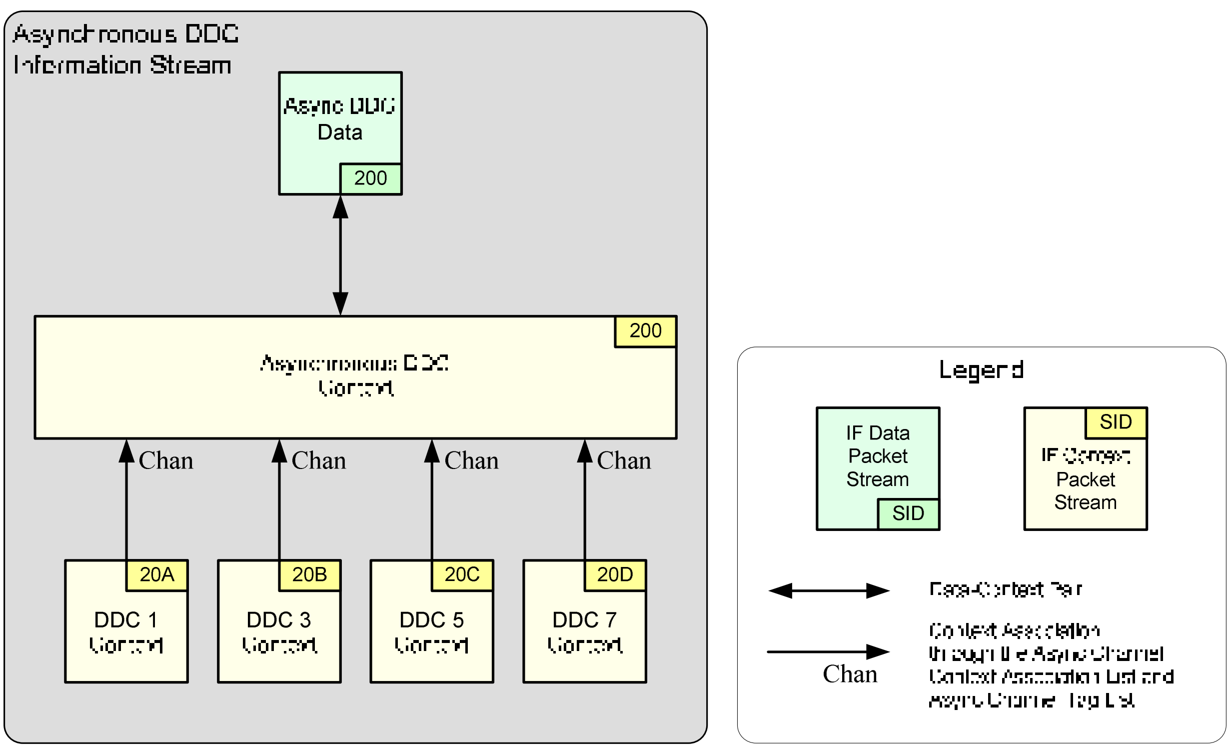


Figure B.14-4: Association Diagrams for the Asynchronous Channel List Example

Appendix C Extension Packet Streams Example

This section describes an example where FFT data is conveyed using Extension Data packets and metadata about the FFT samples is conveyed with Extension Context packets. Figure C-1 and Figure C-2 show how a user might define the format of the FFT Extension Data Packet and FFT Extension Context Packet Classes.

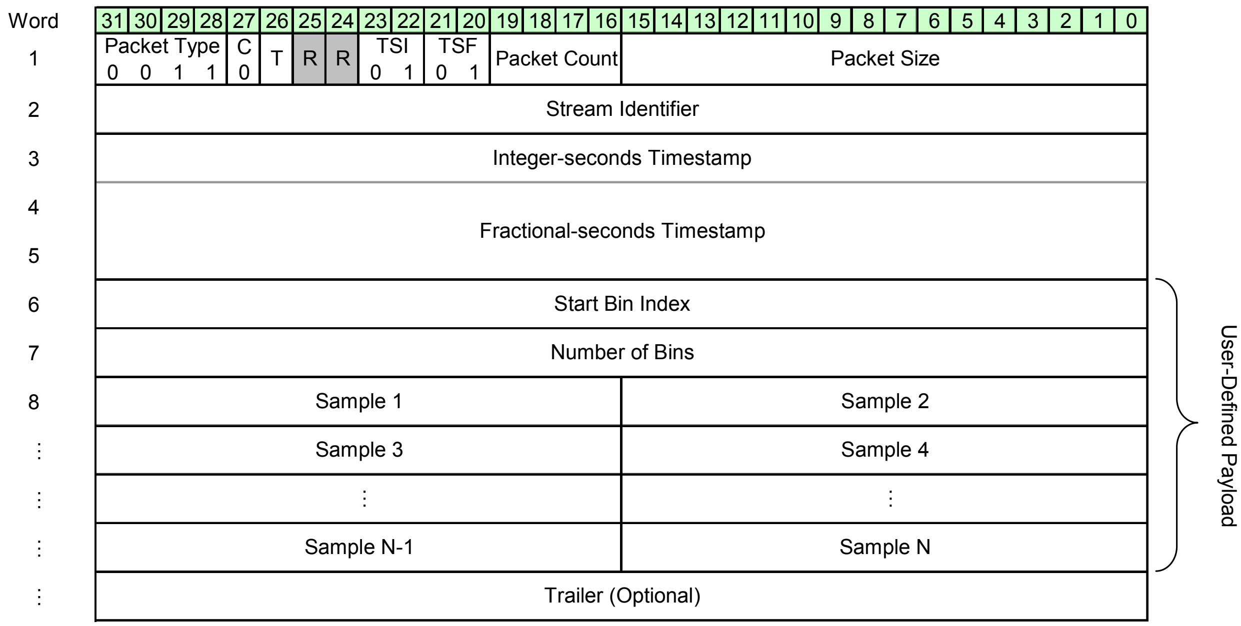


Figure C-1: Format of the FFT Extension Data Packet Example

The format of both the Extension Packet Classes follow the structure of the IF Data and IF Context packets to allow reuse of transmitter and receiver code.

The Packet Type field is set to 0011, the code for an Extension Data packet with Stream Identifier. The 'C' bit is fixed at 0, indicating that the Class ID field is never sent. The 'T' bit is not fixed, indicating the intermittent inclusion of the Trailer. The TSI and TSF bits are fixed to 01 and 01, indicating a Timestamp with UTC integer part and sample count fractional part is included. The Packet Count and Packet Size fields are defined the same as for a normal IF Data packet.

The Timestamp is defined as the time of the first time-domain sample that was included in the particular FFT contained in the packet.

The User-defined Payload section starts at word 6 and is specialized for an application where a very large FFT is performed on the input samples, but the large FFT block is split into smaller contiguous blocks for transmission to a signal processor or group of processors. Each FFT data packet includes fields for the index of the first bin sent in this packet and the total number of bins sent in this packet. The FFT bin samples themselves are sent as 16-bit signed fixed-point numbers.

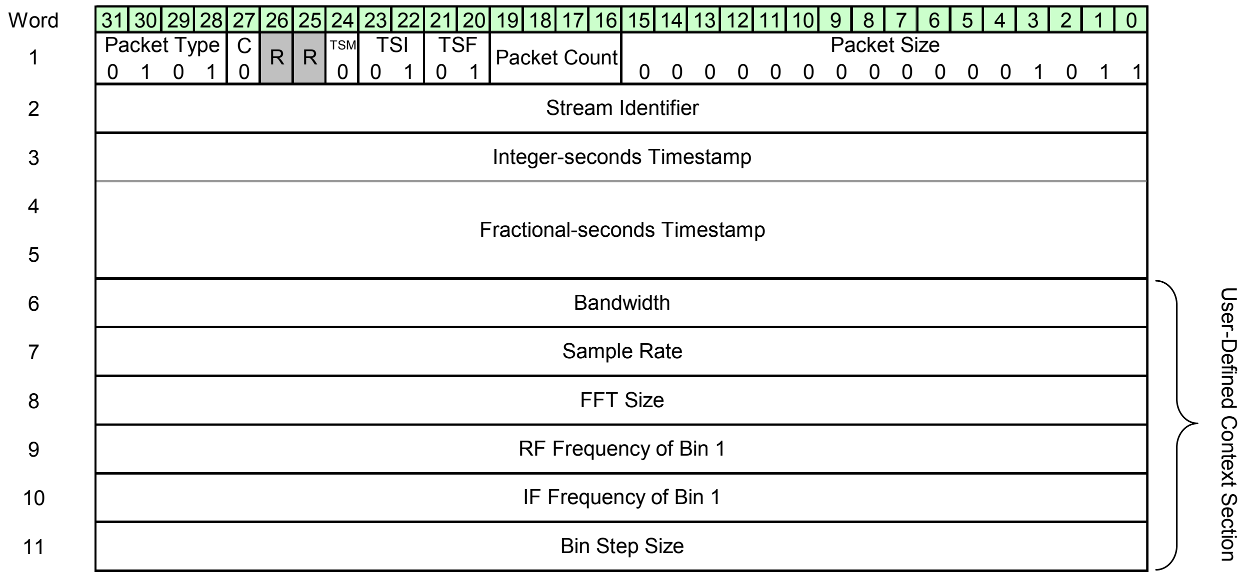


Figure C-2: Format of the FFT Extension Context Packet Example

The Extension Context packet for the example FFT application contains six Context fields. They are included in every Context packet so no Context Indicator Field is necessary.

The Packet Type for this Packet Stream is 0101, the code for an Extension Context packet. The C, TSI, and TSF fields are the same as for the Data packet. The TSM bit is fixed at 0, indicating that the Context applies to the entire Data packet, rather than to a specific sample in the Data packet. The Packet Size is fixed at eleven words.

The Stream ID is always included, and because the FFT Extension Data Packet and Context Packet Streams are paired, the Stream IDs for these two classes are the same. The Timestamp definition is the same as for the Data packet.

The Bandwidth and Sample Rate Context fields have the same definition as the IF Context packet Context fields but the remaining Context fields are specialized for this application. The FFT Size gives the size of the very large FFT before it was broken into smaller pieces for transmission. The next two Context fields give the RF frequency and IF frequency of the first FFT bin and the last Context field gives the frequency step size between consecutive FFT bins.

Appendix D VRT Floating-point Numbers

The VRT specification provides a family of "VRT" floating-point number formats for the purpose of conveying IF Data. This family encompasses a range of exponent and mantissa sizes. These numbers bridge the gap between fixed-point numbers, which have limited dynamic range, and IEEE floating-point numbers, which come only in 32-bit and 64-bit sizes. It should be noted that the VRT number format was not invented for VRT. It was already in use in commercial DDC chips for some time. VRT simply expands on the options by providing a range of exponent and mantissa sizes. This allows an option to be chosen that obtains the required performance for an application while minimizing required link-layer bandwidth.

A VRT floating-point number can have a mantissa size ranging from one to 63 bits and an exponent size of one to six bits. The only other condition is that the total size, *N*, of the number must be no more than 64 bits. As a general rule the exponent size, *E*, is determined by the required dynamic range, while the mantissa size, *M*, is determined by the maximum quantization noise allowed. Like IEEE 754 floating-point numbers, VRT floating-point numbers provide increased resolution near zero amplitude, and decreasing resolution as amplitude increases.

VRT floating-point numbers are formatted with the exponent in the least-significant bit positions. This is different from in IEEE 754 numbers, and indeed from in most floating-point formats, which put the exponent in the most-significant positions. The reason VRT numbers hold the exponent in the least-significant positions is so that the number can be converted to its equivalent fixed-point value without the additional work of masking off the exponent field. The reason for this is that when treated as the lsbs of a fixed-point number the misinterpretation of the exponent bits adds noise that is below the noise floor created by the resolution of the mantissa.

Every VRT floating-point number is interpreted as representing an equivalent fixed-point number. The range of the equivalent fixed-point number for unsigned VRT floating-point numbers, is from zero to slightly less than one, as will be demonstrated.* Similarly, for a signed VRT floating-point number the range of the equivalent fixed-point number is from minus one to slightly less than one.

The operation of VRT floating-point numbers is depicted in Figure D-1 for 5-bit VRT numbers with a 2-bit exponent and a 3-bit mantissa. As the figure shows, the value of the unsigned exponent indicates how far to left-shift the mantissa, with a fill of zeros, to get an equivalent fixed-point number.

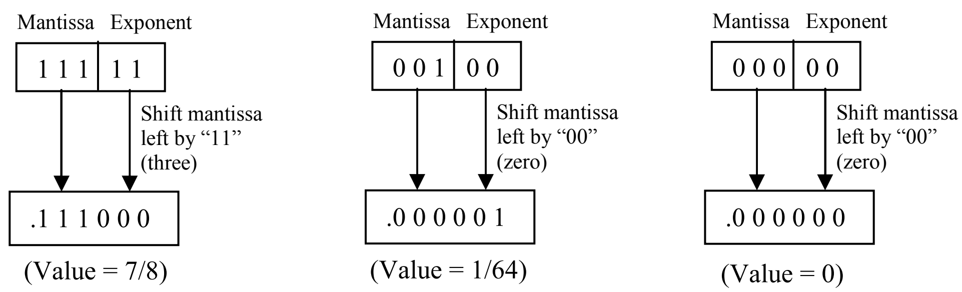


Figure D-1: The Operation of 5-bit Unsigned VRT Floating-point Numbers.
The exponent indicates how far to left-shift the mantissa. The radix point is to the left of the msb for the largest possible VRT floating-point number.

The largest value possible for these numbers is "11111." The "11" exponent says to shift the "111" mantissa left by three. So the resulting fixed-point number is ".111000." The radix point is understood to be just left of the msb, so this number is equal to 7/8.

* *This is according to the Normalized Interpretation. See Section 6.1.6.4 for a definition of the Normalized Interpretation.*

The smallest nonzero value possible corresponds to "00100." The "00" exponent indicates that no left shift is performed to get the equivalent fixed-point value. Since the maximum value has six bits, this number must also have six bits, with the radix point to the far left, so the equivalent number is ".000001." This corresponds to a value of 1/64.

Table D-1 shows a few representative values for this 5-bit example to make the operation of VRT floating-point numbers clear.

VRT Number	Mantissa	Exponent	Left Shift	Equivalent Fixed-point	Value
11111	111	11	3	.111000	7/8
11110	111	10	2	.011100	7/16
11101	111	01	1	.001110	7/32
11100	111	00	0	.000111	7/64
10000	100	00	0	.000010	1/32
01000	010	00	0	.000010	1/32
00111	001	11	3	.001000	1/8
00110	001	10	2	.000100	1/16
00101	001	01	1	.000010	1/32
00100	001	00	0	.000001	1/64

Table D-1: Equivalent Fixed-point Values for a 5-bit Unsigned VRT Floating-point Numbers
(With a 3-bit Mantissa and a 2-Bit Exponent)

As the above table and previous examples show, the maximum value these numbers can take is 7/8 or $1\text{-}2^{3}$. Compare this to the maximum value of $1\text{-}2^{-5}$ possible with a 5-bit unsigned fixed-point number. This increased distance from a value of one is an example of the resolution lost when some bits are dedicated to an exponent. On the other hand, consider the smallest values possible. For the 5-bit fixed-point number it is 2^{-5}, whereas for the 5-bit VRT number being considered it is 2^{-6}. This demonstrates the advantage of VRT numbers. The difference in dynamic range increases as the exponent size, E, is increased. The cost is lost resolution in all but the smallest numbers.

Note that Table D-1 contains three VRT numbers that represent the number 1/32. This phenomenon occurs because there is no requirement to normalize VRT floating-point numbers. These numbers are included in the table to demonstrate this fact. There are also other duplicate values, not shown, for this 5-bit example. VRT numbers with larger mantissas have even more duplicate values. This is not a problem. It is simply a feature of the number format that should be noted.

VRT signed floating-point numbers operate in a manner similar to VRT unsigned floating-point numbers. The difference is that the mantissa is a two's complement value rather than unsigned. Figure D-2 demonstrates their operation.

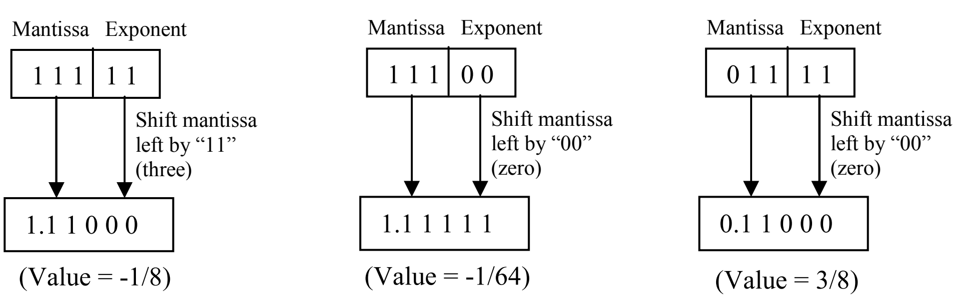


Figure D-2: The Operation of 5-bit Signed VRT Floating-point Numbers.
In contrast with unsigned numbers, here unused bits on the left replicate the sign bit.

There are two features worth noting in this example. First, since the range of the equivalent fixed-pount numbers is minus one to nearly one, the radix point in these numbers is understood to be just to the right of the msb rather than all the way to the left, as in the case of the unsigned VRT format. Second, when a mantissa is not left shifted by the maximum value possible, the sign is extended all the way to the left in the equivalent fixed-point number.

Table D-2 shows some representative numbers for the case of N = 5, M = 3, and E = 2. Note that this table also has duplicate values of -1/4.

VRT Number	Mantissa	Exponent	Left Shift	Equivalent Fixed-point	Value
01111	011	11	3	0.11000	+3/4
01110	011	10	2	0.01100	+3/8
01101	011	01	1	0.00110	+3/16
01100	011	00	0	0.00011	+3/32
00111	001	11	3	0.01000	+1/4
00110	001	10	2	0.00100	+1/8
00101	001	01	1	0.00010	+1/16
00100	001	00	0	0.00001	+1/32
11100	111	00	0	1.11111	-1/32
11101	111	01	1	1.11110	-1/16
11110	111	10	2	1.11100	-1/8
11111	111	11	3	1.11000	-1/4
10000	100	00	0	1.11100	-1/8
10001	100	01	1	1.11000	-1/4
10010	100	10	2	1.10000	-1/2
10011	100	11	3	1.00000	-1

Table D-2: Equivalent Fixed-point Values for a 5-bit Signed VRT Floating-point Number
(With a 3-bit Mantissa and a 2-Bit Exponent)