

# **S T A N D A R D**

---



## **Command Summary Specification**

**HCF\_SPEC-99, Revision 9.0**

**Release Date: 23 July, 2007**

**Release Date:** 23 July, 2007

**Document Distribution / Maintenance Control / Document Approval**

To obtain information concerning document distribution control, maintenance control, and document approval please contact the HART Communication Foundation (HCF) at the address shown below.

**Copyright © 1993 (Rev. 1997, 2001, 2006, 2007) HART Communication Foundation**

This document contains copyrighted material and may not be reproduced in any fashion without the written permission of the HART Communication Foundation.

**Trademark Information**

HART<sup>®</sup> is a registered trademark of the HART Communication Foundation, Austin, Texas, USA. Any use of the term HART hereafter in this document, or in any document referenced by this document, implies the registered trademark. WirelessHART<sup>™</sup> is a trademark of the HART Communication Foundation. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information contact the HCF Staff at the address below.



Attention: Foundation Director  
HART Communication Foundation  
9390 Research Boulevard  
Suite I-350  
Austin, TX 78759, USA  
Voice: (512) 794-0369  
FAX: (512) 794-3904

<http://www.hartcomm.org>

**Intellectual Property Rights**

The HCF does not knowingly use or incorporate any information or data into the HART Protocol Standards which the HCF does not own or have lawful rights to use. Should the HCF receive any notification regarding the existence of any conflicting Private IPR, the HCF will review the disclosure and either (a) determine there is no conflict; (b) resolve the conflict with the IPR owner; or (c) modify the standard to remove the conflicting requirement. In no case does the HCF encourage implementers to infringe on any individual's or organization's IPR.

## Table of Contents

Preface.....	7
Introduction.....	9
1. Scope .....	11
2. References.....	12
2.1 HART Field Communications Protocol Specifications.....	12
2.2 Related HART Documents .....	12
3. Definitions .....	12
4. Symbols/Abbreviations.....	13
5. Data Types .....	13
5.1 String Formats.....	14
5.1.1 Packed ASCII.....	14
5.1.2 ISO Latin-1 .....	16
5.2 Dates .....	16
5.3 Time .....	16
5.4 Floating-Point Formats .....	17
5.4.1 Normalized Numbers .....	18
5.4.2 Denormalized Numbers .....	18
5.4.3 Zeros .....	19
5.4.4 Infinities .....	19
5.4.5 Not-a-Numbers .....	20
5.5 Unsigned Integer Format .....	21
5.6 Signed Integer Format .....	22
5.7 Lookup Table Formats.....	23
5.7.1 Enumerations .....	24
5.7.2 Bit Fields.....	25
6. Field Device Revision Rules.....	26
7. Application Layer Interface.....	29
7.1 Command Number Partitions .....	29
7.2 Command Requirements.....	31
7.2.1 Autonomous & Asynchronous Requirements .....	31
7.2.2 Command Operations .....	31

7.2.3	Indexed Commands .....	32
7.2.4	Multi-Transaction Commands .....	32
7.3	Command Status Bytes .....	32
7.3.1	Communication Status .....	33
7.3.2	Response Code.....	34
7.3.3	Field Device Status .....	35
8.	Dynamic and Device Variables .....	38
8.1	Primary Variable (PV) .....	40
8.1.1	Voltage Mode Devices.....	41
8.2	Device Variable Classification .....	41
8.3	Device Families .....	41
8.4	Device Variable Status.....	43
9.	Field Device Identification .....	44
9.1	User Identification of the Field Device.....	46
9.2	Field Device Revisions .....	46
9.3	Identifying the Field Device's Command Set .....	47
10.	Host Conformance Classifications.....	47
10.1	Host Conformance Class 0.....	48
10.2	Host Conformance Class 1.....	48
10.3	Host Conformance Class 2.....	48
10.4	Host Conformance Class 3 - Generic Host.....	50
10.5	Host Conformance Class 4.....	51
10.6	Host Conformance Class 5 - Requirements for a "Universal Host" .....	51

Annex A.	Revision History .....	52
A.1.	Changes from Rev 7.1 to Rev 8.0 .....	52
A.2.	Changes from Rev 7.0 to Rev 7.1 .....	53
A.3.	Changes from Rev 6.0 - Final to Rev 7.0 - Final .....	53
A.4.	Major Modifications from Rev 5 to Rev 6.0 - Final .....	56
A.5.	Major Modifications from Rev 4 to Rev 5 .....	58
A.6.	Major Modifications from Initial Rev 3 to Rev 4 .....	58

## Table of Figures

Figure 1.	OSI 7-Layer Model .....	9
Figure 2.	Single and Double Precision Floating-Point Formats .....	17
Figure 3.	Normalized Numbers.....	18
Figure 4.	Denormalized Numbers.....	19
Figure 5.	Zeros.....	19
Figure 6.	Infinities.....	19
Figure 7.	NaNs .....	20
Figure 8	Loop Current Saturated versus Alarm Levels .....	37
Figure 9	Device Variables and Dynamic Variables.....	39
Figure 10.	Domains Accessed Using PV .....	40
Figure 11.	Device Variable Status Byte Format .....	43

## Index to Tables

Table 1. Packed ASCII Character Set.....	14
Table 2. Packed ASCII Characters versus 8-Bit Bytes.....	14
Table 3. ISO Latin-1 Characters .....	16
Table 4. Summary of Floating-Point Number Properties .....	18
Table 5. Unsigned Integer to Decimal Conversion.....	21
Table 6. Signed Integer to Decimal Conversion.....	22
Table 7. Single Byte Enumerated Table Format.....	24
Table 8. Single-Byte Bit Field Table Format .....	25
Table 9. HART Command Number Partitions .....	30
Table 10. Communication Status.....	33
Table 11. Response Code Classification.....	34
Table 12. Device Status .....	35
Table 13. Field Device Identifying Data .....	44
Table 14. Application of Identifying Data.....	45
Table 16. Host Conformance Classes .....	47
Table 17. Host Conformance Class 1 Commands .....	48
Table 18. Host Conformance Class 2 Commands .....	48
Table 19. Host Conformance Class 3 Commands .....	50
Table 20. Host Conformance Class 4 Commands .....	51

## Preface

This preface is included for informational purposes only.

The *Command Summary Specification* has long provided the basis for the HART Application Layer. Often, in the past, requirements that did not fit well in other Application Layer documents could be found in the *Command Summary Specification*. The assembly of diverse topics in this specification negatively impacted organization and readability. As a result, some implementations have overlooked requirements of this document.

With the introduction of HART 7 and WirelessHART™, the *Command Summary Specification* has been significantly updated. Previously Network and Transport Layer requirements have been embedded in various specification and a significant number of those requirements were found in this document (e.g., the old Section 10 "Network Management"). With the creation of the *Network Management Specification*, several sections have been moved from this transferred from this document. The section transferred to the *Network Management Specification* include the former Subsection 7.2 "Data Field Format" and Section 10 " Network Management"

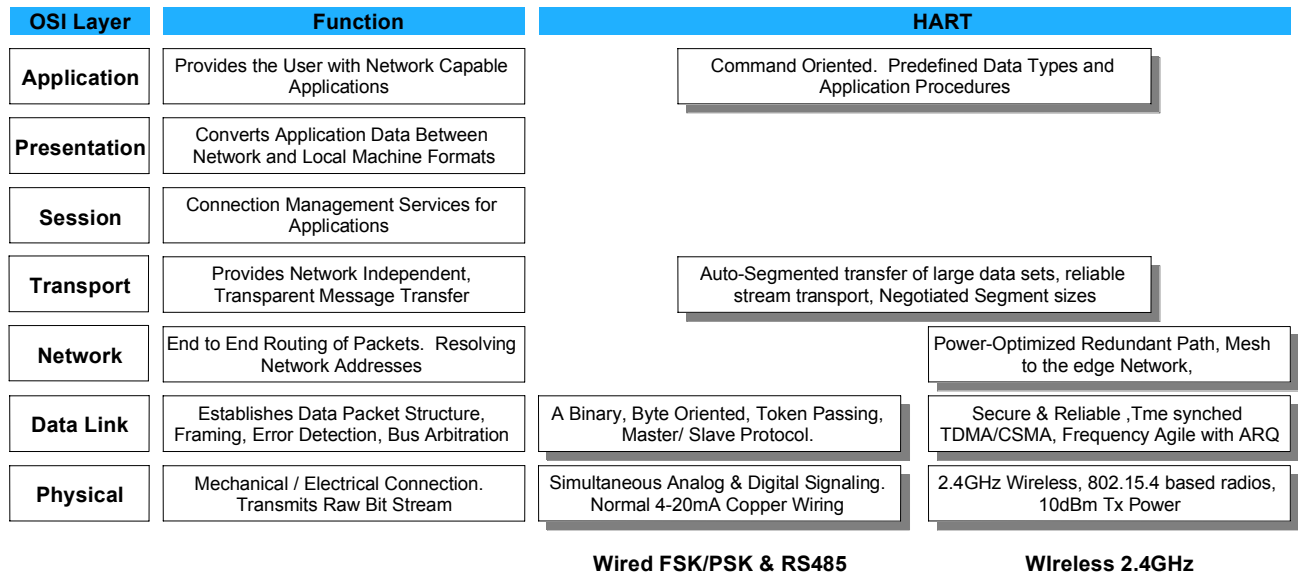
The former Subsections 10.2 "Sub-devices and I/O Systems" and Subsection 10.5 "Burst Mode Operation" have been transferred to the *Common Practice Command Specification*.

In addition to this reorganization there have been two significant enhancements: support for 16-bit Manufacturer ID has been included and the new data type, Time, has been added.



## Introduction

The Application Layer is the topmost layer in the Open System Interconnect (OSI) model for communications protocols. HART is a master/slave protocol loosely organized around the ISO/OSI 7-layer model (see Figure 1). The HART Application Layer defines device commands, responses, data types and status reporting. Conventions in HART, such as how to trim the loop current, are also part of the Application Layer.



**Figure 1. OSI 7-Layer Model**

The *Command Summary Specification* establishes the core requirements for implementation of the HART Protocol Application Layer by master and slave devices. In addition, procedures for utilizing application layer functions in HART networks are defined.

The *Command Summary* defines the data types that are allowed to be communicated using the Protocol. Several numeric data types are allowed including single and double precision floating point; integers and unsigned integers. Text strings may be transmitted using Packed-ASCII or ISO Latin-1. Look-up tables can be used to associate fixed meaning to an unsigned integer (see the *Common Tables Specification* for example usage of these data types). Any combination of the specified data types can be mixed in a HART command.

The *Command Summary* specifies rules governing all HART commands and allocates command numbers (e.g., as Universal or Common Practice). The allowed command behaviors are defined. Commands may read from or write to a field device and there are commands that require a field device to perform a designated action or function. The command requirements found in this specification are adhered to in the *Universal*, *Common Practice*, and *Device Family Command Specifications*. In addition, Device-Specific commands developed by manufacturers must comply to the requirements in this specification.

This specification defines requirements for slaves to provide continuous operational feedback to hosts. This feedback includes device health information; command execution reporting and (for data link layer use) communication error detection. The *Command Response Code Specification* builds on the requirements for command execution reporting and provides detailed information about individual Response Codes.

Finally, this specification includes requirements identifying the level of application layer support found in field devices and host applications. Slave devices are identified using their Manufacturer ID; Expanded Device Type; and Device Revision. This specification includes rules that govern the values returned for the Device Type and Device Revision when the field device's application layer support is changed or enhanced. Host applications are identified by the level of application layer support they offer. This allows end users to understand the capabilities they can expect from a host.

This key document in the HART Specifications specifies requirements that serve as the foundation for all other HART Application Layer specifications.

## 1. SCOPE

This specification provides the core Application Layer requirements for HART-compatible devices. The Application Layer builds on the requirements specified for the Network Layer. While the Network Link Layer is responsible for the error-free end-end transmission of messages, it does not interpret message content. The Application Layer is responsible for message content, including the definition of commands and the interpretation of data. Thus the message's core payload is the focus of the Application Layer.

This document specifies:

- The allowable formats of data transmitted via the Protocol;
- Revision rules for all field devices;
- The allocation of Commands Numbers for use by Universal, Common Practice, Device-Specific and Device Family commands;
- The requirements for the construction of any HART command;
- The Command Status Bytes required to be returned with all commands responses;
- The use of Dynamic and Field Device Variables; and
- Procedures used by masters to identify field devices and manage HART networks

These requirements provide the basis for all HART Application Layer Specifications.

## **2. REFERENCES**

### **2.1 HART Field Communications Protocol Specifications**

*HART Field Communications Protocol Specification.* HCF\_SPEC-12

*Data Link Layer Specification.* HCF\_SPEC-81

*Universal Command Specification.* HCF\_SPEC-127

*Common Practice Command Specification.* HCF\_SPEC-151

*Device Families Command Specification.* HCF\_SPEC-160

*Common Tables Specification.* HCF\_SPEC-183

*Command Response Code Specification.* HCF\_SPEC-307

### **2.2 Related HART Documents**

The HART Protocol Specifications frequently reference the manufacturers' device-specific document. Device-specific documents are developed and controlled by the respective manufacturer and should follow the requirements of the following HART Communication Foundation document:

*Requirements for Device-Specific Documentation.* HCF\_LIT-18

## **3. DEFINITIONS**

Definitions for terms can be found in *HART Field Communications Protocol Specification* (HCF\_SPEC-12). Terms used throughout the *Command Summary Specification* include: ASCII, Data Link Layer, Delayed Response, Delayed Response Mechanism, Device Variable, Busy, DR\_CONFLICT, DR\_DEAD, DR\_INITIATE, DR\_RUNNING, Dynamic Variable, Device Type, Device Revision, Fixed Current Mode, Floating Point, ISO Latin-1, Master, Multi-drop, Not-A-Number, Packed ASCII, Preamble, Request Data Bytes, Response Data Bytes, Response Message, Slave, Slave Time-Out, Time Constant, Units Code.

## 4. SYMBOLS/ABBREVIATIONS

<b>ADC</b>	Analog-to-Digital Converter
<b>DAC</b>	Digital-to-Analog Converter.
<b>DAQ</b>	Data Acquisition. This refers to a device specific ADC or DAC
<b>DR</b>	Delayed Response
<b>DRM</b>	Delayed Response Mechanism
<b>HCF</b>	HART Communication Foundation
<b>LSB</b>	Least Significant Byte. The LSB is always the last byte transmitted over a HART data link
<b>MSB</b>	Most Significant Byte. The MSB is always the first byte transmitted over a HART data link.
<b>NAN</b>	Not-a-Number.
<b>RC</b>	Response Codes
<b>UART</b>	Universal Asynchronous Receiver Transmitter

## 5. DATA TYPES

This section defines the data types supported by the HART Protocol. Devices claiming compatibility with the HART Protocol may only transmit these data types across a HART network:

- Strings using ISO Latin-1 or Packed ASCII Formats
- Date
- Time
- Single and Double Precision Floating-Point
- Signed Integers
- Unsigned Integers
- Table Based Data using Enumerations or Bit Fields

All multi-byte data must be transmitted sequentially starting from the most significant byte (MSB) and ending with the least significant byte (LSB). The length of all data items is fixed. Therefore, the length of a data item may not vary from one bus transaction to another.

Unsigned integers, enumerations and bit fields may be packed together. Various lengths of unsigned integers, enumerations and bit fields may be combined, provided the data (with null bits added as necessary) is always passed in multiples of eight bits.

## 5.1 String Formats

Two string formats are supported: ISO Latin-1 and Packed ASCII. Strings are always fixed in length.

### 5.1.1 Packed ASCII

Packed ASCII is a HART-specific 6-bit character code representing a subset of the ASCII character code set (see Table 1). Produced by compressing four Packed ASCII characters into three 8-bit bytes, Packed ASCII strings must be a multiple of 4 characters (3 bytes) and must be padded out to the end of the data item with space characters. For example, 4 space characters at the end of a string would appear as the 3 bytes: 0x82, 0x08, and 0x20.

Uninitialized Packed ASCII strings must be set to the question mark (?) character. Four question mark characters appear as the bytes: 0xFF, 0xFF, and 0xFF.

**Table 1. Packed ASCII Character Set**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Note: Most significant hexadecimal digit top to bottom; least significant left to right.

**Table 2. Packed ASCII Characters versus 8-Bit Bytes**

Packed ASCII Character	0	1	2	3
Packed ASCII Bit	5 0	5 0	5 0	5 0

Bit Position	7 0	7 0	7 0
Byte Number	0	1	2

Note: A field device does not need to unpack a Packed ASCII string unless the field device can display the string.

### Construction of Packed ASCII characters:

Constructing a Packed ASCII string is a simple matter of discarding the most significant two bits from each character and compressing the result:

1. Truncate Bit #6 and #7 of each ASCII character.
2. Pack four, 6 bit-ASCII characters into three bytes.
3. Repeat until the entire string is processed.

This algorithm can be implemented by masking and shifting four 6-bit characters into a 24 bit register then moving the three bytes into the Packed ASCII string.

**Reconstruction of ASCII characters:**

Unpacking Packed ASCII strings requires flipping some bits in addition to uncompressing the string itself. To unpack a Packed ASCII string:

1. Unpack the four, 6-bit ASCII characters.
2. For each character, place the complement of Bit 5 into Bit 6.
3. For each character, reset Bit 7 to zero.
4. Repeat until the entire string is processed.

This algorithm can be implemented by loading three bytes into a 24-bit register and shifting the four 6-bit characters into the string. Parse the resulting character to flip bit 6 as needed.

### 5.1.2 ISO Latin-1

ISO Latin-1 (ISO 8859-1) strings consist of one character per byte (see Table 3). ISO Latin-1 strings must be padded out to the end of the data item with zeros (0x00). One zero (0x00) at the end of a string ISO Latin-1 data item is not sufficient to meet this requirement. A valid ISO Latin-1 character is allowed in the final byte of an ISO Latin-1 data item. Compared strings (e.g., in Universal Command 21, Read Unique Identifier Associated With Long Tag) are sensitive to character case.

Uninitialized ISO Latin-1 strings must be set to the question mark (?) character.

**Table 3. ISO Latin-1 Characters**

20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
,	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
	í	í	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
ö	ñ	õ	ö	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

- Note: 1. 0x20 is a space character.  
2. 0xA0 is a non-breaking space character.

## 5.2 Dates

In the Protocol, dates are represented by three 8-bit binary unsigned integers representing, respectively, the day, month, and year minus 1900. Date is transmitted day first followed by the month and year bytes. This allows the representation of any date between 1 January 1900 and 31 December 2155.

## 5.3 Time

In the Protocol, time is contained within an unsigned 32-bit binary integer with the with the least significant bit of the time value representing 1/32 of a millisecond (i.e., 0.03125 milliseconds).

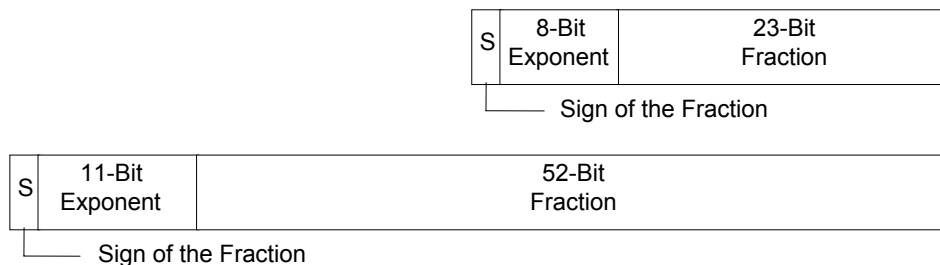
When time data types are used to represent time of day they indicate the number of 1/32 of milliseconds since midnight. Time can span in excess of 37 hours. When a longer span is required the field device should use time in conjunction with a Date (see Section 5.2).

## 5.4 Floating-Point Formats

IEEE-754 (IEC 559) compatible single and double precision floating-point formats are supported by the Protocol. In addition, All devices must support single precision floating-point numbers. Floating-point numbers consist of three parts: a sign bit, the exponent, and the fractional portion of the mantissa. The following summarizes the IEEE 754 floating-point formats. Detailed implementation information is beyond the scope of this specification.

- S** Sign of the mantissa (1 = negative)
- E** The biased exponent. Subtracting the bias results in a 2's complement integer.
- F** The fractional portion of the mantissa. Since the mantissa is between 1.0 and 2.0 the integer portion of the mantissa is always 1. The integer portion is not included in IEEE 754 single and double precision formats.

The sign and MSB of the exponent is transmitted first followed by the balance of the exponent and the MSB-LSB of the fraction. Furthermore, any floating-point number communicated via the Protocol must have either explicitly or implicitly an associated Engineering Unit Code (see *Common Tables Specification*). Figure 2 shows the floating-point formats supported.



**Figure 2. Single and Double Precision Floating-Point Formats**

There are four specially defined values: positive infinity ( $+\infty$ ), negative infinity ( $-\infty$ ), Not-a-Number (NaN), and denormalized numbers. When the Command Specification permits, a floating-point parameter not used by a device must be filled with a specific NaN: 0x7F, 0xA0, 0x00, 0x00.

Table 4. Summary of Floating-Point Number Properties

	<u>Single Precision</u>	<u>Double Precision</u>
Number of Bytes	4	8
Range	3.4E +/- 38 (7 digits)	1.7E +/- 308 (15 digits)
Bias of Exponent	127	1023
Conversion to Real Number	$(-1)^S * 2^{(E-127)} * 1.F$	$(-1)^S * 2^{(E-1023)} * 1.F$
Normalized	$0 < \text{Exponent} < 255$	$0 < \text{Exponent} < 2047$
Denormalized	Exponent = 0 Nonzero Fraction	Exponent = 0 Nonzero Fraction
Zero	Exponent = 0 Fraction = 0	Exponent = 0 Fraction = 0
Infinity	Exponent = 255 Fraction = 0	Exponent = 2047 Fraction = 0
NaNs	Exponent = 255 Nonzero Fraction	Exponent = 2047 Nonzero Fraction

5.4.1 Normalized Numbers

Normalized numbers (Figure 3) represent all real values between the minimum and maximum allowed for that floating-point format. These values can be used directly in calculations or for values to or from a HART field device.

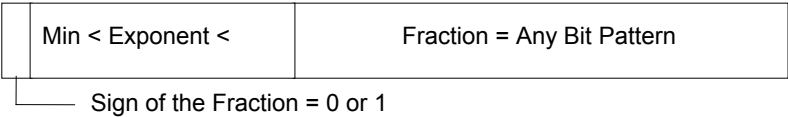
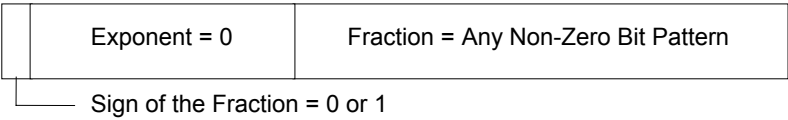


Figure 3. Normalized Numbers

5.4.2 Denormalized Numbers

The IEEE Standard allows for a smooth transition when an underflow occurs. In other words, zero can be gradually approached from nonzero by incrementally losing precision. A denormalized number always has a zero valued exponent; the integer portion of the mantissa equal to zero; and a nonzero fraction. Denormalized number may be positive or negative.

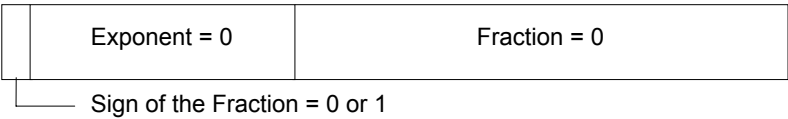
Note: Some HART compatible devices may not support denormalized numbers.



**Figure 4. Denormalized Numbers**

**5.4.3 Zeros**

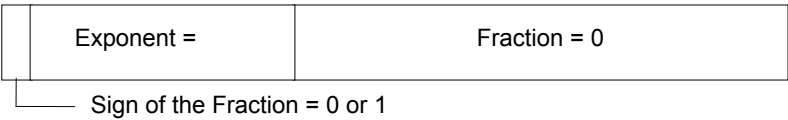
A zero always has: a zero valued exponent; the integer portion of the mantissa is zero; and the fraction is zero. Zeros may be positive or negative (i.e., +0.0 or -0.0).



**Figure 5. Zeros**

**5.4.4 Infinities**

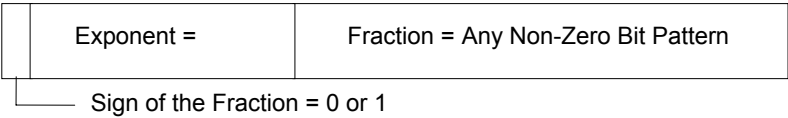
Infinities indicate a positive or negative overflow condition. An infinity always has: the exponent set to its maximum value; the integer portion of the mantissa is zero; and the fraction is zero.



**Figure 6. Infinities**

**5.4.5 Not-a-Numbers**

NaN indicates the number cannot be interpreted. The IEEE Standard allows for both signaling and non-signaling NaNs. A NaN always has the exponent set to its maximum value; the integer portion of the mantissa equal to zero, and a nonzero fraction. The most significant bit of the fraction is set to indicate a non-signaling NaN. Non-Signaling NaNs are generated as the result of a calculation that has no mathematical interpretation and signaling NaNs allow for implementation specific data definitions. Signaling NaNs must not be created as the result of a calculation. For example, a single, specific signaling NaN (0x7F, 0xA0, 0x00, 0x00) is allowed in some Command Specifications to indicate when the field device does not support certain data values.



**Figure 7. NaNs**

## 5.5 Unsigned Integer Format

This data type is a binary representation of a positive whole number. Unsigned integers must not have engineering units explicitly or implicitly associated their value. While the length of an individual data item is fixed, unsigned integers range from 1 to 24 bits or more in length. Unsigned integers shall be transmitted MSB first.

Note: The maximum size of an unsigned integer supported by Host Applications varies.  
Caution should be exercised when using long unsigned integers in field device designs

**Table 5. Unsigned Integer to Decimal Conversion**

15		Bit		8		7		Bit		0					
15		12		11		8		7		4		3		0	
4 <sup>th</sup> Hex Digit				3 <sup>rd</sup> Hex Digit				2 <sup>nd</sup> Hex Digit				1 <sup>st</sup> Hex Digit			
Hex		Decimal		Hex		Decimal		Hex		Decimal		Hex		Decimal	
0		0		0		0		0		0		0		0	
1		4,096		1		256		1		16		1		1	
2		8,192		2		512		2		32		2		2	
3		12,288		3		768		3		48		3		3	
4		16,384		4		1024		4		64		4		4	
5		20,480		5		1280		5		80		5		5	
6		24,576		6		1536		6		96		6		6	
7		28,672		7		1792		7		112		7		7	
8		32,768		8		2048		8		128		8		8	
9		36,864		9		2304		9		144		9		9	
A		40,960		A		2560		A		160		A		10	
B		45,056		B		2816		B		176		B		11	
C		49,152		C		3072		C		192		C		12	
D		53,248		D		3328		D		208		D		13	
E		57,344		E		3584		E		224		E		14	
F		61,440		F		3840		F		240		F		15	

Note: The decimal number is calculated by adding the values of each hex digit together.

## 5.6 Signed Integer Format

This data type is a 2's complement binary representation of a whole number. Signed integers must not have engineering units explicitly or implicitly associated their value. Since the most significant bit being set indicates the signed integer is negative, this bit must be extended when a signed integer is extended (e.g., when moving a 1 byte signed integer into a 2-byte signed integer for internal calculations). Signed integers must always be passed in multiples of eight bits. Signed integers shall be transmitted MSB first.

Note: The maximum size of an signed integer supported by Host Applications varies. Caution should be exercised when using long signed integers in field device designs.

**Table 6. Signed Integer to Decimal Conversion**

Positive Values								Negative Values							
Bit								Bit							
15	12	11	8	7	4	3	0	15	12	11	8	7	4	3	0
4 <sup>th</sup> Hex Digit	3 <sup>rd</sup> Hex Digit	2 <sup>nd</sup> Hex Digit	1 <sup>st</sup> Hex Digit	4 <sup>th</sup> Hex Digit	3 <sup>rd</sup> Hex Digit	2 <sup>nd</sup> Hex Digit	1 <sup>st</sup> Hex Digit	4 <sup>th</sup> Hex Digit	3 <sup>rd</sup> Hex Digit	2 <sup>nd</sup> Hex Digit	1 <sup>st</sup> Hex Digit	4 <sup>th</sup> Hex Digit	3 <sup>rd</sup> Hex Digit	2 <sup>nd</sup> Hex Digit	1 <sup>st</sup> Hex Digit
0	0	0	0	0	0	0	0	0		0	-3840	0	-128	0	-16
1	4096	1	256	1	16	1	1	1		1	-3584	1	-240	1	-15
2	8192	2	512	2	32	2	2	2		2	-3328	2	-224	2	-14
3	12288	3	768	3	48	3	3	3		3	-3072	3	-208	3	-13
4	16384	4	1024	4	64	4	4	4		4	-2816	4	-192	4	-12
5	20480	5	1280	5	80	5	5	5		5	-2560	5	-176	5	-11
6	24576	6	1536	6	96	6	6	6		6	-2304	6	-160	6	-10
7	28672	7	1792	7	112	7	7	7		7	-2048	7	-144	7	-9
8		8	2048	8	128	8	8	8	-28672	8	-1792	8	-112	8	-8
9		9	2304	9	144	9	9	9	-24576	9	-1536	9	-96	9	-7
A		A	2560	A	160	A	10	A	-20480	A	-1280	A	-80	A	-6
B		B	2816	B	176	B	11	B	-16384	B	-1024	B	-64	B	-5
C		C	3072	C	192	C	12	C	-12288	C	-768	C	-48	C	-4
D		D	3328	D	208	D	13	D	-8192	D	-512	D	-32	D	-3
E		E	3584	E	224	E	14	E	-4096	E	-256	E	-16	E	-2
F		F	3840	F	240	F	15	F	0	F	0	F	0	F	-1

Note: The decimal number is calculated by adding the values of each hex digit together. If the most significant bit is set then the number is negative and the right hand table should be used.

## 5.7 Lookup Table Formats

The Protocol uses lookup tables to associate specific definitions to numeric codes or to define the interpretation of an individual bit when it is set. When a field device uses tables, the numeric values have specific definitions that allow all hosts to consistently utilize that data. Tables are coded using one of the following:

- **Enumerations** communicated as an unsigned integer with each value having a specific, unique, unambiguous definition. The enumeration provides the index to an entry in a lookup table containing the corresponding definition. A single byte allows a data item to depict one of 256 possible terms
- **Bit Fields** communicated as part of an unsigned integer with each bit having a specific, unique, unambiguous meaning when that bit is set. Each bit in the unsigned integer is listed as separate entry in a lookup table. A single byte can depict up to 8 terms, one definition for each bit set.

The *Common Tables Specification* contains a collection of standard lookup tables (e.g., Engineering Units, Manufacturer Identification Code) used throughout the Protocol. These tables must be used whenever possible.

When Common Tables are referenced, the tables or subsets of the tables must be used exactly as specified. For tables not covered by the *Common Tables Specification*, refer to the associated device-specific tables. Device-specific tables must be included in the device-specific document. All tables must comply with the following revision requirements:

1. When a table is created, all possible values need not be defined. All undefined codes in the table are reserved. Reserved values must not be used, returned or communicated by any device.
2. A term or definition may be added to a table only by replacing a reserved code and performing a major revision to the table. This will result in an increase in the increment of the major revision level of the associated document.
3. Codes may be added to a table. Once defined, however, the term and its corresponding code must not be changed and may not be deleted.

### 5.7.1 Enumerations

Data items that take on a single meaning from a list or table are encoded as enumerations and communicated as an unsigned integer. When an enumerated table is created, the largest code is reserved. Single byte (i.e., 8-bit) enumeration-based lookup tables are defined using the following format:

**Table 7. Single Byte Enumerated Table Format**

Code	Definition
0	Specified definition 0
1	Specified definition 1
⋮	⋮
n	Specified definition n
n+1	Reserved
n+2	Reserved
⋮	⋮
249	Reserved
250	"Not Used"
251	"None"
252	"Unknown"
253	"Special"
254	"Expansion"
255	Reserved

The definitions for codes 250–255 in single byte enumeration-based look-up tables are mandatory if the enumerations are allowed (i.e. the codes are not reserved). Codes need not be allocated sequentially. Any intervening unused codes shall be "Reserved". When designated by the Command Specification, a table-based data item unused by a device is coded 250 ("Not Used").

### 5.7.2 Bit Fields

Communication of information encoded as single-bit data (such as status and diagnostic information) may be communicated using bit fields. Individual bits are packed together, and reserved bits are added as necessary to communicate a whole number multiple of eight bits. As a result, only a whole number of bytes is communicated.

A lookup table is used to define the meaning of each bit set in the bit field. In other words, each bit shall be either true (set to a 1) or false (set to a 0). The entry in the lookup table shall define the meaning of the bit when that bit is true (set to a 1). Single-byte (8-bit) tables for bit fields must use the following format:

**Table 8. Single-Byte Bit Field Table Format**

Bit Mask	Specified Definition
0x01	For Bit 0 (LSB)
0x02	For Bit 1
0x04	For Bit 2
0x08	For Bit 3
0x10	For Bit 4
⋮	⋮
0x80	For Bit 7 (MSB)

Codes need not be allocated sequentially. Any unused codes shall be "Reserved". Multi-byte bit fields are transmitted MSB first.

## 6. FIELD DEVICE REVISION RULES

Compatibility between devices is a primary objective of the Protocol. The Field Device Revision Rules in this section allow enhancement and modification of HART compatible field devices while ensuring compatibility is maintained. These rules ensure that a field device may be replaced with a new version of the same Device Type without disrupting system operation. Furthermore, hosts must provide the same functionality for a new Device Revision as available for the previous, supported Device Revision without requiring a software upgrade or change.

All devices must adhere to the following revision rules:

1. All devices must adhere to a valid revision of the *HART Field Communications Protocol Specification*.
2. Device support of Command 48 must adhere to the requirements in the *Common Practice Command Specification*.

Note: HART 5 Field Devices that do not adhere to Protocol requirements for Operating Mode 1, Operating Mode 2, Analog Channel Saturated, and Analog Channel Fixed must correct their implementation and change their [Expanded](#) Device Type number.

3. At no time may a command be deleted from any device. Once a command is supported by any Device Revision all subsequent revisions must support the command.
4. The meaning of any data item in a command supported by the field device must never be changed. Commands may not be modified in any way to produce results different than would be obtained in a previous Device Revision. For example, any supported command issued to a Device Revision 1 field device must have exactly the affect when issued to Device Revision 2 field device.
5. Data items may be added to the end of any command. Any addition will result in an increment of the major revision level of the associated document or specification (i.e. a major Device Revision).
6. Entries may not be deleted from enumerated tables, bit fields or Response Codes. Furthermore, the meaning of an enumeration, bit field or Response Code may not be changed. Any addition will result in an increment of the major revision level of the associated document or specification (i.e. a major Device Revision).
7. The code number for a Device Variable must not change. The addition of a new Device Variable shall result in an increment of the Device Revision and the major revision level of the manufacturer's device-specific document (i.e. a major Device Revision).
8. The Device Family of a Device Variable must remain fixed. The Device Family may be changed from 250, "Not Used" by adding the appropriate Device Family Commands; an increment of the Device Revision; and the major revision level of the manufacturer's device-specific document (i.e. a major Device Revision).

9. The classification of a Device Variable must remain fixed. The classification may be changed from 0, "Unassigned" by supporting the appropriate unit codes; an increment of the Device Revision; and the major revision level of the manufacturer's device-specific document (i.e. this change is a major Device Revision).
10. Field Devices may upgrade from HART 4 without changing the Device Type by following these rules:
- All requirements of the HART Field Communications Protocol Specification being targeted by the Device Revision must be adhered to.
  - Commands 4 and 5 must be deleted.
  - All commands, except Command 0, must be implemented only in the Long Frame Format. Command 0 must be implemented in both Short and Long Frame Format.
  - A separate Configuration Changed bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master). When received Command 38 must reset only the bit corresponding to the master issuing the command. As required in HART 5, these bits must be non-volatile.
  - A separate Cold Start bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master).
  - [Commands 38 and 48 must be supported as specified in the \*Common Practice Command Specification\*.](#)
11. Field Devices may upgrade from HART 5 without changing the Device Type by following these rules:
- All requirements of the *HART Field Communications Protocol Specification* being targeted by the Device Revision must be adhered to.
  - A separate Configuration Changed bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master). When received, Command 38 must reset only the bit corresponding to the master issuing the command. As required in HART 5, these bits must be non-volatile.
  - A separate Cold Start bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master).
  - [Commands 38 and 48 must be supported as specified in the \*Common Practice Command Specification\*.](#)
  - [If Burst Mode is supported then 2 additional Burst Messages must be added. Event Notification must be added.](#)
12. [Field Devices may upgrade from HART 6 without changing the Device Type by following these rules:](#)
- [All requirements of the \*HART Field Communications Protocol Specification\* being targeted by the Device Revision must be adhered to.](#)
  - [Commands 38 and 48 must be supported as specified in the \*Common Practice Command Specification\*.](#)

- [Time must be supported and the time stamp added to Command 9.](#)
- [If Burst Mode is supported then 2 additional Burst Messages must be added. Event Notification must be added.](#)

When device changes do not comply with these revision rules, a new [Expanded](#) Device Type number must be [acquired from the Foundation and](#) assigned for the device. To avoid customer confusion, a visible change to the product (such as a new product name or packaging) should accompany the assignment of the new Device Type number.

## 7. APPLICATION LAYER INTERFACE

The HART Protocol provides specifications to facilitate two-way communication between field devices and host applications. The Protocol's Application Layer is command based. In other words, commands from master or slave devices are the basis for HART communication and the command number, embedded in the communication, determines the content of the message.

The Command Number indicates the specification of a unique, unambiguous packet of Data with a fixed Byte Count. These command number specifications are classified in Section 7.1.

### 7.1 Command Number Partitions

This section allocates command numbers between Universal, Common Practice, Device Family and Device-Specific functions. The Protocol supports single byte command numbers (0–255) and two byte extended command numbers (256–65,535). Commands 0–255 use the Command field of the message to indicate the command number. For commands 256–65,535 the Command field contains 31 (0x1F) and the two byte extended command number is found in the Data field. The command set is divided into the following classes (see Table 9):

- **Universal Commands.** A collection of commands that must be supported by all HART compatible devices. All Universal Commands must be implemented exactly as specified (see Universal Command Specification).
- **Common Practice Commands.** A collection of commands applicable to a wide range of devices. Common Practice Commands shall be supported by Devices whenever possible (see Common Practice Command Specification). If a device uses a Common Practice Command, the command must be implemented exactly as specified.

Note: Two formerly optional Common Practice Commands are now mandatory: Command 38 and Command 48. All devices must implement Commands 38 and 48.

- **Non-Public.** A special set of commands (122 through 126) intended for factory-only use during the construction of a field device. These commands should not be used when servicing a device in the field.
- **Wireless Commands.** A collection of commands to support WirelessHART products. All products supporting WirelessHART must implement all Wireless Commands.
- **Device Family Commands.** Collections of commands that allow the setup and parameterization of field devices without requiring using device specific commands or special device-specific drivers (see *Device Family Command Specification*). Device Variables are classified into families based on the type of process connection they support (e.g., pressure, temperature, valve/ actuators, flow, etc.). The Device Variable is the target of the Device Family Commands.
- **Device-Specific Commands.** Commands defined by the manufacturer according to the need of the field device. These commands are controlled by the manufacturer of the field device (see the manufacturer's device-specific document).

Reserved command numbers shall not be used in any device.

**Table 9. HART Command Number Partitions**

Command No.	Type	Description
0–30, <a href="#">38, 48</a>	Universal	See <i>Universal Command Specification</i> .
31	Expansion Flag	Used to signal the presence of a 16-bit command number in the data field of the command. See <i>Network Management Specification</i> .
32–121 ( <a href="#">Except 38 and 48</a> )	Common Practice	See <i>Common Practice Command Specification</i> .
122–126	Non-Public	Intended for factory use only during the construction of the field device.
127	Reserved	
128–253	Device-Specific	See the manufacturer's device-specific document.
254– <a href="#">511</a>	Reserved	
<a href="#">512–767</a>	<a href="#">Additional Common Practice</a>	<a href="#">See Common Practice Command Specification.</a>
<a href="#">768–1023</a>	<a href="#">WirelessHART</a>	<a href="#">See Wireless Command Specification.</a>
1024–33,791	Device Family	See <i>Device Family Command Specification</i>
<a href="#">33,792–64,511</a>	<a href="#">Reserved</a>	
<a href="#">64,512–64,765</a>	<a href="#">Wireless Device-Specific</a>	<a href="#">These command numbers are reserved for use by wireless networking manufacturer specific commands. No commands shall be created that limit or degrade the operation of compliant WirelessHART products. Use of commands in this range must be disclosed and reviewed by HCF prior to product release.</a>
<a href="#">64,766–64,767</a>	<a href="#">Reserved</a>	
<a href="#">64,768–65,021</a>	<a href="#">Additional Device-Specific</a>	<a href="#">These device specific commands shall only be used when the command numbers in 128-253 are substantially (greater than 90%) consumed by the device. See the manufacturer's device-specific document.</a>
65,022–65,535	Reserved	

## 7.2 Command Requirements

The HART Application Layer is command based. To insure consistent implementations all HART commands must meet the requirements in this section. These requirements are applicable to both commands in the HART Protocol Specifications and device-specific commands developed by individual manufacturers of HART compatible devices.

### 7.2.1 Autonomous & Asynchronous Requirements

HART commands must be designed to be autonomous and allow stateless operation of the device's application layer. In other words, neither a slave nor a master can be required to remember prior communications in order to understand the current message. The response to be supplied by a slave device must be clearly and uniquely specified by the corresponding master request alone.

### 7.2.2 Command Operations

A HART command must perform one and only one of the following functions:

- **Read** data from a field device. READ commands contain no data in the Request Data Byte field other than that required by the field device to return the desired data items. In other words, the only data item that may be in the request is an index referencing an array entry in a field device (e.g., the device variable number). READ commands should have several related data items in the Response Data Byte field in order to minimize host upload times. A READ command must not change the operation of the field device in any way or change any data item stored in the field device.
- **Write** data to a field device. WRITE commands shall contain the same data items in the Request and Response Data Byte fields. Furthermore, the field device must return in the Response Data Byte sub-field the actual value of the data item as used by the field device in the same Engineering Units as sent with the data item. Any data item that can be written by a host must be contained in a READ command to allow the configuration of the field device to be determined or saved by the host. Index data items and unit codes may not be stored or changed in a field device as the result of a WRITE commands.

Unless otherwise stated in the Command Specification all data transmitted to a field device using a WRITE commands must be retained through a Device Reset, Self Test or removal of power from the field device.

- **Command** the device to perform some action. As a result, a COMMAND command may or may not have data items in the Request or Response Data Byte fields. COMMAND commands may affect the operation or configuration of the field device, or data items in the field device.

Unless otherwise stated in the Command Specification, all device operations or configurations affected by a COMMAND command shall be retained through a Device Reset, Self Test, or removal of power from the field device.

### 7.2.3 Indexed Commands

Commands may contain indices allowing access to arrays or tables of data stored in a field device. The index is represented as an unsigned integer. This allows single command access to an array of data. The number and the type of data items must be the same and must occur in the same sequential order in the Request and Response Data Byte field for all values assumed by the index. In other words, an indexed command must define a rigid packet of information and, for each index value, the structure of this packet must be identical.

There must be only one set of Command-Specific Response Codes for all values of the index.

### 7.2.4 Multi-Transaction Commands

Multi-transaction commands allow a sub command number to be placed in the Request and Response Data field to increase the number of device-specific commands. Multi-transaction commands should only be used when a field device exhausts the allowed set of device-specific commands. Furthermore, all transactions must perform the same READ, WRITE or COMMAND operation (see Section 7.2.2).

Unlike indexed commands, the number and the type of data items in the Request and Response Data Byte field can vary by transaction number. As a result, the command specification must include a separate Command-Specific Response Code specification for each transaction.

Both the Request and Response Data field must include the transaction number to meet the Protocol requirements for autonomous operation.

## 7.3 Command Status Bytes

This section defines the Command Status Bytes that provide host application feedback on slave device command execution. All slave response messages must return two Command Status bytes in the first two bytes of the Data field. The first byte is multiplexed and contains either the Communication Status (most significant bit is set) or the Response Code (most significant bit is reset). The second byte of a slave response message always contains Field Device Status.

- **Communication Status** is returned if a communication error is detected by the field device.
- If there are no communication errors then the **Response Code** gives the result of the executed command.
- The **Device Status** represents the current state of the slave.

The Response Data Bytes must not be returned if a communications or command error is reported in the Command Status Bytes.

### 7.3.1 Communication Status

This byte is multiplexed with the Response Code byte and indicates field device detection of a communication error. A communication error is always indicated when the most significant bit is set to one. The Communication Status is defined as bit field table (see Table 10). The Communication Status is only returned when a communication error is detected.

**Table 10. Communication Status**

Bit Mask	Definition
0x80	<b>1</b> - This bit must always be set to indicate a communication error
0x40	<b>Vertical Parity Error</b> – The parity of one or more of the bytes received by the device was not odd.
0x20	<b>Overrun Error</b> – At least one byte of data in the receive buffer of the UART was overwritten before it was read (i.e., the slave did not process incoming byte fast enough).
0x10	<b>Framing Error</b> – The Stop Bit of one or more bytes received by the device was not detected by the UART (i.e. a mark or 1 was not detected when a Stop Bit should have occurred)
0x08	<b>Longitudinal Parity Error</b> – The Longitudinal Parity calculated by the device did not match the Check Byte at the end of the message.
0x04	<b>Reserved</b> , set to zero
0x02	<b>Buffer Overflow</b> – The message was too long for the receive buffer of the device.
0x01	<b>Reserved</b> , set to zero

### 7.3.2 Response Code

When no communications errors are detected the first byte in the Data field contains the Response Code. The Response Codes contain a command completion report indicating the status of the command's execution by the field device. Response Codes provide a Notification, Warning or Error indication to the host (see Table 11)

**Table 11. Response Code Classification**

<b>Response Code Class</b>	<b>Definition</b>
<b>Notification</b>	Command executed properly. The Response Code equals zero (0) and the Response Data Bytes are returned.
<b>Warning</b>	Command executed with the deviation as described in response (e.g., a value was set to its nearest legal value). The Response Data Bytes are returned.
<b>Error</b>	Command execution was not properly completed and the Response Code indicates the reason (e.g., the device is in Write Protect mode). While the Extended Command number is included (if appropriate) in the slave response, the Response Data Bytes are NOT returned.

The most significant bit of the Response Code is always set to zero. As a result, the Response Code is encoded as a 7-bit enumeration (i.e., as an enumeration between 0 and 127). In addition to the above classifications, some enumeration must always use the same definition while others may have one of several possible definitions.

Single-definition Response Codes have the same meaning independent of the command that uses them. Multiple-definition Response Codes have several meanings. However, all Response Codes have a single meaning for a given command. The only legal Response Codes for a command are documented in the Command Specification found in the Protocol Specifications or, for device-specific commands, the manufacturer's device-specific documentation. Reserved Response Codes may not be used by any device.

Device-Specific Commands must use a single-definition Response Code wherever possible. Multiple-definition Response Codes may be recycled and used in device-specific commands. In other words, a manufacturer may use multiple-definition Response Codes as needed to return command completion information for their device-specific commands. For multiple-definition Response Codes used in this manner, the meaning of the Response Code must be defined in the manufacture's device-specific documentation.

For more information on Response Codes and their use, see the *Command Response Code Specification*.

### 7.3.3 Field Device Status

The Field Device Status is contained in the second data byte in a Slave-to-Master frame as a bit field table. The second data byte indicates the current operating status of the field device as a whole and is not associated with the completion of any command. Unlike the requirements of the HART Protocol prior to revision 6.0, this byte must be meaningful even if a communication error is reported in the first byte. Table 12 lists the meaning of bit masks for the Device Status Byte.

Note: Since this byte contains no unused bit the Extended Device Status Byte has been added to Identity Commands and commands providing cyclical process data.

**Table 12. Device Status**

Bit Mask	Definition
0x80	<b>Device Malfunction</b> – The device detected a serious error or failure that compromises device operation.
0x40	<b>Configuration Changed</b> – An operation was performed that changed the device's configuration.
0x20	<b>Cold Start</b> – A power failure or Device Reset has occurred.
0x10	<b>More Status Available</b> – More status information is available via Command 48, Read Additional Status Information.
0x08	<b>Loop Current Fixed</b> – The Loop Current is being held at a fixed value and is not responding to process variations.
0x04	<b>Loop Current Saturated</b> – The Loop Current has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further.
0x02	<b>Non-Primary Variable Out of Limits</b> – A Device Variable not mapped to the PV is beyond its operating limits.
0x01	<b>Primary Variable Out of Limits</b> – The PV is beyond its operating limit.

Note: Voltage Mode Field Devices must use "Loop Current Fixed" and "Loop Current Saturated" to indicate the signaling voltage is fixed or saturated (see Section 8.1.1).

A **Device Malfunction** must be indicated when an error that prevents proper operation of the field device is detected (for example, at least one of the Device Variables is incorrect). Hosts should consider this an alarm condition and refrain from using the device for process control applications. Devices supporting Command 48 must provide diagnostic information in the Command 48 Response Data Byte field. The Device Malfunction bit mask must remain reset until a device completes its power-on self tests. If possible, the device must answer any requested HART command even if a Device Malfunction is detected.

A field device must contain a **Configuration Changed** bit for both the primary and the secondary master. Both bits are set when any configuration item in a field device is modified. These bits are used along with the **Configuration Change Counter** (see Command 0) to allow host to monitor the configuration of the field device. The Configuration Change Counter is a 16 bit counter that must be incremented once for every command received that changes the devices configuration. The value of the Configuration Changed bits Configuration and Change Counter must be maintained even if power is removed from the device or a Device Reset is performed.

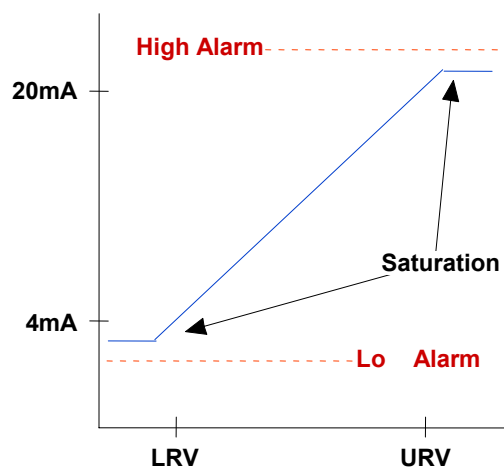
A **Cold Start** bit is maintained by the field device for each master as well. Both Cold Start bits are set when the field device is powered up and after a Device Reset. The first command from a primary or secondary master automatically resets the corresponding Cold Start bit.

When **More Status Available** is set, the measurements may still be correct and suitable for use by control systems. More Status Available merely indicates that Command 48 contains diagnostic information that is useful to the host. Since setting More Status Available will cause most hosts to issue Command 48, thus decreasing available communication bandwidth, field device designers must consider which diagnostics must set this bit.

When the Loop Current is directly (e.g., using Command 40) or indirectly (e.g., using Command 79) fixed and not responding to process changes, the **Loop Current Fixed** bit must be set. Setting this bit indicates to hosts that the Loop Current should not be used for analog signaling of the PV between the control system and the field device.

Note: A field device may be in multi-drop with the Loop Current still responsive to changing process conditions (see Universal Command 6)

The Loop Current varies from 4–20mA. The Upper and Lower Range Values set the 4 and 20mA points (see Commands 15 and 35). Most field devices allow the loop current to exceed these endpoints by some small amount. Once these electrical limits established by the field device are exceeded, the **Loop Current Saturated** bit must be set. For transmitters, their output becomes saturated. For actuators, their input is over or under range. Figure 8 shows the Loop Current operation as it reaches saturation. The alarm level set by the device (e.g., when the **Device Malfunction** bit is set) must be sufficiently different from the Loop Current Saturated levels to allow differentiation by devices monitoring the Loop Current.



**Figure 8 Loop Current Saturated versus Alarm Levels**

The Percent Range, PV and the corresponding Device Variable must return valid readings beyond the Upper and Lower Range Values. The reading must be accurate across the range defined by the Upper and Lower Transducer Limits (See Command 14). When the Device Variable (input or output) mapped to the PV reaches its Upper or Lower Transducer Limit, the **PV Out of Limits** bit must be set. For a device variable not mapped to the PV, the **Non-PV Out of Limits** bit must be used for this purpose instead. Limits may be electrical (e.g., in a transmitter) or mechanical (e.g., actuator at its travel limit as indicated by a limit switch).

## 8. DYNAMIC AND DEVICE VARIABLES

The HART Protocol is designed to support smart field device technology and the 4–20mA Loop Current (see HART commands 1–3). Commands 1 and 2 return the Primary Variable (PV), Loop Current and Percent Range. Command 3 adds to these the Secondary (SV), Tertiary (TV) and Quaternary (QV) Variables. Collectively these are called the Dynamic Variables:

**Dynamic Variables** consist of a Device Variable and an Analog Channel that when combined establish a communication link between the field device and the host application (e.g. the control system). All HART field devices may contain Primary, Secondary, Tertiary, and Quaternary Variables that are mapped to the first 4 analog channels in a field device. (see Figure 9). The first analog channel must support the Loop Current and HART communication. The SV, TV, and QV may or may not be supported and, furthermore, may not have an associated Analog Channel.

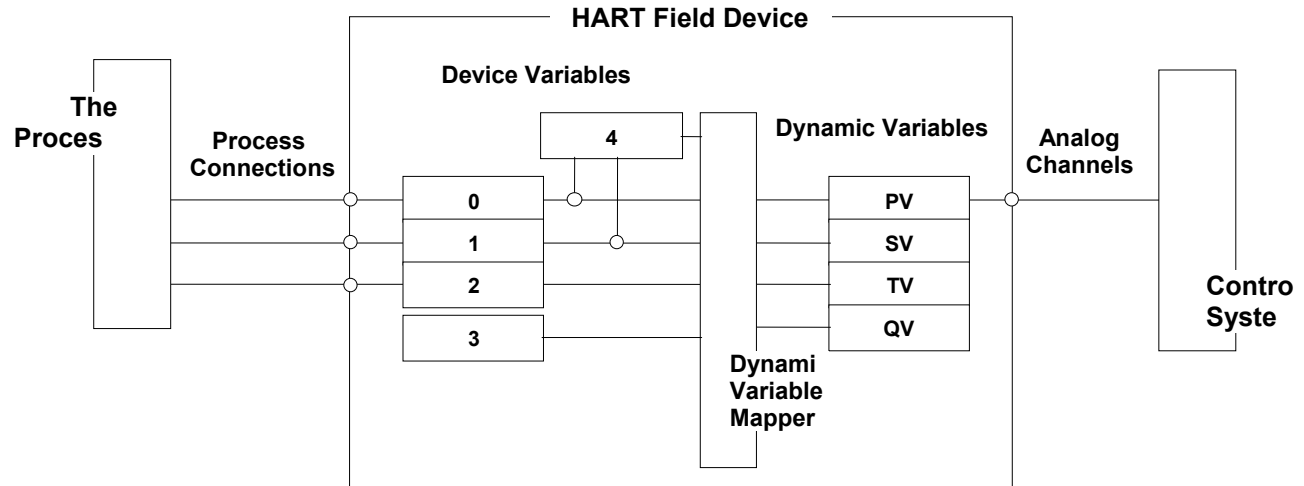
In addition, the Protocol supports Device Variables for use in more sophisticated smart field devices and multi-variable field devices. Furthermore, multi-variable field devices can configure which Device Variable to connect to the current loop.

**Device Variables**<sup>1</sup> are uniquely defined data items within a field device providing process-related information. A Device Variable's value changes as its connected process varies. A code number is assigned to each Device Variable and this assignment must never be changed for a given Device Type. The number of Device Variables is returned in Identity Commands. Device Variables should be numbered consecutively starting from zero (0).

Each Device Variable represents a direct or indirect connection between the process and the field device. An indirect connection infers the process state or level via calculations using direct process values from other Device Variables. In any case, a field device has a number of Device Variables (see Figure 9). Up to four of these Device Variables are mapped to the Dynamic Variables using the appropriate Device Variable number (see Command 50 and Command 51).

---

<sup>1</sup> HART Rev. 5 referred to Device Variables as "transmitter variables". HART Rev. 6 uses the term "device variables" in order to include actuator devices supporting the HART Protocol.



**Figure 9 Device Variables and Dynamic Variables**

Note: In addition to supporting access to the Device Variables all field devices must support access the Dynamic Variables and the Loop Current in all commands accessing Device Variable data or properties. In other words, the Device Variable Code Table found in the manufacturer's device-specific document always includes the codes found in Common Table 34, Device Variable Code Table.

The Dynamic Variables allow access to process-related data via HART Universal Commands and all HART compatible devices have Dynamic Variables. However, Device Variables are accessed via Common Practice commands. This means that even though all HART compatible devices have Device Variables, the device developer may choose to not expose the underlying Device Variables. As a result, a simple device may support only the Dynamic Variables and not implement the Common Practice Commands that access Device Variables.

## 8.1 Primary Variable (PV)

The term "Primary Variable" is widely used in Command Specifications. Commands referring to the Primary Variable include Commands 1, 2, 3, 9, 34-37, 40, 43-47, 49. In all cases, Primary Variable refers to properties of the Device Variable and Analog Channel associated with the Loop Current. In effect, the Primary Variable Commands allow convenient access to all configuration properties and process-related values connected with the Loop Current. In addition, this coupling of the Loop Current to the Primary Variable allows the Loop Current to serve as a communication channel that transmits the Primary Variable's value from the field device to the measurement or control system.

The Primary Variable's properties can be segregated into two domains (see Figure 10).

- **Transducer.** This domain characterizes the connection between the field device and the process. Properties found in this domain include: Upper and Lower Transducer Limit, Transducer Serial Number, Transducer Trim Points, and Damping Factor.
- **Analog Channel.** This domain is responsible for conversion between the digital value and engineering units found in the Device Variable domain and the milliamp signal found on the current loop. There are two sub-domains encapsulated by any Analog Channel:
  - **Range Conversion.** This domain is responsible for converting between the Percent Range and the Primary Variable's value. Properties found in this domain include: Upper and Lower Range Values and the Transfer Function.
  - The **DAQ** domain converts between the physical milliamp value that can be measured on the loop and the Loop Current value returned in Command 2 and Command 3. Properties found in this domain include: the Alarm Code, the DAQ's Zero and Gain, and Loop Current Damping.

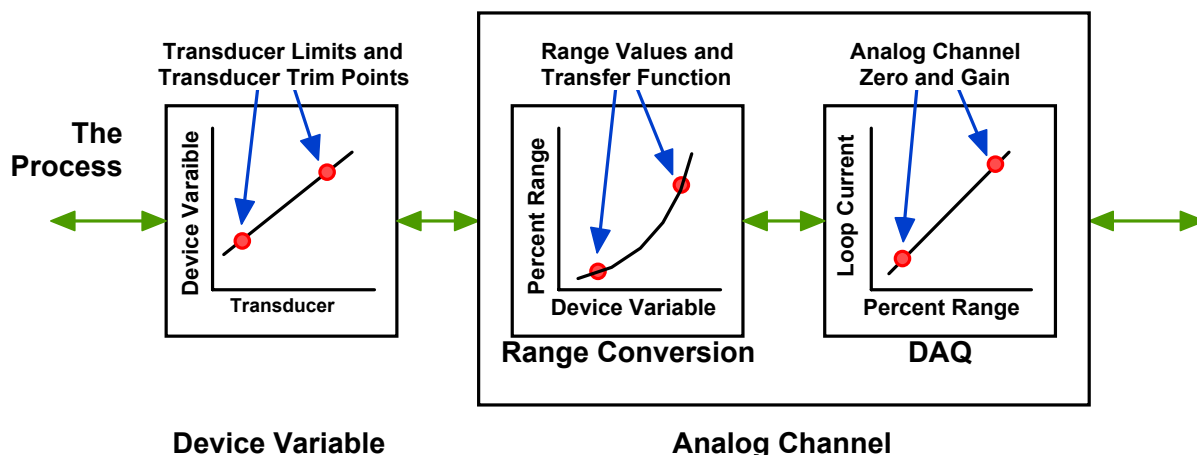


Figure 10. Domains Accessed Using PV

### 8.1.1 Voltage Mode Devices

A small number of HART compatible Field Devices support voltage signaling (e.g., 1-5 VDC) rather than Loop Current signaling (see Common Table 10, Physical Signaling Codes and the FSK Physical Layer Specification). Despite this the phrase "Loop Current" is always used in the Protocol Specifications to refer to the first analog channel in a Field Device which, in turn is communicating the PV.

Voltage Mode Field Devices returns data in engineering units of "Volts DC" everywhere the Loop Current value is used (e.g, Commands 2, 3, 40, 45, 46)

Host applications must recognize voltage mode Field Devices when they are encountered. These devices are identified by the Physical Signaling Code found in Identity Commands.

## 8.2 Device Variable Classification

Device Variables can be classified by the function performed (see Common Table 21). The Device Variable Classification property can be read with Command 54, Read Device Variable Information. Once defined, the classification of Device Variable must not change. The Device Variable Classification indicates the type of process connection the Device Variable characterizes and the Engineering Units that may be supported by the Device Variable (See the *Common Tables Specification*). Any Device Variable that does not support a classification must set its classification to zero ("0") indicating that the base unit code table must be used by the Host.

Since field devices are not required to support the Device Variable Common Practice Commands (e.g., Command 54), Universal Command 8, Read Dynamic Variable Classifications, can be used to determine the classification of the Dynamic Variables returned in Commands 1, 2, and 3.

## 8.3 Device Families

The *Device Family Command Specification* provides collections of Command Specifications based on the Device Variable Family (see Common Table 20). These collections of commands allow the setup and parameterization of field devices without requiring special device-specific commands. Command 54 indicates the Device Family supported by a Device Variable (any Device Variable that does not support a family returns 250, "Not Used"). The Device Variable Family code indicates:

- Applicable Device Family commands (see the *Device Family Specification*); and
- Meaning of the Device Variable Status byte (see Section 8.4).

Note: A Field Device must not set any of the Device Family specific status bits or the "More Device Variable Status Available" unless that Device Variable supports a Device Family.

The *Device Families Command Specification* include a separate sub-specification document for each Device Family defined. These Device Family Specifications are developed for specific types of process connection or for specific process related functions. Each Device Family:

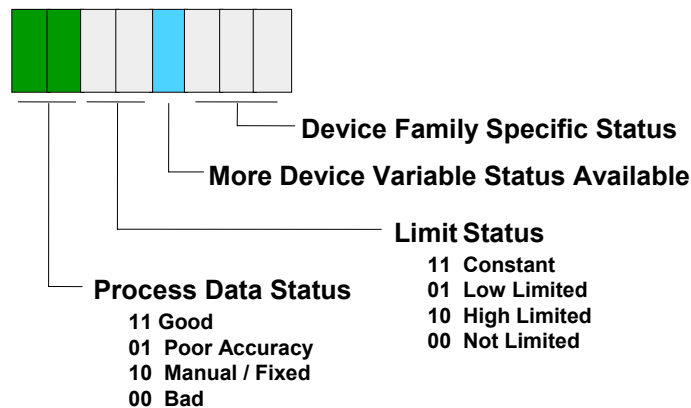
- Has a narrowly defined specified scope.
- Defines the properties for configuration or commissioning of the device variable.
- Classifies the device variable properties as optional or mandatory.
- Groups the properties into READ commands to optimize upload speeds.
- Defines WRITE commands for all properties in the collection.
- Specifies recommended Standard Operating Procedures (SOPs) for managing the device variable.

Device Variables that support the corresponding Device Family must support all mandatory commands and properties for that Device Family.

Since there are a limited number of HART commands and a large number of potential device families, Device Family commands all use Extended Command Numbers. Furthermore, Device Family command numbers are two bytes long with the Device Variable Classification in the most significant byte of the extended command number. 256 commands are allowed per device family.

## 8.4 Device Variable Status

All cyclical process data (i.e., Device Variables and Dynamic Variables) include a Device Variable Status byte (see Figure 11). The most significant two bits (i.e., bits 6 and 7) of every Device Variable Status byte return the overall status of the Device or Dynamic Variable value. The next two bits indicate whether the Device Variable value is limited (i.e., not responding to the process). These four bits provide useful status about the Device Variable's value. For example, if the Process Data Status is "Manual / Fixed" and the Limit Status is "Not Limited" then the value is being manually controlled. Limit status would not be "Constant" because the value could be changed at any time by the user.



**Figure 11. Device Variable Status Byte Format**

In addition, the content of the lower 4 bits depend on the Device Variable Family. Each Device Family can have its own Device Family-specific status defining the least significant bits. If set, Bit 3 indicates that additional Device Family-specific status is available via the appropriate Device Family Command (see the *Device Families Command Specification*).

Note: If the Field Device ever sets any of the least significant 4 bits then the Field Device must support Command 54 and the appropriate Device Family.

## 9. FIELD DEVICE IDENTIFICATION

All field devices must provide identification information to master devices upon request. The identifying data allows a master to, for example, address the field device or to ascertain the set of commands supported by the field device. The following data items are used for field device identification:

**Table 13. Field Device Identifying Data**

Property	Definition
<b>Manufacturer ID</b>	Indicates the company that produced the device. Manufacturer IDs are allocated by the HART Communication Foundation. Only the designated manufacturer may use this ID.
<b><a href="#">Expanded</a> Device Type</b>	Indicates the type of the device (i.e., the product name). The <a href="#">Expanded</a> Device Type indicates the set of commands and data items supported by the Field Device. See Section 9.2 and 9.3 for more information.
<b>Device ID</b>	A number unique to a particular field device. This number must be different for every device produced with a given Device Type.
<b>Device Revision</b>	A whole number indicating the revision level of command and data item set for this Device Type. Additions may be added to the commands and data items for a Device Type (see Sections 9.2 and 9.3).
<b>Software Revision</b>	An unsigned integer indicating the revision level of the firmware in the field device. An increment of this revision number must occur for every released version of the field device's firmware.
<b>Hardware Revision</b>	An unsigned integer indicating the major revision level of the hardware in the field device.
<b>Universal Command Revision</b>	A whole number indicating the HART major revision level supported by the field device.
<b>Private Label Distributor</b>	Indicates the company that sells or distributes the device. This number must be an entry in Common Table 8, Manufacturer Identification Codes. This code is set to the primary manufacturer of the device whenever the device is not private labeled (i.e., labeled and marketed by someone other than the manufacturer).
<b>Tag</b>	An 8-character label assigned by the end user based on the location and use of the field device. The Tag supports only the Packed ASCII character set (see Section 5.1.1).
<b>Long Tag</b>	A 32-character label assigned by the end user based on the location and use of the field device. The Long Tag supports ISO Latin-1 characters.

Groups of these data items are used for several purposes. Table 14 shows different identification activities and the data items used in each.

**Table 14. Application of Identifying Data**

<b>Data Items</b>	<b>Purpose</b>
Tag; Long Tag; Private Label Distributor; Device Type	User Identification of the Field Device
<a href="#">Expanded</a> Device Type <a href="#">and</a> Device ID (see the <i>Data Link Layer Specification</i> )	Link Layer Addressing of the Field Device. When combined, identify a unique field device. In other words, no two devices ever manufactured may have the same combination of <a href="#">Expanded Device Type and Device ID</a> .
<a href="#">Expanded</a> Device Type; Device Revision	Field Device Command and Data Item Set Identification
Device Revision; Software Revision; Hardware Revision; Universal Command Revision	Field Device Revision Information

## 9.1 User Identification of the Field Device

When providing the end user with field device identification information the host must display:

- The Private Label Distributor and Device Type; and
- The Tag, the Long Tag, or both.

The Private Label Distributor and Device Type allows the user to know which product is being communicated with. Since the Protocol allows for private labeling of devices, the Manufacturer ID should not be displayed. This allows one manufacturer to sell devices manufactured by another manufacturer without the name of the primary manufacturer appearing.

The Tag or Long Tag allows the end user to know where the field device is installed in the process.

## 9.2 Field Device Revisions

Identity Commands provide the following revision information about a field device:

- Device Revision
- Software Revision
- Hardware Revision
- Universal Command Revision

Each of these revision numbers returns an unsigned integer (i.e., a whole number) and provides a different function. The Universal Command revision indicates the major revision of the Universal Command Specification and, as a result, the major revision of the Protocol as well. This allows the host to determine the minimum command set supported by a field device and infers operation of the Data Link Layer, Response Codes, and other Protocol related features.

The Hardware and Software revisions track the configuration of the field device. The Hardware revision indicates the major revision level of the electronics of the field device. Minor, non-functional hardware changes are not indicated. However, an increment of the software revision is necessary for all changes to the field device's firmware. These two revision numbers allow identification of the field device for technical support operations.

The Device revision indicates the revision level of the device's Application Layer and indicates the commands and data items available via the Protocol. The major revision level of the manufacturer's device-specific document must match this number. An increment of this revision number is necessary whenever a new command or data item is added to the field device. In addition, the rules governing the use of Device Type and Device Revision in Section 9.3 must be adhered to.

### 9.3 Identifying the Field Device's Command Set

To identify the commands and data items supported by a field device, a host recognizes three data items returned by the Identity Commands:

- [Expanded](#) Device Type
- Device Revision

In effect the [Expanded](#) Device Type indicate a specific product and the Device Revision indicates the version of the command set for that product. Together these data items define the field device in relationship to the HART Protocol. Since a HART host is only concerned with this relationship, several devices that are packaged differently may use the same firmware.

A complete list of commands, Device-Specific Command Definitions, and Common-Practice Command Definitions must be documented in the manufacturers Device-Specific documentation.

## 10. HOST CONFORMANCE CLASSIFICATIONS

Host conformance classes identify host capabilities based on the level of host functionality. This functionality encompasses the level of data access and manipulation provided by the host and includes references to Common Practice and Device Family Commands. A host may choose to implement any function regardless of its class. However, conformance to a specific class may be claimed only when a host has implemented all of the functions in that class, particularly with regard to Common Practice and Device Family Command support for connected field devices.

Each conformance class includes the functions of all the lower classes; Class 1 hosts afford minimal host capabilities while Class 5 hosts provide the greatest functionality. Class 1 is the minimum classification that can be claimed by any host application. Host Applications must clearly and visibly indicate their capabilities using the classifications listed in Table 15.

**Table 15. Host Conformance Classes**

Class	Description
0	The host does not meet the minimum requirements of Conformance Class 1.
1	The host can utilize cyclical process data from any field devices.
2	The host can supply the user with basic identification and configuration data about any field device.
3	The host can perform basic configuration of any field device. Minimum level required to be classified as a "Generic Host".
4	The host provides basic commissioning and calibration support for any device.
5	The host is capable of accessing all field device data items and all device-specific commands for any field device.

Note: See the *Network Management Specification* for Master network management requirements. Depending on the system architecture, the network management requirements may be applicable to the Host Application.

## 10.1 Host Conformance Class 0

Class 0 identifies only those hosts that do not meet the minimum requirements of Conformance Class 1. In other words, any host that cannot access and utilize the cyclical process data supplied by all HART field devices fall into this conformance class.

## 10.2 Host Conformance Class 1

Class 1 host can perform the minimum set of recommended capabilities. A Class 1 host shall be able to access and utilize cyclical process data from any field device. Cyclical data is provided via HART Commands 1, 2, 3, and 9 (see Table 16). This data may be polled by the master or acquired from field devices in burst mode.

Note: The number of Device Variables is indicated in byte 13 of all Identity Commands and indicates the Device Variables that may be accessed using Command 9.

Compliance with Host Conformance Class 1 requires supporting commands in Table 16.

**Table 16. Host Conformance Class 1 Commands**

Cmd	Description	Cmd	Description
1	Read Primary Variable	3	Read All Dynamic Variables and Loop Current
2	Read Loop Current and Percent of Range	9	Read Device Variables with Status
93	Read Trend	33	Read Device Variables

## 10.3 Host Conformance Class 2

Class 2 hosts can supply the user with basic identification and configuration data about any field device. A Class 2 host can, for example:

- Indicate the Device Variable Classifications for all cyclical process data (i. e., for all Dynamic Variables and Device Variables);
- Display the Tag, Long Tag, Device Type, and Private Label Distributor;
- Show the Message, Descriptor and Date;
- Display the Loop Current configuration (i.e., Polling Address, Range Values and Loop Current Mode); and
- Decode the Command 48 Response Data Bytes sub-fields.

Compliance with Host Conformance Class 2 requires supporting all Class 1 requirements plus the commands in Table 17.

**Table 17. Host Conformance Class 2 Commands**

Cmd	Description	Cmd	Description
-----	-------------	-----	-------------

7	Read Loop Configuration	777	Read Wireless Device Capabilities
8	Read Dynamic Variable Families	778	Read Battery Life
11	Read Unique Identifier Associated with Tag	779	Report Device Health
12	Read Message	780	Report Neighbor Health List
13	Read Tag, Descriptor, Date	781	Read 2-Byte Nickname Address
14	Read Primary Variable Transducer Information	782	Read Session List
15	Read Device Information	783	Read Superframe List
16	Read Final Assembly Number	784	Read Link List
20	Read Long Tag	785	Read Graph List
48	Read Additional Transmitter Status	786	Read Neighbor Property Flag
50	Read Dynamic Variable Assignments	794	Read UTC Time Mapping
54	Read Device Variable Information	796	Read Timer Interval
76	Read Lock Device State	798	Read Radio Output Power
84	Read Sub-Device Identity Summary	799	Request Service
85	Read I/O Channel Statistics	800	Read Service
86	Read Sub-Device Statistics	802	Read Route
90	Read Real-Time Clock	803	Read Source-Route
91	Read Trend Configuration	804	Read CCA Mode
94	Read I/O System Client-Side Communication Statistics	806	Read Maintenance Frame Mode
95	Read Device Communications Statistics	808	Read Packet Time-to-Live
96	Read Synchronous Action	810	Read Join Priority
98	Read Command Action	812	Read Packet Receive Priority
101	Read Sub-device to Burst Message Map	814	Read Device List Entries
105	Read Burst Mode Configuration	817	Read Channel Blacklist
114	Read Caught Device Variable	832	Read Network Information
115	Read Event Notification Summary	833	Read Neighbor Information
512	Read Country Code	834	Read Network Topology Information
769	Read Join Status	835	Read Burst Message List
772	Read Join Shed Time	841	Read Network Access Mode
774	Read Network Id	64,512	Read Wireless Module Revision
776	Read Network Tag		

Note: Command 48 support in a Class 2 may be limited to the display of the hexadecimal values in the Response Data Byte sub-field for device-specific status. The Operating Mode 1, Operating Mode 2, Analog Channel Saturated, and Analog Channel Fixed data fields should be decoded.

## 10.4 Host Conformance Class 3 - Generic Host

This Conformance Class defines the minimum capabilities required for a host to be classified as a "Generic Host". A Generic Host can perform basic configuration of any field device even though the host does not have special drivers for the field device. Basic configuration includes:

- HART communications (e.g., the Tag, Long Tag, Response Preambles, Flush Delayed Response Buffers and Polling Address)
- Loop Current operation (e.g., Range Values, Damping Constants)
- Simple diagnostic procedures (Loop Current testing, Self Test and Device Rest)
- Units Code selection for cyclical process data

Compliance with Host Conformance Class 3 requires supporting all Class 2 requirements plus the commands in Table 18

**Table 18. Host Conformance Class 3 Commands**

Cmd	Description	Cmd	Description
6	Write Polling Address	103	Write Burst Period
17	Write Message	104	Write Burst Trigger
18	Write Tag, Descriptor, Date	106	Flush Delayed Response Buffers
19	Write Final Assembly Number	107	Write Burst Transmitter Variables
22	Write Long Tag	108	Write Burst Mode Command Number
34	Write Primary Variable Damping Value	109	Burst Mode Control
35	Write Primary Variable Range Values	113	Catch Device Variable
36	Set Primary Variable Upper Range Value	116	Write Event Notification Bit Mask
37	Set Primary Variable Lower Range Value	117	Write Event Notification Timing
38	Reset Configuration Changed Flag	118	Event Notification Control
40	Enter/Exit Fixed Primary Variable Current Mode	119	Acknowledge Event Notification
41	Perform Transmitter Self Test	513	Write Country Code
42	Perform Device Reset	768	Write Join Key
44	Write Primary Variable Units	770	Request Active Advertising
51	Write Dynamic Variable Assignments	771	Force Join Mode
53	Write Device Variable Units	773	Write Network Id
55	Write Device Variable Damping Value	775	Write Network Tag
59	Write Number of Response Preambles	795	Write Timer Interval
71	Lock Device	797	Write Radio Power Output
72	Squawk	805	Write CCA Mode
79	Write Device Variable	807	Write Maintenance Frame Mode

Cmd	Description	Cmd	Description
87	Write I/O System Master Mode	809	Write Packet Time-to-Live
88	Write I/O System Retry Count	815	Add Device List Table Entry
89	Set Real-Time Clock	816	Delete Device List Table Entry
92	Write Trend Configuration	818	Write Channel Blacklist
97	Configure Synchronous Action	836	Flush Cached Responses for a Device
99	Configure Command Action	840	Write Network Access Mode
102	Map Sub-device to Burst Message		

## 10.5 Host Conformance Class 4

Class 4 hosts provide basic commissioning and calibration support for any field device. Calibration is performed using the Loop Current Commands and the Transducer Trim Commands (see Table 19). Basic field device commissioning shall be supported using the Device Family Commands (see the *Device Family Command Specification*)

Compliance with Host Conformance Class 4 requires supporting all Class 3 requirements, the commands in Table 19 and the Device Family Commands. Since Device Families are added periodically to the Protocol, Class 4 hosts shall list the Device Families or the Protocol major and minor revision level supported.

**Table 19. Host Conformance Class 4 Commands**

Cmd	Description	Cmd	Description
45	Trim Loop Current Zero	81	Read Device Variable Trim Guidelines
46	Trim Loop Current Gain	82	Write Device Variable Trim Point
80	Read Device Variable Trim Points	83	Reset Device Variable Trim

## 10.6 Host Conformance Class 5 - Requirements for a "Universal Host"

This Host Conformance Class indicates the maximum HART support by a host. "Universal Hosts" must be capable supporting all capabilities of all HART-enabled field devices. To be classified as a "Universal Host", the host must support all Encoded DDs available from the HCF DD Library without requiring any additional files or other information.

## **ANNEX A. REVISION HISTORY**

### **A.1. Changes from Rev 8.1 to Rev 9.0\***

1. The Time data Type was added
2. The modifications indicated in the Addendum were added to support 16-bit Manufacturer ID Codes.
3. The former Subsection 7.2 "Data Field Format" and Section 10 "Network Management" were transferred to the *Network Management Specification*
4. The former Subsections 10.2 "Sub-devices and I/O Systems" and Subsection 10.5 "Burst Mode Operation" have been transferred to the *Common Practice Command Specification*.

\* October 2008 – document updated to reflect the new HCF logo and copyright information.

### **A.2. Changes from Rev 8.0 to Rev 8.1**

The only change to the document in this revision is adding an addendum for 16-bit Manufacturer ID Codes.

### **A.3. Changes from Rev 7.1 to Rev 8.0**

1. These new sections were added as part of the format revisions for all HART Protocol Specification documents: Scope, Reference, Definitions, Symbols/Abbreviations.
2. Replaced "transmitter" with "device" to demonstrate applicability of commands to many device types.
3. Section 5 Data Types was expanded to include ISO Latin-1 Strings. Specification of all data types clarified. As a result, many tables and figures were added. In addition, requirements for lookup tables were incorporated from Section 6.3 of the previous version.
4. Section 6 Field Device Revision Rules updated for HART 6. In addition, these requirement were elevated to a separate section (i.e. was Section 6.1 in previous version)
5. The Command Number Partitions, Command Requirements, and Command Status Bytes are now consolidated in the Section 7 Application Layer Interface. These were formally found in Sections 2.1, 3 and 6.2 in the previous revision. Section 7.2 Data Field is new and explains, for example, the use of extended command numbers.
6. Section 8, Dynamic and Device Variables, was clarified and figures added (this was Section 5 in previous version).
7. Section 9 Field Device Identification is clarified and expanded from Sections 6.1 and 6.6–6.9 of the previous version.

8. Network Management requirements (Section 10) was added to include information formerly spread across many documents. Some parts that existed previously were updated to reflect changes in HART Rev. 6. Identification of field devices, Delayed Response Mechanism, Sub-Devices are some of the requirements added or clarified.
9. Host Conformance Class moved to Section 11 and requirements clarified.

#### **A.4. Changes from Rev 7.0 to Rev 7.1**

The document was translated from a MultiMate document to Microsoft Word. As a result of this translation the document format was altered. No other modifications were made to the document.

#### **A.5. Changes from Rev 6.0 - Final to Rev 7.0 - Final**

1. This revision adds commands for devices with Multiple Analog Outputs and Analog Outputs other than Current.
2. This revision adds More Status Available to the Field Device Status Byte of the Response Codes.
3. Summarized Release Notes from Rev 5 to Rev 6.0 - Final

<u>Page</u>	<u>Line</u>	<u>Change</u>	<u>Text</u>
TP	4	Replace	"6.0" by "7.0"
TP	5	Replace	"14 February 1990" by "11 October 1990"
TP	6	Replace	"14 February 1990" by "11 October 1990"
TP	7	Replace	"14 February 1990" by "11 October 1990"
TP	8	Replace	"D8900069;" by "D9000035;"
2	19	Insert	"P. V."
2	31	Insert	"Primary Variable"
2	44	Insert	"60 * Read Analog Output and Percent of Range..."
2	49	Replace	"110" by "110*"
3	8	Insert	"Primary Variable"
3	9	Insert	"Primary Variable"
3	9	Delete	"(Push SPAN Button)"
3	10	Insert	"Primary Variable"
3	10	Delete	"(Push ZERO Button)"
3	13	Insert	"Primary Variable"
3	16	Insert	"Primary Variable"
3	18	Insert	"64 * Write Analog Output Additional Damping..."
3	36	Insert	"Primary Variable Current"
3	37	Insert	"Primary Variable Current"
3	45	Insert	"67 Trim Analog Output Zero 68 Trim Analog..."
7	5	Replace	"Command Specific" by "Command-Specific"
7	10	Delete	"they"
7	12	Replace	"#32 and #64" by "#32, Busy, or #64, Command..."
7	24	Insert	"Command-Specific"

<u>Page</u>	<u>Line</u>	<u>Change</u>	<u>Text</u>
7	34	Replace	"Reponse" by "Response"
9	24	Replace	"Reserved, set to zero." by "More Status..."
9	29	Replace	"Bit #3 Output..." by "Bit #3 Primary Variable. -."
9	34	Insert	"Primary Variable"
9	35	Replace	"current" by "analog outputs for the Primary..."
9	36	Replace	"cannot respond to..." by "no Longer represent..."
12	13	Replace	"the analog output." by "Analog Output #1."
12	14	Replace	"the analog output..." by "Analog Output #1..."
12	16	Replace	"the analog output." by "Analog Output #1."
12	21	Insert	"When more than one Analog Output exists in a..."
12	31	Replace	"Variables," by "Variables and their..."
12	33	Delete	"to select the variable for the analog output,"
14	27	Replace	"type code," by "Type Code,"
15	22	Insert	"EXAMPLE PRIMARY VARIABLE CURRENT"
15	24	Insert	"Primary Variable"
15	24	Replace	"4mA" by "4 ma"
15	25	Insert	"Primary Variable Current"
15	26	Insert	"Primary Variable"
15	26	Replace	"20mA" by "20 mA"
15	27	Insert	"Primary Variable Current"
15	28	Insert	"Primary Variable"
16	32	Replace	"Rosemount Devices" by "Rosemount's devices"
18	6	Delete	"the"
18	6	Insert	"#1"
18	12	Replace	"the Analog Output..." by "Analog Output #1...."
18	16	Delete	"the"
18	16	Insert	"#1"
18	17	Delete	"the"
18	17	Insert	"#1"
18	17	Replace	"the current" by "this Analog Output"
18	18	Replace	"4 milliamperes;" by "its minimum;"
18	18	Replace	"#4," by "#3, Primary Variable Analog"
18	19	Delete	"Current"
18	20	Delete	"the"
18	21	Insert	"#1"
18	22	Insert	"The operation of Analog Outputs other than #1..."
18	36	Delete	"the"
18	36	Insert	"#1"
18	38	Insert	"Primary Variable"
18	39	Insert	"Primary Variable Current"

<u>Page</u>	<u>Line</u>	<u>Change</u>	<u>Text</u>
18	39	Insert	"Primary Variable Current"
18	40	Replace	"#46" by "#46;"
18	42	Insert	"In addition, Command #66, Enter/Exit Fixed..."
18	47	Replace	"information ." by "information."
20	26	Delete	"the"
20	26	Insert	"#1"
20	27	Replace	"4 milliamperes." by "its minimum."
21	14	Delete	"the"
21	14	Insert	"#1"

## **A.6. Major Modifications from Rev 5 to Rev 6.0 - Final**

1. This revision incorporates Unique Identifier, Burst Mode Operation, Polling Address, and other changes pertaining to the Extended Frame Format.
2. A decimal point and integer has been added to the HART document revision numbering system.
3. Rearranged document and renamed several sections.
4. Changed most occurrences of "transmitter" to "field device".
5. Deleted all references to the Transmitter Type Code being returned in the Transmitter-Specific Commands.
6. Deleted Command #4, Read Common Static Data.
7. Deleted Command #5, Write Common Static Data.
8. Removed Expansion from each of the Transmitter Command Class Partitions.
9. Changed the titles of several commands.
10. Added Command #11, Read Unique Identifier Associated with Tag.
11. Added Command #12, Read Message.
12. Added Command #13, Read Tag, Descriptor, Date.
13. Added Command #14, Read Primary Sensor Information.
14. Added Command #15, Read Output Information.
15. Added Command #16, Read Final Assembly Number.
16. Added Command #17, Write Message.
17. Added Command #18, Write Tag, Descriptor, Date.
18. Added Command #19, Write Final Assembly Number.
19. Added Command #57, Read Unit Tag, Descriptor, Date.
20. Added Command #58, Write Unit Tag, Descriptor, Date.
21. Added Command #59, Write Number of Response Preambles.
22. Added Command #108, Write Burst Mode Command Number.
23. Added Command #109, Burst Mode Control.

24. Added Command #110, Read All Dynamic Variables.
25. Added Command #111, Transfer Service Control.
26. Added Command #112, Transfer Service.
27. Changed Communications Error Summary Bit #2 from Message Time-out Reserved.
28. Changed the Command-Specific Response Code bit assignments from #0 - #3 to #0 - #6 and expanded the number of codes to 127.
29. Moved Command Response Summary, Bit #4, Transmitter Fault to Command-Specific Response Code #16 and renamed it Access Restricted.
30. Changed Command Response Summary Bit #5 to Response Code #32.
31. Changed Command Response Summary Bit #6 to Response Code #64.
32. Changed Field Device Status Bit #4 from In Burst Mode to Reserved.
33. Added section on IEEE 754 Floating Point Format.
34. Added section on ASCII Data Format.
35. Added section on Compatibility.
36. Removed section on Model 268 Compatibility.
37. Added section on DAC Trim Sequence.
38. Added section on Unique Identifier.
39. Added section on Polling Address.
40. Added section on Burst Mode.

(Refer to document Revision 6, D8900069, for detailed information.)

### **A.7. Major Modifications from Rev 4 to Rev 5**

1. This revision incorporates Write Protect Mode and adds Transmitter Variable Commands.
2. Added sections on variable descriptions, Write Protect Mode, Private Label Distributor, and Transmitter Type Code Expansion.
3. Added Command #50, Read Dynamic Variable Assignments.
4. Added Command #51, Write Dynamic Variable Assignments.
5. Added Command #52, Set Transmitter Variable Zero.
6. Added Command #53, Write Transmitter Variable Units.
7. Added Command #54, Read Transmitter Variable Damping Value.
8. Added Command #55, Write Transmitter Variable Damping Value.
9. Added Command #56, Write Transmitter Variable Sensor Serial Number.

### **A.8. Major Modifications from Initial Rev 3 to Rev 4**

1. Added Command #49, Write Sensor Serial Number.  
(Refer to document Revision 3, D8700038, and Revision 4, D8900067, for detailed information.)