How To Understand Telemetry Formats – the “tn.b”

Please refer to the XXT TechNote “SeqStringsGVsVVars&tn.b” for complete details on this topic. This “How To” document is intended to complement the Tech Note by providing an overview of “tn.b” usage and to serve as a quick reference. The Tech Note is a “must read” for anyone who will be working with Sequence Definition Strings.

Data is acquired by the Tool downhole with the full resolution native to each parameter. When it’s time to telemeter that data to the surface, the telemetry format (the “tn.b”) for each parameter appearing in the Sequence Definition String specifies how the data will be represented during transmission, which may be different from how it’s stored in the Tool. For example, a value may be telemetered with less than full resolution or with reduced range.

The goal of the Telemetry Format is simply to get the data to the surface using as few bits as possible while still providing for the necessary range and resolution. Because different parameters have different characteristics, a variety of telemetry formats are available so that the most efficient format for a given parameter can be used.

The full format specification for a numeric object has 3 fields and is referred to as the “tn.b” (verbalized as “t” - “n” - dot - “b”), where…

⇒ The “t” attribute specifies to the format and/or scaling type and is case sensitive. There are several numeric scaling and format types…

⇒ The “n” attribute specifies to the total number of bits telemetered (excluding Parity and ECC bits), which typically defines the data range and resolution

⇒ The “.” delimits the “n” and “b” fields and is required when b is specified

⇒ The “b” attribute is interpreted based on the value of “t” and may be limited by “n”

---

**Important Notice**

*XXT incorporated provides the information and ideas in this document for consideration and potential use by its customers and makes no representation or warranty regarding the accuracy of the information herein or that the ideas or statements herein are free from copyright or patent infringements, and assumes no responsibility or liability for their use.*
Default “tn.b” Values

In general, if “t” is not specified it will default to a "t" value that makes the most sense, which is either the Zero-Offset “o” telemetry format for predefined parameters (the parameter may not be a Generic Variable/GV or Virtual Variable/VVar) with a predefined range (hard-coded /limits) or based on the format of the parameter as stored in memory. For example, floating-point parameters will default to the floating-point “e” telemetry format. Parameters stored as 2’s complement integer parameters will default to the 2’s complement integer “i” telemetry format.

If “n” and/or “b” are not specified, they will default based on the specified or default value of “t”. For example, floating-point parameters will default to the floating-point “e16.10” telemetry format. Parameters stored as 2’s complement integer parameters will default to the 2’s complement integer “i8.0” telemetry format.

CAUTION

**XXT Incorporated urges all end-users to fully specify the tn.b requirements for all parameters that are not predefined or do not have defined limits – specifically all Generic and Virtual Variables. DO NOT allow the TM components to assign default “tn.b” values.**

As always, XXT customers are encouraged to design and test all fieldable sequence definition strings in the shop prior to using them in the field to ensure that they function exactly as anticipated.

Some of the things to consider when specifying a telemetry format are:

a) Can the reading be a negative number, or will it always be positive?

b) Will the number be an Integer? Fixed point? Floating point?

c) Does the parameter already have predefined limits for its possible range of values?

d) Do a very large range of values (many decades) need to be accommodated?

e) Does a logarithmic profile suit the range of parameter values best?
The following telemetry formats are available and described in the following sections along with some examples:

1) o – Zero-Offset Scaling
2) u – Unsigned Integer Format
3) i – Two’s Complement Integer Format
4) m – Signed Magnitude Integer Format
5) e – Floating Point Format
6) r – Register Format
7) x – Log10 Scaling Format
8) l – Logical Objects Format

1. **Zero-Offset Scaling Format (The Common “o” Format)**

A lower-case “o” designates “Zero-Offset Scaling”. The use of this format requires that the parameter be predefined (the parameter may not be a Generic Variable/GV or Virtual Variable/VVar) and that it have defined upper and lower limits, as is the case with the most common predefined data objects, making this the most commonly used format. The defined range of the data object is scaled over n-bits using the defined low limit as the offset, causing a low-limit value to be telemetered as 0. For objects defined with an inclusive upper limit (such as Inc, DipA, MagF & Grav) an upper-limit value scales to \((2^n-1)\) and for objects defined with an exclusive/modulo upper limit (such as Azm, TAzm, gTFA & mTFA) the upper-limit value scales to \((2^n)\), which is transmitted as 0. The “o”-type format is also the default format for defined-limit objects, which allows the user to only specify the “n” field in most cases, which is the most common practice.

**Restrictions:** The following restrictions apply to the “on.b” specifications...

\[ 1 \leq n \leq 21 \]
\[ b = 0 \text{ (required if specified)} \]

**Default Format:** “o10.0” *(for predefined & limited variables when “t” and/or “n” are not specified)*

**Examples:**

\[ \Rightarrow \text{Azm:11} \quad (\text{LSB} = 360.0 / 2048) \]
\[ \Rightarrow \text{Inc:10} \quad (\text{LSB} = 180 / 1023) \]
2. **Unsigned Integer Format ("u" Format)**

A lower-case “"u" indicates an n-bit, unsigned magnitude integer format. Set “b" to 0 to indicate a whole integer. Set “b" to a signed, non-zero value to create a fixed-point integer with the weight of the least-significant bit telemetered being $2^b$.

**Restrictions:** The following restrictions apply to “un.b" specifications…

$$1 \leq n \leq 21$$
$$-128 \leq b \leq 127$$

**Default Format:** “u8.0” *(for values stored in memory as unsigned integers)*

**Default “b”:** 0 (effects a “whole integer”)

**Examples:**

- \(\text{Var2:u5}\)  
  (range: \(0 \leq \text{Var2} \leq 31\); LSB = 1)
- \(\text{Var1:u8.-3}\)  
  (range: \(0.000 \leq \text{Var1} \leq 31.875\); LSB = 0.125)
- \(\text{Var3:u8.3}\)  
  (range: \(0 \leq \text{Var3} \leq 2040\); LSB = 8)

3. **2’s-Complement Integer Format ("i" Format)**

A lower-case “"i" indicates an n-bit, 2’s complement format. Set “b" to 0 to indicate a whole integer. Set “b" to a signed non-zero value to create a fixed-point integer with the weight of the least-significant bit telemetered being $2^b$.

**Restrictions:** The following restrictions apply to “in.b" specifications…

$$1 \leq n \leq 21$$
$$-128 \leq b \leq 127$$

**Default Format:** “i8.0” *(for values stored in memory as 2’s complement integers)*

**Default “b”:** 0 (effects a “whole integer”)

**Examples:**

- \(\text{Var2:i5}\)  
  (range: \(-16 \leq \text{Var2} \leq 15\); LSB = 1)
- \(\text{Var1:i8.-3}\)  
  (range: \(-16.000 \leq \text{Var1} \leq 15.875\); LSB = 0.125)
- \(\text{Var3:i8.3}\)  
  (range: \(-1024 \leq \text{Var3} \leq 1016\); LSB = 8)
4. Signed Magnitude Integer Format ("m" Format)

A lower-case "m" indicates an n-bit, signed magnitude integer format. The "i" format is preferred over the "m" format. The "m" format is not commonly used.

Set "b" to 0 to indicate a whole integer. Set "b" to a signed non-zero value to create a fixed-point integer with the weight of the least-significant bit telemetered being \(2^b\).

**Restrictions:** The following restrictions apply to “mn.b” specifications...

\[
\begin{align*}
&\Rightarrow 1 \leq n \leq 21 \\
&\Rightarrow -128 \leq b \leq 127
\end{align*}
\]

**Default Format:** None ("m" is not a valid format for memory storage)

**Default “b”:** 0 (effects a “whole integer”)

**Examples:**

\[
\begin{align*}
&\Rightarrow \text{Var2:m5} \quad \text{(range: -15 \leq \text{Var2} \leq 15; LSB = 1)} \\
&\Rightarrow \text{Var1:m8.-3} \quad \text{(range: -15.875 \leq \text{Var1} \leq 15.875; LSB = 0.125)} \\
&\Rightarrow \text{Var3:m8.3} \quad \text{(range: -1016 \leq \text{Var3} \leq 1016; LSB = 8)}
\end{align*}
\]

**Using “i” vs “m” Formats**

The Signed-Magnitude Integer Format “m” is typically not used. Typically the Signed, 2’s-complement Integer Format “i” is used where signed whole integers and signed fixed-point integers are required.

The “m” “Signed Magnitude Integer” Format is a legacy format and XXT provides it for compatibility with QDT.

The “m” “Signed Magnitude Integer” Format has 1 sign bit and (n-1) magnitude bits and differs from the "i" format in that it has symmetrical signed limits: The range (capacity) of the "m11.-8" format is: \(-3.99609375 \leq \text{GV5} \leq 3.99609375\) as opposed to \(-4.00000000 \leq \text{GV5} \leq 3.99609375\) for “i11.-8”.

It is possible that a -0.000 (yes – minus zero) value can result due to half-rounding down of a relatively small negative value to zero. XXT generally recommends using “i” format over “m” format where applicable.
5. **Floating Point Format ("e" Format)**

The "e" Telemetry Format is the Floating Point format (based on IEEE 754) and is useful when a very large range of values must be accommodated. Please refer to the XXT TechNote "SeqStringsGVsVVars&tn.b" for the exact implementation of this format.

When using the "en.b" format:

a) "e" indicates the "Floating-Point" format type  

b) "n" is the total number of bits to be telemetered  

c) "b" is the number of "mantissa" bits  

d) There is always one bit used for "sign"  

e) The number of exponent bits will then be \((n - b - 1)\)

**Restrictions:** The following restrictions apply to "en.b" specifications…

\[
\begin{align*}
5 & \leq n \leq 21 \\
2 & \leq b \leq 18 \\
3 & \leq (n-b) \leq 9
\end{align*}
\]

**Default Format:** "e16.10"

**Default “b”:** For \(n \leq 18\), \(b = n / 2\)  
For \(n > 18\), \(b = n - 9\)

**Examples…**

\[
\begin{align*}
\Rightarrow & \textbf{Res:e12.6} \text{ (signed 6-bit mantissa resolution; 5-bit exponent range: } 2^{-15} \text{ to } 2^{16}) \\
\Rightarrow & \textbf{Var1:e16.7} \text{ (signed 7-bit mantissa resolution; 8-bit exponent range: } 2^{-127} \text{ to } 2^{128})
\end{align*}
\]

For any given total number of bits "n", there will be a trade-off between the maximum value which can be represented and the amount of precision in that value, depending on what value is chosen for "b".

The largest number (positive or negative) which can be represented is approximately:

\[
\text{Largest Possible Number is about: } 2^{(n-b-2)}
\]
An easy way to specify a floating point format would be to first decide how many total bits “n” you want to telemeter (total resolution) and then choose a value for “b” that will provide you with the range necessary for the maximum required parameter value.

For example, if you want to telemeter values up to 100,000,000 using 12 bits (“n”), then a “b” value of 5 would give you numbers up to about 4.3 billion, so you’d specify “e12.5”. A “b” value of 6 would only give you numbers up to 65,000 or so – not enough.

Keep in mind that a telemetry word cannot be more than 21-bits total, including the use of Parity (1 bit) or ECC (3 bits). So for example, if you’d be using ECC bits in your Sequence String for an en.b formatted parameter, then “n” could be at most 18 bits.

 PLEASE NOTE…

XXT customers using “e” format should upgrade to the latest xxMPTx/Q V01.51 or xxMPTx/G2/*** V02.10 firmware, which corrects an ‘e’ format encoding firmware error.

6. Register (“r” Format)

A lower-case “r” indicates an n-bit, register format. Set “b” to indicate the least-significant bit to be encoded and transmitted. For example, an “SReg:r4.2” format would transmit the 4 bits B2, B3, B4 and B5 of the “SReg” register. Register parameter bits assign “B0” to the least-significant bit. If “SReg” is not defined, a 4-byte/32-bit virtual variable of type “Register” is created.

Restrictions: The following restrictions apply to “rn.b” specifications...

⇒ \( 1 \leq n \leq 21 \)
⇒ \( b \geq 0 \) (b=0 is the default)
⇒ \( (n+b) \leq (8 \times \text{reg_size}) \) where “reg_size” is the register size in bytes

Default Format: \( r <8 \times \text{reg_size}> .0 \) (which will fail if the register is a VVar)

Examples:

⇒ Reg2:r5.0 Reg2 bits[4:0]
⇒ vvReg:r8.24 vvReg bits[31:24]
7. **Log10 Scaling (“x” Format)**

Please refer to the XXT Tech Note: XXT TechNote “SeqStringsGVsVVars&tn.b” for information on the “x” format.

8. **Logical Objects (“l” Format)**

Please refer to the XXT Tech Note: XXT TechNote “SeqStringsGVsVVars&tn.b” for information on the “l” format.

---

**Important Notice**

*XXT incorporated neither warrants nor supports the use of its xxMWD™ telemetry components and other system components when configured with or used with other third-party PC applications. Only the use of xxMWDconfig/PC™ and other XXT xxMWD/PC™ applications are recommended and supported.*

Likewise, *XXT incorporated neither warrants nor supports the use of its xxMWDconfig/PC™ application to configure components that claim to be QDT-compatible or QDT-telemetry-compatible other than QDT V01.6x and, within certain limitations, QDT V02.0x-firmware-based telemetry components.*

As always, *XXT customers are encouraged to test all fieldable hardware and software configurations, including telemetry sequence definition strings, in the shop prior to using them in the field to ensure that they function exactly as required and anticipated.*

*XXT incorporated reserves the right to make changes to any of its PC software or embedded firmware applications without notice in order to improve product design and/or performance, or for whatever other reason.*

*XXT incorporated makes no representation or warranty that its PC software or embedded firmware applications are free from copyright or patent infringements, and assumes no responsibility or liability for their use.*