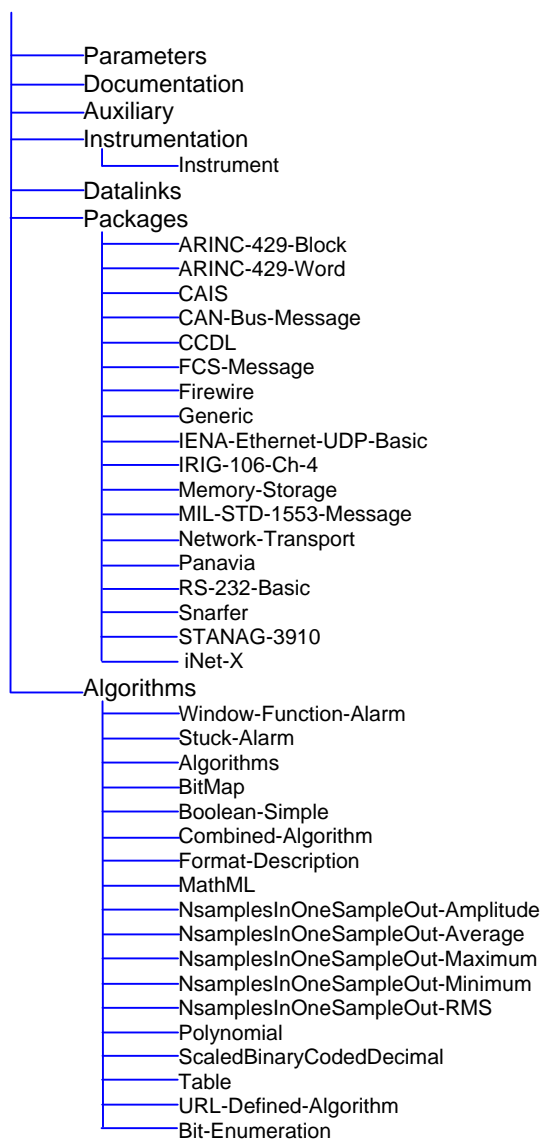
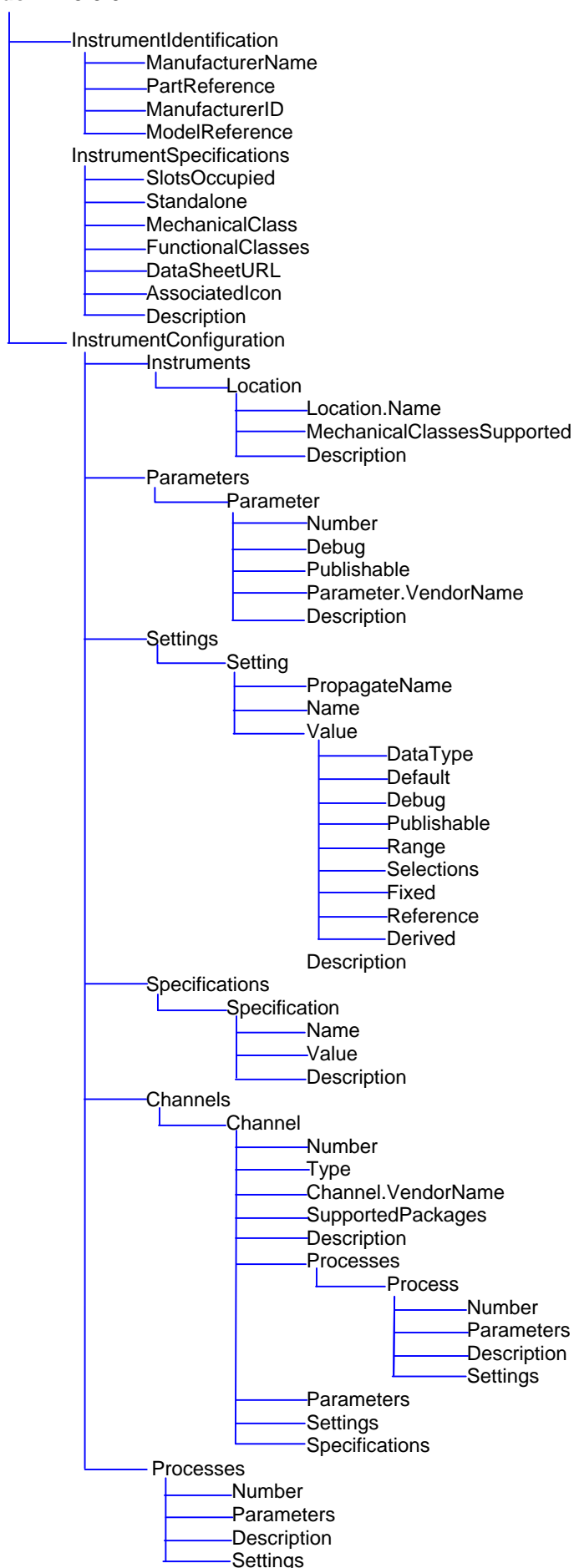


XidML 3.0.0



XdefML 3.0.0



Coding conventions followed in the development of XidML 3.0

Since XidML 2.2 and later, the ordering of elements is not important.

Use elements to store data and only use attributes to locate or describe data (e.g. Names, Indexes).

Elements (e.g. Syncword) that are the same for the transmitter and the receiver but not for all packages on the Link are properties of the package.

Elements (e.g. Misses To Loss) that are properties of the receiver only are defined as settings for the receiver instrument.

Wherever a key component (Instrument, Link, Parameter, Package, or Algorithm) maybe referenced (keeping the file flat) in the scope of some other key component, it may also be instantiated (using a tree structure).

Naming conventions followed in the development of XidML 3.0

Keep names short (<20 characters), with no white space, or any of the following five characters "/><. Note that specific vendors may not support the full range of characters that are allowed in XidML files. Document authors should check with individual vendors to see which character sets their software supports.

The prefix "My" is used in examples to clearly indicate a name that can be changed by the user (e.g. MyP1).

The prefix "Data" is used to denote bits and words containing parameters (e.g. DataBitsPerWord or DataWordsPerFrame).

Use clear names rather than abbreviations (e.g. RangeMaximum rather than Max).

Use the suffix "Index" to indicate a numbering scheme starting at 0, and use the suffix "number" if it starts with 1 and there is a risk of confusion (e.g. DataWordIndex or MinorFrameNumber).

Reserve the suffix "Tag" for parameters associated with a package but not actual traffic (e.g. TimeTag or StatusTag).

Be consistent with the names of elements. For example, don't use "MicroTime" in one schema and "TimeMicro" in another.

To encourage reuse of schemas, keep element names flexible (e.g. ExcitationAmplitude allows it to be used for excitation voltage or current, ac or dc).

XidML 3.0 dictionaries and validation

Since the introduction of one generic instrument schema and a simplified Link (formally DataLink) schema in XidML 3.0, the schema has become a lot more flexible. That is, all instrument settings are name/value pairs. In order to maintain consistency in how vendors name their settings, XidML.org has created and published (on www.xidml.org) a dictionary of recommended setting names and defaults, and urges users to use these setting names wherever possible. If you find that settings you regularly use in your instrument definitions are missing from this dictionary, please submit them to us and we will add them to the dictionary.

As a consequence of this extra flexibility, some of the validation power, with respect to instrumentation setup, has been reduced. To rectify this, XidML.org has introduced a validation schema called XdefML. XdefML allows vendors to specify vendor-specific constraints for their XidML instrument definitions. The constraint types provided by XdefML can be seen on the left. See www.xidml.org for more information on XdefML.

Hints, notes, and answers to some frequently asked questions

The same 14-bit offset binary parameter can be transmitted as 12 bits and stored as 16 bits if everyone agrees to truncate/pad the LSBs, however, with BitStream (e.g. embedded PCM, CVSD, MPEG-4) all bits must be used.

Use the BaseType Ratio for $\mu\epsilon$, V/V, %, ppm and scale accordingly.

It is recommend that XidML files are encoded as UTF-8.

Got any questions, comments, suggestions? Please feedback to www.xidml.org

Base Units

XidML 3.0 has chosen the 63 SI units listed below as its base units, as well as others such as Unitless and Bytes.

XidML units

BitStream
BitVector
Boolean
Bytes
Count
Ratio
Unitless

SI base units

Meter (m)
Kilogram (kg)
Second (s)
Ampere (A)
Kelvin (K)
Mole (mol)
Candela (cd)

SI derived units

SquareMeter (m²)
CubicMeter (m³)
MeterPerSecond (m/s)
MeterPerSecondSquared (m/s²)
ReciprocalMeter (m⁻¹)
KilogramPerCubicMeter (kg/m³)
CubicMeterPerKilogram (m³/kg)
AmperePerSquareMeter (A/m²)
AmperePerMeter (A/m)
MolePerCubicMeter (mol/m³)
CandelaPerSquareMeter (cd/m²)

SI special units

Radian (rad)
Steradian (sr)
Hertz (Hz)
Newton (N)
Pascal (Pa)
Joule (J)
Watt (W)
Coulomb (C)
Volt (V)
Farad (F)
ohm (Ω)
Siemens (S)
Weber (Wb)
Tesla (T)
Henry (H)
Lumen (lm)
Lux illuminance (lx)
Becquerel (Bq)
Celsius ($^{\circ}$ C)
Gray (Gy)
Sievert (Sv)

SI special derived units

PascalSecond (Pa-s)
NewtonMeter (N-m)
NewtonPerMeter (N/m)
RadianPerSecond (rad/s)
RadianPerSecondSquared (rad/s²)
WattPerSquareMeter (W/m²)
JoulePerKelvin (J/K)
JouleperKilogramKelvin (J/(kg-K))
JoulePerKilogram (J/kg)
WattPerMeterKelvin (W/(m-K))
JoulePerCubicMeter (J/m³)
VoltPerMeter (V/m)
CoulombPerCubicMeter (C/m³)
CoulombPerSquareMeter (C/m²)
FaradPerMeter (F/m)
HenryPerMeter (H/m)
JoulePerMole (J/mol)
JoulePerMoleKelvin (J/(mol-K))
CoulombPerKilogram (C/kg)
GrayPerSecond (Gy/s)
WattPerSteradian (W/sr)
WattPerSquareMeterSteradian (W/(m²-sr))
KatalPerCubicMeter (kat/m³)

Data Formats

XidML 3.0 has chosen the nine data formats listed below for parameter types.

OffsetBinary
BinaryCodedDecimal
IEEE-754-Float-Single

SignedMagnitude
BitVector
IEEE-754-Float-Double

TwosComplement
BitStream
UserFormat



www.xidml.org