Energistics Unit of Measure Usage Guide

Version 1.0

Energistics Unit of Measure Standard	A set of resources that defines a standard unit of measure (UOM) dictionary to promote consistent usage, data exchange, and unit conversion. The set includes the base Energistics Unit of Measure Dictionary and related documentation for creating, implementing, and maintaining a UOM dictionary that is patterned after the Energistics dictionary.
Version	**1.0**
Abstract	This document outlines best practices for creating, implementing and maintaining a UOM dictionary that is patterned after the Energistics Unit of Measure Dictionary.
Prepared by	Energistics
Date published	June 2014
Document type	Usage Guide
Keywords:	standards, energy, data, information, process, units of measure

energistics®
The Energy Standards Resource Centre

Document Information	
DOCUMENT VERSION	1.0
DATE	June 2014
Language	US English

Acknowledgements

Energistics wishes to thank its member community, the Professional Petroleum Data Management (PPDM™) Association, and the Society of Exploration Geophysicists (SEG) for their contributions to this standard.

Amendment History			
Version	**Date**	**Comment**	**By**
1.0	June 2014	First publication of the standard and document.	Energistics

Table of Contents

1 Introduction

Accurate use, exchange, and conversion of units of measure (UOM) in upstream oil and gas software are crucial. Errors in units of measure can cause serious problems for the accuracy and integrity of earth and reservoir models and the decisions that are based on those models.

The *Energistics Unit of Measure Standard* (the *UOM Standard*) represents the collaborative work of Energistics, its member community, the Professional Petroleum Data Management (PPDM™) Association, and the Society of Exploration Geophysicists (SEG) to update previous standards and best practices for use of units of measure used in the upstream oil and gas industry. (For a list of previous standards and specifications included, see Section 1.3 (page 6).)

The goal of the *UOM Standard* is to provide the necessary components and guidelines to improve accuracy and consistency of implementation, usage, exchange, and conversion of units of measure related to all upstream activities.

1.1 Audience, Purpose and Scope

This document is intended for IT professionals (e.g., software developers, information architects) who want to implement and maintain in their software a units of measure dictionary patterned after the *Energistics Unit of Measure Dictionary*, the key component of the *UOM Standard*. Software that implements Energistics data-exchange standards (e.g., WITSML, PRODML or RESQML) MUST implement the *UOM Standard*.

Implementation of the *UOM Standard* also includes:

- Conformance to the *Energistics Unit Symbol Grammar Specification.*
- Optional, but strongly recommended use of Energistics quantity classes and unit dimensions, vital information that adds usage context to what would otherwise be a bunch of symbols.

1.2 Resource Set

All resources that comprise the *Energistics Unit of Measure Standard* can be downloaded from the Energistics website at http://www.energistics.org/asset-data-management/unit-of-measure-standard.

This resource set includes:

Resource	**Description**
Energistics Unit of Measure Dictionary	XML file that is the normative dictionary that developers will implement. Also, an "information-only" spreadsheet version of the normative XML file is provided as an easier-to-read resource for people. NOTE: If any discrepancies exist between the XML and the spreadsheet, the XML is the normative source.
Energistics Unit Symbol Grammar Specification	Defines the Energistics unit symbol grammar, which uses an administered dictionary and algebraic operations to specify, combine, and constrain the unit components from the dictionary to produce unit symbols that represent desired units of measure. Conforms to the intent of RP66 V2.0.
Energistics Unit of Measure Usage Guide (this document)	Intended for IT professionals, provides usage requirements and guidelines for implementing and maintaining a units of measure dictionary patterned after the *Energistics Unit of Measure Dictionary*.
Readme.txt	A text file that explains contents of other supporting files and information contained in the *UOM Standard* download.

Resource	Description
Integer Codes	In association with this standard, a set of integer codes has been defined for use in special cases where only binary values can be used. The Society of Exploration Geophysicists uses these codes as part of its new SEGD Rev3.0 format.
Mapping Documents	To aid users and adopters, mappings for POSC v2.2, EPSG v8.1, Open Spirit Unit Dictionary v3.0 and RP66 v1 and v2 are included with this standard.

1.2.1 Additional Resources

It is recommended that anyone who is maintaining a dictionary also consult:

- The *Guide for the Use of the International System of Units (SI)* document (NIST Special Publication 811 – 2008 Edition), which is freely available online in PDF format.
- Recommended Practices for Exploration and Production Data Digital Interchange (RP66 v2) specification, which is explained Section 1.3 below.

1.3 Sources for UOM Standard

The current *UOM Standard* draws from these UOM sources as described below, which Energistics has developed over the years.

- **Epicentre Data Model** contains standard instance data for unit symbols and for unit conversions (based on the Society of Petroleum Engineer's (SPE) The SI Metric System of Units and SPE METRIC STANDARD, 1984). The Epicentre information is the source for the Unit of Measure Dictionary (currently V2.2) that is used by most current Energistics XML specifications (e.g., WITSML). (http://www.energistics.org/energistics-standards-directory/epicentre-archive)
- **Recommended Practices for Exploration and Production Data Digital Interchange (RP66 v2) specification**, which contains a unit model. The RP66 unit model defines an algorithm for creating a unit symbol, as opposed to the explicit list of unit symbols defined by Epicentre. (http://w3.energistics.org/RP66/V2/Toc/main.html)
- **Quantity Class specification**, which assigns unit symbols to a high-level quantity term. These classes are used, for example, by WITSML to define data types that constrain the allowed unit symbols for a quantity. (http://w3.energistics.org/uom/poscUnitsClasses22.xml)

2 Key Concepts

This chapter explains key concepts used in the *UOM Standard*. For more information see the *Energistics Unit Symbol Grammar Specification.*

2.1 Unit Dimension

The unit dimension represents a dimensional analysis of a unit. For example, a meter is of dimension *L*, which represents length. A foot is also in the *L* dimension. Each unit dimension specifies both:

- a base for conversion (the dimensional base).
- a canonical unit, which represents a reduction of the base for conversion to the SI base units.

There is no requirement for the dimensional base or the canonical unit to exist in the set of allowed units, but, if they exist, then they must conform to the grammar.

SI treats radian as a dimensionless ratio of length and omits it from dimensional analysis. However, in this dictionary, the concept of an angle is more important than how its base is measured. Therefore, the angle is incorporated as an underlying dimensional concept. For a similar reason, the solid angle is included in the analysis.

The following nomenclature is used when forming a dimension:

Symbol	Definition/Meaning	SI Unit (if applicable)
A	angle	radian
D	temperature difference	kelvin
I	electrical current	ampere
J	luminous intensity	candela
K	thermodynamic temperature	kelvin
L	length	meter
M	mass	kilogram
N	amount of substance	mole
S	solid angle	steradian
T	time	second
1	the number one	*NA*
2	squared (e.g., *M2*)	*NA*
3	cubed	*NA*
4	4^{th} power ($5=5^{th}$, etc.)	*NA*
/	division	*NA*
none	Special notation indicating unit for which a normal dimensionless analysis is possible.	*NA*

The following constraints apply:

- The values may be broken into numerator and denominator separated by a slash "/", but multiple slashes must not be used and a slash must not terminate the string. For example, *length per time* is indicated by *L/T* and *(mass per time) per area* uses *M/L2T* not *M/T/L2*.

- A number only follows a single dimensional character and represents the power of that component. For example, the concept of "area" is represented by *L2*, which indicates length times length.
- The number *1* only exists by itself or as the whole numerator (e.g., *1* or *1/T*). The number *1* must not be used as a power or a denominator because it is implied (e.g., *M* implies *M1/1*).
- The items within a numerator or denominator must be listed in alphabetical order (e.g., *LM* not *ML*). Adjacent components represent multiplication (e.g., *LM* represents length times mass, while *L2M* represents length squared times mass).
- Each letter must only exist once in the result. For example, the concept of "volume per area" is represented by the dimension *L*, where the underlying components cancel out.
- The letter *K* must not be combined with any other characters. See Section 3.2 (page 11).
- The notation *none* must not be combined with any other characters. Thus, a derived symbol that contains a component of dimension *none* must itself have the dimension of *none*. For example, the *bel* represents a logarithm of a ratio with a dimension of *none*, so the concept of *bel per meter* also has a dimension of *none* instead of *none/L*.
- Similar to the dimension, the components of the canonical unit are sorted alphabetically within the numerator or denominator.

2.2 Quantity Class

A quantity class represents a set of units with the same dimension and same underlying measurement concept. For example, length is a quantity class. Each quantity class specifies:

- a base for conversion and, optionally, an alternative base. Each base must be unique within the set of classes. It is the base that provides a distinction between classes that might otherwise seem similar.
- a dimension.
- a set of member units.

The quantity class can be used to constrain items in a data model because the class defines all of the units that are allowed to be used with something that represents a specialization of that class.

For example, tubular goods like drill pipe are described by their mass per length, typically pounds per foot. We would like to have a list of expected units for this kind of thing. So the standard has a quantity class called massPerLengthQuantity, which has a property called massPerLengthUom, which is limited to a certain list of values, including “lbm/ft”.

In XML, a type can be created to represent a particular class with a unit attribute that constrains the values to the list of symbols that are members of that class, as in the example below.

In a relational database, triggers can be used to constrain the units that are populated based on the class that is assigned to a column. In programming languages, a proxy generator would turn this into an enumeration. Other implementations are possible, but XML is the official definition of the standard.

```
<xsd:element name="weight" type="tns:massPerLengthQuantity"
minOccurs="0" maxOccurs="1"/>
<xsd:complexType name="massPerLengthQuantity">
   <xsd:simpleContent>
      <xsd:extension base="xsd:double">
         <xsd:attribute name="uom" type="tns:MassPerLengthUom"
use="required"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="MassPerLengthUom">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Mg/in"/>
      <xsd:enumeration value="kg.m/cm2"/>
      <xsd:enumeration value="kg/m"/>
      <xsd:enumeration value="klbm/in"/>
      <xsd:enumeration value="lbm/ft"/>
   </xsd:restriction>
</xsd:simpleType>
```

2.2.1 Guidelines for Specifying a Quantity Class

If there are multiple quantity classes with the same dimension, then any of those classes may be designated as a specialization of the dimensional base by specifying that dimensional base as an alternative base for conversion. The specialized base must then have an underlying definition that matches the dimensional base. That is, all bases related to a dimension tie back to the dimensional base.

For example, *energy per area* (whose base is *J/m2*) is a specialization of *force per length* (whose base is *N/m*). The alternative base for *energy per area* is *N/m* and because *J=N.m*, the underlying definition of *J/m2* is *N/m*. It is not required for the specialized base to be a member of the class—but it cannot then be a member of any other class. That is, a base unit may be created solely for clarifying the underlying semantics of a class. For example, the base of *thermal diffusivity* is *W.m2.deltaK/(J.deltaK)*, which is not a member of the class. It provides a semantic distinction between other specializations of *area per time*, but it is not a commonly used term.

A class that represents a specialization of the dimension may have members whose base is the dimensional base but not vice versa. For example, *length per length* has the member *Euc*, which is the base of the more general "dimensionless" class, but "dimensionless" does not have *m/m* as a member. Unit conversion may occur through any combination of the two bases because they are numerically equivalent. If there is only one class for a particular dimension, then the base must match the dimensional base and an alternative base must not be specified.

2.2.1.1 Unit Dimension of None

The unit dimension of *none* represents special cases where the traditional dimensional analysis of a unit does not work. For example, a *bel* represents the logarithm of a ratio. A class with a dimension of *none* must not have an alternative base (i.e., there is no specialization of *none*). The unit dimension of *none* does not have a base unit or a canonical unit as is represented by the invalid symbol *0* (zero).

2.3 Aliases

The importance of creating a robust set of unit aliases in any practical implementation of any units standard cannot be overstated. This section describes several use cases involving units that need unit aliases.

Any software that wants to handle data from outside a specific well-protected environment must be able to understand unit symbols that do not conform to the Standard. For example, even for the lowly “m”, we find “M”, “meter”, “meters”, “metres”, “Meters”, etc. Casing and valid spelling variants for every unit abound; therefore, when looking for a conversion, useful software and data management environments allow for storing these aliases and referencing them, in addition to the true standard values. The best practice for this situation is to use the alias to recognize the unit associated with an incoming piece of data but to store and return it with the standard unit.

As a corollary to the previous point, an alias table can also hold variants like “MEters” or “metrers” or other common misspellings of the word, when that is useful to a loader program.

Another use for aliases is in display units. There are occasions where an underlying unit symbol is too complicated to display to users but is correct for purposes of data storage or transfer. For example, the unit “1E-6 A2.s4/(m2.kg)” is correct, but a user would prefer to see “uF” as a unit of capacitance on a user interface.

The same is true for units that are unfamiliar or academic. For example, this standard continues the value from the prior POSC standard for a dimensionless number as the “Euc” or Euclid. This value is needed because so many applications—including Energistics transfer standards—have a mandatory unit of measure attribute. However, on a user interface, you may choose to display a space character or a null string alias because the “Euc” symbol itself is not very familiar to the user community. The same is true for the new deltaK temperature gradient unit, which should probably be displayed simply as K until the community becomes accustomed to the new symbols.

Aliases with namespaces can also be used as a way to organize aliases and anticipate when certain categories of variants will appear. For example, it is well known that the older Schlumberger LIS file format always used uppercase letters for its units. Creating a set of unit aliases within the “LIS” namespace would allow creation of a set of such uppercase units, which could be used to limit a log file loader to that set of aliases, if it knew it was loading a LIS file.

3 Special Considerations

This chapter provides guidance for how to work with some of the more unusual or complex units-of-measure situations.

3.1 Ratios

When there are multiple classes for the same dimension, all but one class generally represents, wholly or partially, a ratio of identical quantities that dimensionally cancel. However, the semantics of that ratio is significant. For example, a ratio of volumes represents a fundamentally different measurement than a ratio of masses, and a ratio of volume per area represents a different concept than measuring a length. To capture those semantics, the specified base units explicitly capture the notion of the ratio (e.g., *m3/m3* or *m3/m2*). Thus, the base units are instrumental in distinguishing between classes of the same dimension.

The class of "dimensionless" is reserved for concepts whose dimensions cancel via a combination of various quantities, rather than via a ratio of identical quantities. That is, a *volume per volume* is a distinct specialization of the dimensionless concept.

3.2 Point Temperature versus Temperature Interval

SI uses the kelvin unit for both a temperature that represents a point on a temperature scale as well as for a temperature interval (i.e., a difference between two points on that scale). Unfortunately, a temperature interval cannot be converted to a point on a temperature scale or vice versa. Similar to conversions of one-dimensional coordinates, conversions of point temperatures commonly involve an offset from an origin. Conversion of an interval does not involve an offset because any offset values cancel. This discussion also applies to the Celsius, Rankine and Fahrenheit scales, except that the offset for Kelvin and Rankine scales is zero.

To resolve this issue, the *Energistics Unit of Measure Dictionary* strongly distinguishes between point temperature units (K, degC, degR, degF) and interval temperature units (deltaK, deltaC, deltaR, deltaF). Each group is part of a different class with different base units and a different dimension. Because all derived units involve temperature intervals, this is the concept of temperature that is used in a dimensional analysis. The dimension used for point temperatures is *none*, because it does not represent a normal dimensional concept.

3.3 Conditions of Measurement

NIST and SI specify that it is incorrect to attach information about the conditions of measurement to a unit. As a result, the *Energistics Unit of Measure Dictionary* does not contain concepts such as gauge pressure, absolute pressure, brake horsepower, or standard volumes.

These conditions of measurement concepts must be associated with the property—not the value and unit combination. When mapping from dictionaries whose unit contains this information or adding new units to the dictionary, care must be taken to insure that the measurement condition is somehow associated with the property.

Conversion between these values typically requires external information. For example, converting from gauge pressure to absolute pressure requires the ambient pressure. As another example, the number of molecules within a cubic meter may change with pressure, but the definition of a cubic meter is independent of pressure. Knowledge of the external information must be captured with the property—not the unit.

Note that the units described in this section (3.3) are not part of the Standard. Rather, this section contains guidance on how implementers of this Standard can most effectively handle common units encountered in the field.

3.3.1 Gauge and Absolute Pressure

When dealing with properties involving pressure, a good default is to assume that all properties represent gauge pressure, unless the property specifically asserts that it represents an absolute pressure.

Be aware that some conversion factors add or subtract one "standard atmosphere" (a standardized average pressure at sea level) as part of the conversion. Unfortunately, a value whose unit is flagged as absolute does not tell you if the value represents a measured absolute value at an elevation of 6,000 ft or if it was a gauge pressure with a standard atmosphere added to the measured value.

Unless a property is known to represent an absolute pressure, a best practice is to copy all gauge values and to subtract one standard atmosphere *(= (101325 * 6.4516e-4)/4.4482216152605 psi*) from absolute values before choosing the target unit.

When a property can represent gauge or absolute conditions, then a flag must be associated with that property to indicate the measurement condition.

3.3.2 Brake Horsepower

The concept of brake horsepower refers to how the measurement was taken (i.e., excluding power losses caused by auxiliary components) and does not affect the underlying definition of horsepower.

- If the concept of brake is important to the property, then that information must be captured by the property itself.
- Otherwise, a best practice is to copy the value and use the equivalent horsepower unit in the target dictionary.

3.3.3 Standard Temperature and Pressure

Concepts involving standard temperature and pressure generally apply to volumes of gasses so that the number of molecules can be compared. That is, two identical volumes of the same gas at the same temperature and pressure contain about the same number of molecules of that gas and a "standard cubic meter" does NOT refer to any variation of the concept of a "cubic meter". Therefore any standard volume unit maps to the equivalent normal volume such as *m3* or *ft3*.

Knowledge about the pressure and temperature must somehow be associated with the property. Commonly, only a pressure of one standard atmosphere is involved in all variations of standard, such that only the temperature needs to be captured (but this is not universally true). In a database management system (DBMS), this situation might require that an extra column be added for this information. Alternatively, a separate property might be defined for each combination of conditions. The relationship between the conditions is captured by the ideal gas law formula:

$P2V2/T2 = P1V1/T1$

where,

V = volume

P = pressure

T = temperature in Kelvin (absolute-temperature as opposed to temperature-interval).

Thus $V2 = V1P1T2/(P2T1) = V1(T2/T1)(P1/P2)$, and if the pressures are the same,

then $V2 = V1(T2/T1)$.

So another alternative is to convert all values to the same condition. However, after the conditions are unknown, then conversion to a different condition is no longer possible, but conversion to a different volume unit is still possible.

The best practice is to capture the conditions as part of the data model, rather than as part of the unit. However, for legacy systems that cannot be modified, a work-around is to create special codes that combine the concept of the unit symbol and the concept of the condition as described in the next section.

3.3.3.1 Standard Condition Codes

This section defines a recommended practice for defining special symbols for handling standard conditions in legacy situations. The assumption is that these systems cannot be updated to capture the conditions independent of the unit. A naming convention is defined for combining conditions into a code that conforms to the *Energistics Unit Symbol Grammar*. Those codes are then treated like any other unit symbol. When these codes are imported into a system that can capture the conditions independent of the unit, then normal units can be used without any conversion required.

3.3.3.1.1 Code Naming Convention

By following this naming convention, it should be possible to unambiguously support any condition for a volume.

1. Start with a normative non-qualified unit symbol (e.g., *ft3*).
2. Add brackets containing @ to indicate that the symbol is at the subsequent conditions (e.g., *ft3[@]*).
3. Add the first condition after the @ sign. For temperature and pressure, specify the temperature first (e.g., *ft3[@60degF]*). A condition is indicated by an integer number followed immediately by an atom symbol (e.g., *60degF*). That is, no decimal points or spaces are allowed.
4. For various condition situations, follow these rules:
   - For multiple conditions, add a comma followed by the next condition (e.g., *ft3[@60degF,1atm]*).
   - For floating point conditions, use a prefixed symbol that allows the value to be an integer (e.g., *m3[@15degC, 100kPa]*).
   - Use a unit symbol that allows the number to be the smallest whole number (e.g., *1atm* instead of *101325Pa*).
   - The value of a condition should be an actual value as opposed to a nominal value.

3.3.3.1.2 Example Codes

The conversion data for the following codes should be considered to be inexact. While the calculations with respect to the ideal gas law and the units are exact, the ideal gas law inherently represents an approximation of the volume.

If these values were loaded into a unit dictionary, then they would have the following values:

- symbol=*code*
- dimension=*L3*
- isSI=*false*, category=*atom*
- conversionRef=*DEFINITION*
- isExact=*false*,
- *A=0*, *B=b*, *C=c* and *D=0*. They would be part of a new class such as volume with conditions.
- Column b is equal to *B*T2*P1*
- Column C is equal to *C*T1*P2,* where *B* and *C* are the conversion factors for the normal unit. But when the pressures are the same, this simplifies to *B*T2* and *C*T1*.

code	baseUnit	b	c	description
m3[@15degC,1atm]	IS-BASE			One cubic meter at a temperature of 15 degrees Celsius and a pressure of one standard atmosphere. The conversion uses the ideal gas law.
m3[@0degC,1atm]	m3[@15degC,1atm]	288.15	273.15	One cubic meter at a temperature of zero degrees Celsius and a pressure of one standard atmosphere. The conversion uses the ideal gas law.
ft3[@60degF,1atm]	m3[@15degC,1atm]	73.4354941093632	2598.35	One cubic foot at a temperature of 15 degrees Celsius and a pressure of one standard atmosphere. The conversion uses the ideal gas law, in addition to the normal unit conversion.

The following table represents derived variations of the above atom codes. Note that the conversion factors represent applying the unit's multiplier to the above factors. If these values were loaded into a unit dictionary, then they would have the same values as above except for category=*derived* and conversionRef=*DERIVED*.

code	baseUnit	b	c	description
1E6 m3[@15degC,1atm]	m3[@15degC,1atm]	1E6	1	One million cubic meters at a temperature of 15 degrees Celsius and a pressure of one standard atmosphere.
1E6 ft3[@60degF,1atm]	m3[@15degC,1atm]	7.34354941093632E+7	2598.35	One million cubic feet at a temperature of 15 degrees Celsius and a pressure of one standard atmosphere.

A ratio of units at the same condition is equivalent to a ratio of normal units, because the ideal gas law factors cancel out. For example, *ft3[@60degF,1atm]/m3[@60degF,1atm]* is equivalent to *ft3/m3*. That is, the conditions become irrelevant.

3.3.4 Reservoir and Stock Tank Barrel

The concepts of reservoir barrel and stock tank barrel are slightly different.

- A reservoir barrel represents the volume of oil in the reservoir.
- A stock tank barrel represents the volume of that same oil after its pressure and temperature have stabilized on the surface.

When the temperature and pressure are reduced at the surface, then dissolved gasses is released, which reduces the volume of that oil. Regardless, because there are no compression or expansion considerations for the liquid, either concept represents 42 gallons of liquid, with the only difference being where the volume is measured.

The property should capture the concept of either reservoir or stock tank, while the target unit should just be a 42-gallon barrel. Note that after any unit representing a reservoir or stock tank barrel is converted to anything else, the original concept of where the measurement occurred is lost (which is why the concept should exist on the property).

4 Mapping

This chapter outlines best practices for mapping a unit dictionary to the *Energistics Unit of Measure Dictionary*. This information is supplemental to the Energistics-provided XML mapping files (included as part of the *UOM Standards* download) that assert mappings from existing industry dictionaries. The mappings described here are predominately between aliases that have the same underlying definition.

4.1 Conversion Factors

The conversion factors are defined as *y=(A+Bx)/(C+Dx).*

However, because *D=0* for all current units, a common solution is to reduce the conversion factor definition to *y=offset+scale*x*

where,

offset = *A/C*

scale = *B/C.*

Regardless of which variation of the formula is used, the division should occur using calculations that retain a precision that is equal to or greater than is possible on the target machine.

4.2 Mapping State

Each Energistics mapped unit includes one of the following states to indicate how well the *mapsTo* content in the target (Energistics) dictionary relates to the *mapsFrom* content in the source dictionary.

State	Description
identical	Indicates that the mapsTo conversion factor is identical to the mapsFrom conversion factor. A copy of the value should be assigned the unit mapsTo.
precision	Indicates that the mapsTo conversion factor has greater precision (i.e., changes in least significant digits). If there is knowledge that the value was converted from a different unit in the source dictionary, then consider using that conversion factor to reproduce the original value before converting to the target unit using a more precise target conversion factor. Otherwise, a copy of the value should be assigned the unit mapsTo.
corrected	Indicates that the mapsTo conversion factor represents a correction to the mapsFrom conversion factor. That is, the mapsFrom conversion factor has a problem. The value should be treated the same as for a state of precision.
conversion	A direct mapping is not specified, but the value can be converted to a supported item if the mapsFrom definition is presumed to be valid.
conditional	Indicates a mapping to a similar alternative with a different underlying conversion factor in the less significant digits. Other mapping alternatives may be available.
unsupported	A mapping is not specified because the underlying concept is not supported. For information on how to add a new concept to a dictionary, see Section 5.1, page 17.

4.3 Year

The length of a year inherently varies with time, which has resulted in several definitions of a year, depending on the usage. The SI specifications recommend the use of "*a"* as a symbol for year but does not assign any particular definition to the symbol. The *Energistics Unit of Measure Dictionary* supports two particular usages: short time periods and long time periods.

- **For short time periods**, the Julian definition that accounts for leap year (but not leap seconds) is used. This definition means that a year has exactly 365.25 days. This definition has been assigned to the "*a"* symbol, which is used with any derived symbol that represents a "per year" concept and with any variation that represents one hundred years or less.

- **For long time periods**, the tropical definition adopted by the International Union of Pure and Applied Chemistry (IUPAC) and the International Union of Geological Sciences (IUGS) for use with very long time periods has been adopted. This definition has been assigned to the symbol "*a[t]*", which is used with any variation that represents a thousand years or more.

One feature of the usage of years in the oil and gas industry is that there was probably no assumption of an underlying conversion to seconds when the original value was defined. If someone asserts that something is one million years old, then that value should probably be maintained, regardless of which dictionary is being used to assign a symbol.

- The recommended best practice is to copy the value and assign the most appropriate symbol from the target dictionary.
- For systems that internally keep all time values in a common unit such as seconds, large values should be converted to year using the definition within that system, before assigning a symbol in the target dictionary. One reason for not converting a value in years when moving between dictionaries is because the precision of a value involving year will generally be low. If you convert *1e6* years from a sidereal definition to a tropical definition, then the resultant value is something like *1.00003880e+6*, where the implied precision exceeds the precision of the original value.
- A more general statement of the best practice is to avoid, where reasonable, any conversion of the time component in values involving year. Conversion of something like "*ft/a*" to "*m/a*" does not involve any change to the time component.

5 Extending a Unit Dictionary

While efforts have been made to be as comprehensive as feasible with this *UOM Standard*, some companies and organizations have very specialized needs and thus must extend the dictionary. This chapter provides rules for creating *UOM Standard*-compliant components and derived symbols.

5.1 Creating a New Unit

Observe the following rules when creating new units of measure:

- Any new symbol must conform to the *Energistics Unit Symbol Grammar Specification,* which represents a subset variation of the grammar allowed by SI. A key aspect of this grammar is that it strongly distinguishes between atoms and prefixed-atoms, within the context of creating a resulting symbol. Atoms represent the elemental symbols from which all other symbols are derived. For example, if *s* represents the atom for second, then *ms* represents the prefixed-atom for millisecond. All other symbols are derived from these two kinds of components.
- All symbols must be unique within a dictionary.
- All symbols used as a base for conversion must be SI compliant (including the use of symbols allowed for use with SI by the dictionary). This requirement allows a dictionary to be used within a context where only SI units are used and insures the applicability across many domains. The *Energistics Unit of Measure Dictionary* allows certain units defined by the American Petroleum Institute (API) to be used with SI. It also allows units that are allowed by the National Institute of Standards and Technology (NIST).
- Each unit must be assigned to be a member of at least one quantity class (See definition, Section 2.2 (page 8).) An exception is that the base of a class with an alternative base is not required to be a member of the class, but if not a member, then the alternative base must be a member. The class must have the same dimension as all members of the class. A unit in a quantity class may be converted to any other unit in that class without loss of semantics because the conversion represents a dimensionless factor of one. Two values of a class that have the same unit may be compared in magnitude (e.g., greater than).

5.1.1 Creating a New Atom

Observe the following rules when creating a new atom:

- **Singular Term.** When assigning a name to a symbol, use singular terms such as *foot* instead of *feet*. While the plural form may be appropriate for written and spoken text, the underlying unit definition is always singular. Note however that some singular terms end in an *s* such as *seimens*.
- **Atoms and Prefixes**. An atom must not represent a prefix (e.g., *k* for kilo representing *1E3*). Rather, a prefix must be combined with an atom (e.g., *km* for kilometer). That is, an atom cannot represent only a multiplier. Alternatively, a prefix may be associated with *Euc*, which represents the SI *1* (e.g., *mEuc*).
- **Variant Definitions and Qualifiers**. Many non-SI (i.e., customary) unit concepts have several variant definitions. Concepts like *inch, foot, yard, rod, link, chain, mile, hundredweight, fluid-ounce, gallon, horsepower, ounce-mass, pound-mass, pint, quart, ton-mass, ton-force, calorie, BTU, therm*, and *year* have historically developed different meanings within different contexts. Variations in the same underlying concepts are captured using the qualifier (i.e., enclosed square bracket) notation from the grammar (refer to the *Energistics Unit of Measure Grammar*).

  For example, *gal[US]* and *gal[UK]* are used to capture two variations of the concept of a gallon. This convention makes it easier to understand when there is more than one definition to a concept.

  - If there is one internationally accepted definition that is used in most derived symbols, then the brackets are omitted (e.g., *ft* represents the international foot while there are many historical variants).
  - If the qualifier represents a quantity, then the normative unit symbol must be specified immediately after the numeric value (e.g., *[60degF]*). Note that the value of the quantity can be a nominal value because the qualifier is just a label. Do NOT use the symbol *@* in a qualifier, because it is reserved for legacy scenarios (see Section 3.3.3.1 (page 12)).

- **Conversion Data.** The conversion data is a fundamental part of the definition of a symbol and serves to remove any ambiguity involving a definition. Because of the importance of the conversion data, a conversion reference MUST be defined for all atoms. The NIST *Guide for the Use of the International System of Units (SI)* is an excellent source for information on the SI system and on conversion data. While NIST defines many conversions to only seven digits, it indicates when a conversion is exact and it can be used to help validate a more exact derived value. Other good references are governmental decrees that assert a legal definition in a country. The European Petroleum Survey Group (EPSG) is an internationally recognized source related to surveying (i.e., historical length measurements) and recognized experts on geodetic issues. EPSG has been absorbed into the Surveying and Position Committee of the International Association of Oil & Gas Producers (OGP).
- **Aliases.** Beware of aliases for the same definition. For example, a *gon* is also known as a *gradian*, *grad*, or *grade*; an *mho* is another name used for *siemens*; the term *micron* represents a *micrometer*. Rather than creating a separate atom for each alternative, these alternatives must be captured in the description of the preferred symbol so that a textual search can find them.
- **Underlying Definition.** The underlying definition specifies that the true definition of an atom is dependent on one or more atoms. The underlying definition is not needed for prefixed symbols and most derived symbols because the relationship to other atoms is already explicit. The definitive set of underlying atoms is defined by those atoms that do not have an underlying definition. The conversion data for atoms with underlying definitions can be tested against that definition using the same technique that is used for deriving conversion data.

  Currently, testing validates the consistency of all conversion data to 15 digits of precision. Underlying definitions are also defined for most base conversion symbols, which would potentially allow for conversions to SI base units (i.e., because of the "coherent" nature of SI, the exact conversion data is inherent to the symbol).

  NOTE: The optional underlying definition is not the normative conversion data—it is only used to cross-validate the data. If any discrepancies occur, then the mandatory conversion data must be used.

5.1.2 Creating a Prefixed Atom

A prefixed atom is constructed by appending a prefix defined in the grammar to the beginning of an existing atom. Observe these rules when constructing a prefixed atom:

- Only prefixes defined in the grammar are allowed.
- A prefix cannot be combined with an existing prefixed atom. That is, compound prefixes (e.g., *mum* for millimicrometer) are expressly forbidden.
- Do not arbitrarily combine all prefixes with an atom. Only combinations that are in common use should be created.
- The resultant symbol must be unique across all symbols.
- When assigning a name to the symbol, there are NO spaces between the prefix name and the atom name, because prefix names are inseparable from the unit names to which they are attached. Thus, for example, *millimeter*, *micropascal*, and *meganewton* are single words.

5.1.3 Creating New Derived Symbols

A derived symbol contains one or more of the following:

- A space (following a multiplier)
- A period ".", which indicates multiplication
- A solidus "/", which indicates division
- Parenthesis "()", which indicate grouping
- An atom exponent (e.g., *m2*)

In addition:

- A derived symbol contains one or more component symbols (i.e., an atom or prefixed atom). Each contained component symbol must independently exist as a symbol so that its conversion data can be defined. That is, a derived symbol must not contain unknown concepts.
- All components and the resultant symbol must belong to the same dictionary. The order of components within the derived symbol MUST be consistent (by dimension) for all members of the same quantity class. This consistency helps prevent accidental semantic duplications (e.g., *x.y* is semantically the same as *y.x*).
- The conversion data for the derived symbol must be consistent with the conversion of its component symbols. The category of the resultant symbol must be "derived".

5.1.4 Assigning a Category

The category of a unit explicitly captures whether the symbol is an atom, a prefixed atom, or a derived expression. The categories listed in the table below are allowed.

Knowledge of the atom is the key to ensuring that everything else is consistent. A prefixed atom is inherently defined by the atom and by the multiplier associated with the prefix. The conversion data of all prefixed and derived symbols is inherently defined by the multipliers, powers, prefixes, and atoms associated with that expression. While the dictionary explicitly provides conversion information for each symbol, it is tested against the atom definitions.

Category	Description
atom-base	Indicates an atom that is an SI base unit. Note that *g* is an atom-base for assigning prefixes, while *kg* is a prefixed atom that is a base for conversion.
atom-special	Indicates an atom that represents combinations of other units (e.g., *psi*, *Pa*). Basically, it indicates a special term has been assigned to a commonly used combination of components.
atom-allowed	Indicates an atom that is not SI but is allowed for use with SI by the dictionary.
atom	Indicates a symbol that cannot be parsed into sub-components.
prefixed	Indicates a symbol that is the concatenation of a prefix with an atom (e.g., *km, kpsi, kPa* versus *knot*). The atom must independently exist. Prefixed atoms cannot themselves be prefixed.
derived	Indicates a symbol that contains a space, period, solidus, parenthesis or power. It contains atoms and prefixed-atoms (possibly raised to a power, such as *m2*, *km2*). The space indicates the presence of a multiplier.

5.2 Deriving Conversion Data

A key characteristic of the *Energistics Unit Symbol Grammar Specification* is that each unit symbol has a defined conversion to a symbol that represents the "base for conversion." All units that are a member of the same quantity class can be converted to each other by converting through the base unit. The conversion does not represent any loss of semantics because the conversion represents a dimensionless factor of one. Logically, you convert a value in a source unit to a value in the base unit and then convert the value in the base unit to the value in the target unit.

This section:

- Provides conversion formulas.
- Explains how to derive conversion information for a unit expression.
- Explains how to perform exact conversions.

5.2.1 Conversion Formulas

The formula associated with the conversion data is:

$y=(A+Bx)/(C+Dx)$

where,

x represents a value in the source unit

y represents the resulting value in the base unit (Dx indicates D times x).

If we solve for x, then:

$y(C+Dx)=A+Bx$

$Cy+Dxy=A+Bx$

$Dxy-Bx=A-Cy$

$x(Dy-B)=A-Cy$

$x=(A-Cy)/(Dy-B)$

which represents the reverse conversion from a value in the base unit.

Using lower case constants for a target z (with the same base associated with y), then we have:

$z=(a-cy)/(dy-b)$.

Substituting $(A+Bx)/(C+Dx)$ for y, we get the general formula for conversion from x to z:

$z=(a-c((A+Bx)/(C+Dx))) / (d((A+Bx)/(C+Dx))-b)$.

However, if $A=a=D=d=0$, then the formula reduces to a simple ratio of factors:

$z=(-cBx/C)/(-b)$

$z=(cB/Cb)x$

$z=((B/C)/(b/c))x$.

If the ratio B/C represents a factor, F, (and $A=a=D=d=0$) then:

$y=Fx$

$z=(F/f)x$

where f represents the ratio b/c.

5.2.2 Deriving Conversion Information for a Unit Expression

Given a unit expression, we can derive the conversion to the "base for conversion" symbol; complete the following steps:

1. Identify all of the components in the expression and whether they represent a numerator or a denominator. Care must be exercised when processing a symbol containing multiple slashes (e.g., in *(bbl/d)/(bbl/d)* the second *d* is logically a numerator as in *(bbl/d) * (d/bbl))*.
2. For each part of the symbol determine a conversion factor:
    a. For a component symbol, determine the factor, $F=(B/C)$, for conversion to the component's base unit.
    b. For any multiplier part (e.g., *1E3*), the multiplier is the factor.
    c. The factor for a component with an exponent is the component's factor raised to the same exponent.
3. Multiply the factors for all parts that represent numerators and divide by all factors that represent denominators.

The resulting value represents the factor for converting the derived symbol to its base unit. For simplicity, *B* can be set to the factor with *C=1*.

Using only the factor in the above process takes advantage of the fact that all current symbols have *D*=0 and no derived symbol will contain components where *A* is not zero. Currently, only temperature atoms have a non-zero, *A*, but only temperature-intervals are used in derived symbols (i.e., the *A*'s cancel).

Other factors that allow the above to work are if the base units are one of the following:

- SI base units.
- SI special symbols, which reduce to an expression in terms of SI base units.
- Dimensionless SI allowed units.
- Non-dimensional units with very limited alternatives.

A conversion of symbol components to their individual bases inherently reduces the whole symbol to its base or to an equivalent expression.

5.2.3 Exact Conversions

The conversion information is defined to the maximum available precision. For example, a value of *PI* in a conversion value indicates that the irrational value 3.14159... must be substituted (with a number of digits that is appropriate for the implementation), while a value of *2*PI* indicates that the value of *Pi* must be multiplied by 2 before substitution, etc. Conversions involving *PI* are considered to be exact because extremely precise values can be substituted.

If it is desirable to derive exact conversion information (e.g., for use across machines with different precision), then the *B* and *C* values (from the conversion formulas above) must be retained instead of calculating *F*, and no division should occur when determining the values for *B* and *C*. For components that are in the denominator, invert the *B/C* ratio such that *C* becomes a numerator and *B* becomes a denominator. For example, if the symbol for the unit *U1* is *S2/S3* where *S2* and *S3* are component symbols, then the conversion for *U1* is:

(B2/C2)/(B3/C3)

(B2/C2)(C3/B3)*

*(B2*C3)/(C2*B3)*

B1/C1

where *B1=(B2*C3)* and *C1=(C2*B3)*.

If both *B2/C2* and *B3/C3* are exact, then *B1/C1* will be exact.

For normative definitions that are independent of any implementation, the conversion data should be captured as text data so that truncation or alteration is not unintentionally applied to the data. Any multiplication should be performed on a device with the maximum available precision. For example, the Microsoft Windows calculator in scientific mode in a 64-bit operating system can retain 32 digits of precision. However, if stored as numbers in the Microsoft Excel or Access applications, that precision is not retained.

Note that, if the target unit is a variant of the source unit, then an inexact conversion to a base is canceled out. Thus, a conversion from *electronvolt* to *millielectronvolt* is exact (to the level of machine precision) even though the conversion from each to the base is not exact.

5.3 Validation Tests

This section lists validation tests that must be applied to the data. It is assumed that no atom represents a prefix.

The following terminology is used in this section:

- Any unqualified reference to "base" refers to the base for conversion.
- The term "dimensional base" refers to a base in the unit dimension set.

- When two things must "match", then they must be the same (e.g., the symbols must be equal).
- "SI compliant" means that all components of a symbol are SI and there is no multiplier (conformance to the grammar insures the rest). A symbol that is SI compliant may incorporate non-SI components that are allowed to be used with SI by the dictionary (e.g., Energistics allows some API units and units allowed by NIST).

5.3.1 XSD Schema

The data must validate against the provided XSD schema file, which defines constraints (e.g., optionality and string lengths) on the XML file. It also validates unit symbols against the requirements of the *Energistics Unit Symbol Grammar Specification* and validates the syntax of the unit dimension.

5.3.2 Unit Dimension Set

- The name must be unique within the dimension set.
- The dimension must be unique within the dimension set.
- The base must be unique within the dimension set.
- The canonical unit must match the canonical unit derived from the base.
- The base must be SI compliant.
- The dimension base must be referenced by a class's base for conversion, a class's alternative base, and/or a unit's underlying definition.
- The dimension must be referenced by a class.

5.3.3 Quantity Class Set

- The name must be unique within the class set.
- The base must be unique within the class set.
- If the dimension is not *none* and an alternative base is not specified, then either the base or the underlying definition of the base must match the dimensional base.
- The dimension must exist in the unit dimension set.
- An alternative base must match the dimensional base.
- An alternative base must not match the base.
- An alternative base must not be specified unless there are multiple classes with the same dimension.
- If an alternative base is specified, then the underlying definition of the base must match the dimensional base.
- If an alternative base is specified, then either the alternative base must be a member or at least one member unit must have a base that matches the alternative base.
- The base must be SI compliant.
- At least one member unit must be SI compliant.
- Each class must have at least one member unit.
- Each member unit must either: a) match the base or b) must have a base that matches the base of the class or the alternative base of the class.
- A class of dimension *none* must not have an alternative base.
- Each member unit must have the same dimension as the class.
- Either the base or the alternative base must be member units.
- The base must exist in the unit set.
- The alternative base must exist in the unit set.
- Each member unit must exist in the unit set.

5.3.4 Unit Set

- The name must be unique within the unit set.
- The symbol must be unique within the unit set.
- The conversion reference must exist in the reference set.
- The dimension derived from the symbol must match the dimension assigned to the unit.
- Unless the unit is the base of a class with an alternative base, the unit must be a member of at least one class.
- Each component of a derived symbol must independently exist in the unit set.
- The base must exist in the unit set.
- A symbol flagged as SI must be composed of SI components.
- If any component has a dimension of *none* then the containing symbol must have a dimension of *none*.
- A base of dimension *none* must not have an underlying definition.
- A base unit must not contain prefixed components (except for kg).
- A symbol flagged as SI cannot have category=*atom*.
- A symbol flagged as SI, an atom, or a prefixed-atom must not contain a space.
- A symbol with a category that is not derived must not contain a period, slash, parenthesis, space, or exponent.
- A symbol whose category is derived must contain a period, slash, parenthesis, space, or exponent.
- The conversion data must be consistent with the conversion of any components.
- The conversion data must be consistent with the conversion of any underlying definition.
- For a prefixed symbol, the underlying atom must exist in the unit set.
- The name of a prefixed-atom must be a concatenation of its prefix name with its atom name.
- The conversion data of a prefixed symbol must equal the conversion of the underlying atom times the multiplier associated with the prefix.

5.3.5 Conversion Reference

- The ID must be unique.

5.3.6 Class Integer Code (Energistics only)

- The codes must be unique within the integer set.
- The terms must be unique within the integer set.
- Unless it is deprecated, the term must match an existing class name.
- For each class, a unit must be assigned that is the base, the alternative base, or the underlying definition of the base.
- The assigned unit must exist in the unit integer code set.
- A code must exist for each class.

5.3.7 Unit Integer Code (Energistics only)

- The codes must be unique within the integer set.
- The terms must be unique within the integer set.
- Unless it is deprecated, the term must match an existing unit symbol.
- Unless it is deprecated, the term must be a base, an alternative base, or the underlying definition of a base.