## LifeTimeVariableType definition

### Overview

The *LifeTimeVariableType* defines *Variables* representing the remaining life-time. It provides generically the remaining life-time and can be used on anything; for example, on machines, actuators or sensors, but also on immaterial things like software. It is formally defined in Table 1.

### VariableType definition

Table 1 – LifeTimeVariableType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | LifeTimeVariableType | | | | |
| IsAbstract | False | | | | |
| ValueRank | −1 (−1 = Scalar) | | | | |
| DataType | Number | | | | |
| Description | Remaining life-time | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:AnalogUnitType | | | | | |
| 0:HasProperty | Variable | StartValue | 0:Number | 0:PropertyType | M |
| 0:HasProperty | Variable | LimitValue | 0:Number | 0:PropertyType | M |
| 0:HasProperty | Variable | Indication | 0:NodeId | 0:PropertyType | O |
| 0:HasProperty | Variable | WarningValue | 0:Number | 0:PropertyType | O |
| **Conformance Units** | | | | | |
| TBD | | | | | |

The mandatory *StartValue* indicates the initial value, when there is still the full life-time left. The engineering unit is the same as for the *Value* of the *Variable*, provided by the *LifeTimeVariableType's* mandatory 0:*EngineeringUnits*, derived from the 0:*AnalogUnitType*. The *DataType* shall be the same as for the *Value* of the *Variable*.

The mandatory *LimitValue* indicates when the end of life-time has been reached. The engineering unit is the same as for the *Value* of the *Variable*, provided by the *LifeTimeVariableType's* mandatory 0:*EngineeringUnits*, derived from the 0:*AnalogUnitType*. The *DataType* shall be the same as for the *Value* of the *Variable*.

If the *StartValue* is larger than the *LimitValue*, the *Value* of the *Variable* is expected to move over the time downwards to the *LimitValue*, otherwise upwards to the *LimitValue*.

The optional *Indication* gives an indication of what is actually measured / represented by the *Value* of the *Variable* and the *StartValue* and *LimitValue*. The mandatory 0:*EngineeringUnits*, derived from the 0:*AnalogUnitType* already does this, to a certain degree. But for example, a length unit does not indicate what length is provided, like the cutting distance, the feed distance or the abraded length of a tool. The *Indication* contains a *NodeId* of the *BaseIndicationType* or a subtype of it, providing a more detailed indication.

Note: It is expected that companion specifications or vendors may define additional subtypes of *BaseIndicationType*.

The optional *WarningValue* indicates the *Value* when the end of life-time is reached soon and may be used to inform the user when reached. Servers may also generate Events when such a limit is reached. If provided, the *WarningValue* shall be between the *StartValue* and the *LimitValue*. The engineering unit is the same as for the *Value* of the *Variable*, provided by the *LifeTimeVariableType's* mandatory 0:*EngineeringUnits*, derived from the 0:*AnalogUnitType*. The *DataType* shall be the same as for the *Value* of the *Variable*.

This VariableType can be used in various cases. Examples include

* The Variable may just indicate the remaining life-time as a percentage value. In that case, the 0:EngineeringUnits is percentage, the StartValue is 100 and the LimitValue is 0 (or vice versa if counted upwards).
* The Variable may represent the number of parts produced. In this case, the 0:EngineeringUnits is One, the StartValue is 0 and the LimitValue the maximum producible parts (e.g. 100 000); or vice versa if the number of remaining parts that still can be produced is provided.
* The Variable may represent the remaining time, for example until a software license becomes invalid. The 0:EngineeringUnits could be “Day”, “Month”, “Year”, etc., the StartValue 0 and the LimitValue the overall duration, or vice versa if the remaining time is provided.

The child *Nodes* of the *LifeTimeVariableType* have additional *Attribute* values defined in Table 2.

Table 2 – LifeTimeVariableType Attribute values for child Nodes

|  |  |
| --- | --- |
| **BrowsePath** | **Description Attribute** |
| TBD | TBD: Same text as above |
| TBD | TBD: Same text as above |
| TBD | TBD: Same text as above |

## BaseIndicationType definition

### Overview

The *BaseIndicationType* defines the base indication of a Variable of *LifeTimeVariableType*, without defining any specific semantic. Servers should use a more specific subtype, if possible. It is formally defined in Table 3.

### ObjectType definition

Table 3 – BaseIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | BaseIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Base indication type not further defining a semantic | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseObjectType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |

## TimeIndicationType definition

### Overview

The *TimeIndicationType* indicates the time the entity has been in use or can still be used. It is formally defined in Table 4.

### ObjectType definition

Table 4 – TimeIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | TimeIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Indicates the time the entity has been in use or can still be used | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseIndicationType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |

## NumberOfPartsIndicationType definition

### Overview

The *NumberOfPartsIndicationType* indicates total number of parts that have been produced or can still be produced. It is formally defined in Table 5.

### ObjectType definition

Table 5 – NumberOfPartsIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | NumberOfPartsIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Indicates the total number of parts that have been produced or can still be produced. | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseIndicationType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |

## NumberOfUsagesIndicationType definition

### Overview

The *NumberOfUsagesIndicationType* indicates counting the process steps the entity has been used or can still be used for (for example usages of a punching tool). It is formally defined in Table 6.

### ObjectType definition

Table 6 – NumberOfUsagesIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | NumberOfUsagesIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Indicates counting the process steps the entity has been used or can still be used for (for example usages of a punching tool). | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseIndicationType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |

## LenghtIndicationType definition

### Overview

The *LenghtIndicationType* indicates the abraded length, for example of a drill. It is formally defined in Table 7.

### ObjectType definition

Table 7 – LenghtIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | LenghtIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Indicates the abraded length, for example of a drill. | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseIndicationType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |

## DiameterIndicationType definition

### Overview

The *DiameterIndicationType* indicates the abraded diameter, for example of a drill. It is formally defined in Table 8.

### ObjectType definition

Table 8 – DiameterIndicationType definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Value** | | | | |
| BrowseName | DiameterIndicationType | | | | |
| IsAbstract | True | | | | |
| Description | Indicates the abraded diameter, for example of a drill. | | | | |
| **References** | **Node Class** | **BrowseName** | **DataType** | **TypeDefinition** | **Other** |
| Subtype of the 0:BaseIndicationType | | | | | |
| **Conformance Units** | | | | | |
| TBD | | | | | |