搜索



# Qt语言学家手册：TS文件格式

*Qt Linguist*使用的TS文件格式由下面介绍的XSD描述，为了您的方便，我们将其包括在内。请注意，在未来的Qt版本中，格式可能会发生变化。

```
<?xml version="1.0" encoding="utf-8"?>
<!--
 !
 ! Some notes to the XSD:
 !
 ! The location element is set as optional since it was introduced first in Qt 4.2.
 ! The userdata element is set as optional since it was introduced first in Qt 4.4.
 ! The vanished message type was introduced first in Qt 5.2.
 !
  -->
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- value contains decimal (e.g. 1000) or hex (e.g. x3e8) unicode encoding of one char -->
  <xs:element name="byte">
      <xs:complexType>
        <xs:attribute name="value" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  <!--
   ! Type used in order to escape byte entities not allowed in an xml document
   ! for instance, only #x9, #xA and #xD are allowed characters below #x20.
    -->
  <xs:complexType name="byte-type" mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="byte" />
      </xs:choice>
  </xs:complexType>
  <!--
   ! extra-something should be described as extra-* but wildcard is not valid in XSD. No better solution found.
   ! extra elements may appear in TS and message elements. Each element may appear
   !   extra-po-flags (comma-space separated list)
   !   extra-loc-layout_id
   !   extra-loc-feature
   !   extra-loc-blank
    -->
  <xs:element name="extra-something" type= "byte-type"/>
  <xs:element name="TS">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="extra-something" />
        <xs:element minOccurs="0" maxOccurs="1" ref="dependencies" />
        <xs:choice minOccurs="1" maxOccurs="unbounded">
          <xs:element ref="context" />
          <xs:element ref="message" />
        </xs:choice>
```

Topics >

```
      <xs:attribute name="language" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="context">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name" />
        <xs:element minOccurs="0" maxOccurs="1" ref="comment" />
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="message"/>
      </xs:sequence>
      <xs:attribute name="encoding" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="dependencies">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="dependency" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="dependency">
    <xs:complexType>
      <xs:attribute name="catalog" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="name" type= "byte-type"/>
<!-- This is "disambiguation" in the (new) API, or "msgctxt" in gettext speak -->
  <xs:element name="comment" type= "byte-type"/>
<!-- Previous content of comment (result of merge) -->
  <xs:element name="oldcomment" type= "byte-type"/>
<!-- The real comment (added by developer/designer) -->
  <xs:element name="extracomment" type= "byte-type"/>
<!-- Comment added by translator -->
  <xs:element name="translatorcomment" type= "byte-type"/>
  <xs:element name="message">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="location" />
        <xs:element minOccurs="0" maxOccurs="1" ref="source" />
        <xs:element minOccurs="0" maxOccurs="1" ref="oldsource" />
        <xs:element minOccurs="0" maxOccurs="1" ref="comment" />
        <xs:element minOccurs="0" maxOccurs="1" ref="oldcomment" />
        <xs:element minOccurs="0" maxOccurs="1" ref="extracomment" />
        <xs:element minOccurs="0" maxOccurs="1" ref="translatorcomment" />
        <xs:element minOccurs="0" maxOccurs="1" ref="translation" />
        <xs:element minOccurs="0" maxOccurs="1" ref="userdata" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="extra-something" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" />
      <xs:attribute default="no" name="numerus">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="yes" />
            <xs:enumeration value="no" />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
<!--
 ! If the line is omitted, the location specifies only a file.
 !
 ! location supports relative specifications as well. Line numbers are
 ! relative (explicitly positive or negative) to the last reference to a
 ! given filename; each file starts with current line 0. If the filename
 ! is omitted, the "current" one is used. For the 1st location in a message,
 ! "current" is the filename used for the 1st location of the previous message.
 ! For subsequent locations, it is the filename used for the previous location
```

```
    <xs:element name="location">
      <xs:complexType>
        <xs:attribute name="filename" type="xs:string" />
        <xs:attribute name="line" type="xs:string" />
      </xs:complexType>
    </xs:element>
    <xs:element name="source" type= "byte-type"/>
  <!-- Previous content of source (result of merge) -->
    <xs:element name="oldsource" type= "byte-type"/>
  <!--
   ! The following should really say one byte-type or several
   ! numerusform or lengthvariant elements.
    -->
    <xs:element name="translation">
      <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="byte" />
          <xs:element ref="numerusform" />
          <xs:element ref="lengthvariant" />
        </xs:choice>
        <!--
         ! If no type is set, the message is "finished".
         ! Length variants must be ordered by falling display length.
         ! variants may not be yes if the message has numerus yes.
          -->
        <xs:attribute name="type">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="unfinished" />
              <xs:enumeration value="vanished" />
              <xs:enumeration value="obsolete" />
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute default="no" name="variants">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="yes" />
              <xs:enumeration value="no" />
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
  <!-- Deprecated. Use extra-something -->
    <xs:element name="userdata" type="xs:string" />
  <!--
   ! The following should really say one byte-type or several
   ! lengthvariant elements.
   ! Length variants must be ordered by falling display length.
    -->
    <xs:element name="numerusform">
      <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="byte" />
          <xs:element ref="lengthvariant" />
        </xs:choice>
        <xs:attribute default="no" name="variants">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="yes" />
              <xs:enumeration value="no" />
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
    <xs:element name="lengthvariant" type= "byte-type"/>
```

‹ Qt语言学家手册：开发人员

Qt语言学家手册：基于文本ID的翻译 ›

©2022 Qt有限公司 此处包含的文档贡献的版权归 他们各自的所有者。此处提供的文档根据自由软件基金会发布的GNU 自由文档许可证版本 1.3的条款进行许可。Qt和相应的徽标是Qt有限公司在芬兰和/或其他国家/地区的商标 全球。所有其他商标均为其各自所有者的财产。

The Qt Company

联系我们

**公司**

关于我们
投资者
编辑部
职业
办公地点

**发 牌**

条款和条件
开源
常见问题

**支持**

支持服务
专业服务
合作 伙伴
训练

**对于客户**

支持中心
下载
Qt登录
联系我们
客户成功案例

**社区**

为Qt做贡献
论坛
维基
下载
市场

© 2022 Qt公司

反馈 登录