GPX 1.1 Schema Documentation

Table of Contents

top

Schema Document Properties

Target Namespace

http://www.topografix.com/GPX/1/1

Element and Attribute Namespaces

- Global element and attribute declarations belong to this schema's target namespace.
- By default, local element declarations belong to this schema's target namespace.
- By default, local attribute declarations have no namespace.

Documentation

GPX schema version 1.1 - For more information on GPX and this schema, visit http://www.topografix.com/gpx.asp GPX uses the following conventions: all coordinates are relative to the WGS84 datum. All measurements are in metric units.

Declared Namespaces

Prefix	Namespace
Default namespace	**http://www.topografix.com/GPX/1/1**
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema

Schema Component Representation

```
<xsd:schema targetNamespace="http://www.topografix.com/GPX/1/1"
elementFormDefault="qualified">
  ...
</xsd:schema>
```

top

Global Schema Components

Element: gpx

Name	gpx
Type	gpxType
Nillable	no
Abstract	no
Documentation	GPX is the root element in the XML file.

XML Instance Representation

```
<gpx
 version="1.1 [1] ? "
 creator="xsd:string [1] ? ">
   <metadata> metadataType </metadata> [0..1] ?
   <wpt> wptType </wpt> [0..*] ?
   <rte> rteType </rte> [0..*] ?
   <trk> trkType </trk> [0..*] ?
   <extensions> extensionsType </extensions> [0..1] ?
</gpx>
```

Schema Component Representation

```
<xsd:element name="gpx" type="gpxType"/>
```

top

Complex Type: gpxType

Parent type:	None
Direct sub-types:	None

Name	gpxType
Abstract	no
Documentation	GPX documents contain a metadata header, followed by waypoints, routes, and tracks. You can add your own elements to the extensions section of the GPX document.

XML Instance Representation

```
<...
 version="1.1 [1] ? "
 creator="xsd:string [1] ? ">
   <metadata> metadataType </metadata> [0..1] ?
   <wpt> wptType </wpt> [0..*] ?
   <rte> rteType </rte> [0..*] ?
   <trk> trkType </trk> [0..*] ?
   <extensions> extensionsType </extensions> [0..1] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="gpxType">
```

```
  <xsd:sequence>
    <xsd:element name="metadata" type="metadataType" minOccurs="0"/>
    <xsd:element name="wpt" type="wptType" minOccurs="0"
    maxOccurs="unbounded"/>
    <xsd:element name="rte" type="rteType" minOccurs="0"
    maxOccurs="unbounded"/>
    <xsd:element name="trk" type="trkType" minOccurs="0"
    maxOccurs="unbounded"/>
    <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="required" fixed="1.1"/>
  <xsd:attribute name="creator" type="xsd:string" use="required"/>
</xsd:complexType>
```

top

Complex Type: metadataType

Parent type:	None
Direct sub-types:	None

Name	metadataType
Abstract	no
Documentation	Information about the GPX file, author, and copyright restrictions goes in the metadata section. Providing rich, meaningful information about your GPX files allows others to search for and use your GPS data.

XML Instance Representation

```
<...>
  <name> xsd:string </name> [0..1]  ?
  <desc> xsd:string </desc> [0..1]  ?
  <author> personType </author> [0..1]  ?
  <copyright> copyrightType </copyright> [0..1]  ?
  <link> linkType </link> [0..*]  ?
  <time> xsd:dateTime </time> [0..1]  ?
  <keywords> xsd:string </keywords> [0..1]  ?
  <bounds> boundsType </bounds> [0..1]  ?
  <extensions> extensionsType </extensions> [0..1]  ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="metadataType">
  <xsd:sequence>
    <-- elements must appear in this order -->
    <xsd:element name="name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="desc" type="xsd:string" minOccurs="0"/>
    <xsd:element name="author" type="personType" minOccurs="0"/>
    <xsd:element name="copyright" type="copyrightType" minOccurs="0"/>
    <xsd:element name="link" type="linkType" minOccurs="0"
    maxOccurs="unbounded"/>
    <xsd:element name="time" type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="keywords" type="xsd:string" minOccurs="0"/>
    <xsd:element name="bounds" type="boundsType" minOccurs="0"/>
    <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: wptType

Parent type:	None
Direct sub-types:	None

Name	wptType
Abstract	no
Documentation	wpt represents a waypoint, point of interest, or named feature on a map.

XML Instance Representation

```
<...
 lat="latitudeType [1]  ? "
 lon="longitudeType [1]  ? ">
   <ele> xsd:decimal </ele> [0..1]  ?
   <time> xsd:dateTime </time> [0..1]  ?
   <magvar> degreesType </magvar> [0..1]  ?
   <geoidheight> xsd:decimal </geoidheight> [0..1]  ?
   <name> xsd:string </name> [0..1]  ?
   <cmt> xsd:string </cmt> [0..1]  ?
   <desc> xsd:string </desc> [0..1]  ?
   <src> xsd:string </src> [0..1]  ?
   <link> linkType </link> [0..*]  ?
   <sym> xsd:string </sym> [0..1]  ?
   <type> xsd:string </type> [0..1]  ?
   <fix> fixType </fix> [0..1]  ?
   <sat> xsd:nonNegativeInteger </sat> [0..1]  ?
   <hdop> xsd:decimal </hdop> [0..1]  ?
   <vdop> xsd:decimal </vdop> [0..1]  ?
   <pdop> xsd:decimal </pdop> [0..1]  ?
   <ageofdgpsdata> xsd:decimal </ageofdgpsdata> [0..1]  ?
   <dgpsid> dgpsStationType </dgpsid> [0..1]  ?
   <extensions> extensionsType </extensions> [0..1]  ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="wptType">
   <xsd:sequence>
      <-- elements must appear in this order -->
      <-- Position info -->
      <xsd:element name="ele" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="time" type="xsd:dateTime" minOccurs="0"/>
      <xsd:element name="magvar" type="degreesType" minOccurs="0"/>
      <xsd:element name="geoidheight" type="xsd:decimal" minOccurs="0"/>
      <-- Description info -->
      <xsd:element name="name" type="xsd:string" minOccurs="0"/>
      <xsd:element name="cmt" type="xsd:string" minOccurs="0"/>
      <xsd:element name="desc" type="xsd:string" minOccurs="0"/>
      <xsd:element name="src" type="xsd:string" minOccurs="0"/>
      <xsd:element name="link" type="linkType" minOccurs="0"
      maxOccurs="unbounded"/>
      <xsd:element name="sym" type="xsd:string" minOccurs="0"/>
      <xsd:element name="type" type="xsd:string" minOccurs="0"/>
      <-- Accuracy info -->
      <xsd:element name="fix" type="fixType" minOccurs="0"/>
      <xsd:element name="sat" type="xsd:nonNegativeInteger" minOccurs="0"/>
      <xsd:element name="hdop" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="vdop" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="pdop" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="ageofdgpsdata" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="dgpsid" type="dgpsStationType" minOccurs="0"/>
      <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="lat" type="latitudeType" use="required"/>
   <xsd:attribute name="lon" type="longitudeType" use="required"/>
</xsd:complexType>
```

top

Complex Type: rteType

Parent type:	None
Direct sub-types:	None

Name	rteType
Abstract	no
Documentation	rte represents route - an ordered list of waypoints representing a series of turn points leading to a destination.

XML Instance Representation

```
<...>
   <name> xsd:string </name> [0..1] ?
   <cmt> xsd:string </cmt> [0..1] ?
   <desc> xsd:string </desc> [0..1] ?
   <src> xsd:string </src> [0..1] ?
   <link> linkType </link> [0..*] ?
   <number> xsd:nonNegativeInteger </number> [0..1] ?
   <type> xsd:string </type> [0..1] ?
   <extensions> extensionsType </extensions> [0..1] ?
   <rtept> wptType </rtept> [0..*] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="rteType">
   <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="0"/>
      <xsd:element name="cmt" type="xsd:string" minOccurs="0"/>
      <xsd:element name="desc" type="xsd:string" minOccurs="0"/>
      <xsd:element name="src" type="xsd:string" minOccurs="0"/>
      <xsd:element name="link" type="linkType" minOccurs="0"
      maxOccurs="unbounded"/>
      <xsd:element name="number" type="xsd:nonNegativeInteger" minOccurs="0"/>
      <xsd:element name="type" type="xsd:string" minOccurs="0"/>
      <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
      <xsd:element name="rtept" type="wptType" minOccurs="0"
      maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: trkType

Parent type:	None

Direct sub-types:	None

Name	trkType
Abstract	no
Documentation	trk represents a track - an ordered list of points describing a path.

XML Instance Representation

```
<...>
  <name> xsd:string </name> [0..1]  ?
  <cmt> xsd:string </cmt> [0..1]  ?
  <desc> xsd:string </desc> [0..1]  ?
  <src> xsd:string </src> [0..1]  ?
  <link> linkType </link> [0..*]  ?
  <number> xsd:nonNegativeInteger </number> [0..1]  ?
  <type> xsd:string </type> [0..1]  ?
  <extensions> extensionsType </extensions> [0..1]  ?
  <trkseg> trksegType </trkseg> [0..*]  ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="trkType">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="cmt" type="xsd:string" minOccurs="0"/>
    <xsd:element name="desc" type="xsd:string" minOccurs="0"/>
    <xsd:element name="src" type="xsd:string" minOccurs="0"/>
    <xsd:element name="link" type="linkType" minOccurs="0"
    maxOccurs="unbounded"/>
    <xsd:element name="number" type="xsd:nonNegativeInteger" minOccurs="0"/>
    <xsd:element name="type" type="xsd:string" minOccurs="0"/>
    <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
    <xsd:element name="trkseg" type="trksegType" minOccurs="0"
    maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: extensionsType

Parent type:	None
Direct sub-types:	None

Name	extensionsType
Abstract	no
Documentation	You can add extend GPX by adding your own elements from another schema here.

XML Instance Representation

```
<...>
  Allow any elements from a namespace other than this schema's namespace (lax
  validation). [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="extensionsType">
   <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: trksegType

Parent type:	None
Direct sub-types:	None

Name	trksegType
Abstract	no
Documentation	A Track Segment holds a list of Track Points which are logically connected in order. To represent a single GPS track where GPS reception was lost, or the GPS receiver was turned off, start a new Track Segment for each continuous span of track data.

XML Instance Representation

```
<...>
   <trkpt> wptType </trkpt> [0..*]  ?
   <extensions> extensionsType </extensions> [0..1]  ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="trksegType">
   <xsd:sequence>
      <-- elements must appear in this order -->
      <xsd:element name="trkpt" type="wptType" minOccurs="0"
      maxOccurs="unbounded"/>
      <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: copyrightType

Parent type:	None
Direct sub-types:	None

Name	copyrightType
Abstract	no
Documentation	Information about the copyright holder and any license governing use of this file. By linking to an appropriate license, you may place your data into the public domain or grant additional usage rights.

XML Instance Representation

```
<...
 author="xsd:string [1]  ? ">
```

```
    <year> xsd:gYear </year> [0..1] ?
    <license> xsd:anyURI </license> [0..1] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="copyrightType">
  <xsd:sequence>
    <-- elements must appear in this order -->
    <xsd:element name="year" type="xsd:gYear" minOccurs="0"/>
    <xsd:element name="license" type="xsd:anyURI" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="author" type="xsd:string" use="required"/>
</xsd:complexType>
```

top

Complex Type: linkType

Parent type:	None
Direct sub-types:	None

Name	linkType
Abstract	no
Documentation	A link to an external resource (Web page, digital photo, video clip, etc) with additional information.

XML Instance Representation

```
<...
 href="xsd:anyURI [1] ?">
    <text> xsd:string </text> [0..1] ?
    <type> xsd:string </type> [0..1] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="linkType">
  <xsd:sequence>
    <-- elements must appear in this order -->
    <xsd:element name="text" type="xsd:string" minOccurs="0"/>
    <xsd:element name="type" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="href" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

top

Complex Type: emailType

Parent type:	None
Direct sub-types:	None

Name	emailType
Abstract	no
Documentation	An email address. Broken into two parts (id and domain) to help prevent

email harvesting.

XML Instance Representation

```
<...
 id="xsd:string [1]  ? "
 domain="xsd:string [1]  ? "/>
```

Schema Component Representation

```
<xsd:complexType name="emailType">
   <xsd:attribute name="id" type="xsd:string" use="required"/>
   <xsd:attribute name="domain" type="xsd:string" use="required"/>
</xsd:complexType>
```

top

Complex Type: personType

Parent type:	None
Direct sub-types:	None

Name	personType
Abstract	no
Documentation	A person or organization.

XML Instance Representation

```
<...>
   <name> xsd:string </name> [0..1]  ?
   <email> emailType </email> [0..1]  ?
   <link> linkType </link> [0..1]  ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="personType">
   <xsd:sequence>
      <-- elements must appear in this order -->
      <xsd:element name="name" type="xsd:string" minOccurs="0"/>
      <xsd:element name="email" type="emailType" minOccurs="0"/>
      <xsd:element name="link" type="linkType" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ptType

Parent type:	None
Direct sub-types:	None

Name	ptType
Abstract	no
Documentation	A geographic point with optional elevation and time. Available for use by other schemas.

XML Instance Representation

```
<...
 lat="latitudeType [1] ?"
 lon="longitudeType [1] ?">
   <ele> xsd:decimal </ele> [0..1] ?
   <time> xsd:dateTime </time> [0..1] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="ptType">
   <xsd:sequence>
      <-- elements must appear in this order -->
      <xsd:element name="ele" type="xsd:decimal" minOccurs="0"/>
      <xsd:element name="time" type="xsd:dateTime" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="lat" type="latitudeType" use="required"/>
   <xsd:attribute name="lon" type="longitudeType" use="required"/>
</xsd:complexType>
```

top

Complex Type: ptsegType

Parent type:	None
Direct sub-types:	None

Name	ptsegType
Abstract	no
Documentation	An ordered sequence of points. (for polygons or polylines, e.g.)

XML Instance Representation

```
<...>
   <pt> ptType </pt> [0..*] ?
</...>
```

Schema Component Representation

```
<xsd:complexType name="ptsegType">
   <xsd:sequence>
      <-- elements must appear in this order -->
      <xsd:element name="pt" type="ptType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: boundsType

Parent type:	None
Direct sub-types:	None

Name	boundsType
Abstract	no

Documentation Two lat/lon pairs defining the extent of an element.

XML Instance Representation

```
<...
 minlat="latitudeType [1]  ? "
 minlon="longitudeType [1]  ? "
 maxlat="latitudeType [1]  ? "
 maxlon="longitudeType [1]  ? "/>
```

Schema Component Representation

```
<xsd:complexType name="boundsType">
   <xsd:attribute name="minlat" type="latitudeType" use="required"/>
   <xsd:attribute name="minlon" type="longitudeType" use="required"/>
   <xsd:attribute name="maxlat" type="latitudeType" use="required"/>
   <xsd:attribute name="maxlon" type="longitudeType" use="required"/>
</xsd:complexType>
```

top

Simple Type: latitudeType

Parent type:	xsd:decimal (derivation method: restriction)
Direct sub-types:	None

Name latitudeType

Content

- Base XSD Type: decimal
- -90.0 <= *value* <= 90.0

Documentation The latitude of the point. Decimal degrees, WGS84 datum.

Schema Component Representation

```
<xsd:simpleType name="latitudeType">
   <xsd:restriction base="xsd:decimal">
      <xsd:minInclusive value="-90.0"/>
      <xsd:maxInclusive value="90.0"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

Simple Type: longitudeType

Parent type:	xsd:decimal (derivation method: restriction)
Direct sub-types:	None

Name longitudeType

Content

- Base XSD Type: decimal
- -180.0 <= *value* < 180.0

Documentation The longitude of the point. Decimal degrees, WGS84 datum.

Schema Component Representation

```
<xsd:simpleType name="longitudeType">
  <xsd:restriction base="xsd:decimal">
    <xsd:minInclusive value="-180.0"/>
    <xsd:maxExclusive value="180.0"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

Simple Type: degreesType

Parent type:	xsd:decimal (derivation method: restriction)
Direct sub-types:	None

Name degreesType

Content

- Base XSD Type: decimal
- 0.0 <= *value* < 360.0

Documentation Used for bearing, heading, course. Units are decimal degrees, true (not magnetic).

Schema Component Representation

```
<xsd:simpleType name="degreesType">
  <xsd:restriction base="xsd:decimal">
    <xsd:minInclusive value="0.0"/>
    <xsd:maxExclusive value="360.0"/>
  </xsd:restriction>
</xsd:simpleType>
```

top

Simple Type: fixType

Parent type:	xsd:string (derivation method: restriction)
Direct sub-types:	None

Name fixType

Content

- Base XSD Type: string
- *value* comes from list: {'none'|'2d'|'3d'|'dgps'|'pps'}

Documentation Type of GPS fix. none means GPS had no fix. To signify "the fix info is unknown, leave out fixType entirely. pps = military signal used

Schema Component Representation

```
<xsd:simpleType name="fixType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="2d"/>
    <xsd:enumeration value="3d"/>
    <xsd:enumeration value="dgps"/>
    <xsd:enumeration value="pps"/>
```

```
   </xsd:restriction>
</xsd:simpleType>
```

top

Simple Type: dgpsStationType

Parent type:	xsd:integer (derivation method: restriction)
Direct sub-types:	None

Name	dgpsStationType
Content	- Base XSD Type: integer - 0 <= *value* <= 1023
Documentation	Represents a differential GPS station.

Schema Component Representation

```
<xsd:simpleType name="dgpsStationType">
   <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="1023"/>
   </xsd:restriction>
</xsd:simpleType>
```

top