Introduction to IOF Interface Standard 3.0

# What is the IOF Interface Standard?

The International Orienteering Federation (IOF) Interface Standard is a specification of an XML data format for the interchange of orienteering-related information (events, entries, start lists, result lists, etc.) between applications, web services and databases. When adopted by orienteering software packages, it will allow the packages to exchange information between each other. This will benefit event organisers and contribute to simplifying the organisation of orienteering events.

# Background

Among orienteers we have a common wish to benefit from using the latest technology.

A clear indication of this is the number of computer systems we use when we organise orienteering events. Each of the systems performs its own specialized task: Handling event entries, course planning, timing competitors, producing results, providing information for event commentary, and checking electronic punches.

It can be quite an undertaking to make all these systems cooperate, and often much hard work is put in at events to ensure safe and reliable transfer of information from one computer system to another.

This leads to the conclusion that there is a need for a simple means to enable all these disparate systems to communicate information in a reliable manner.

# History

The efforts to define an interface standard for orienteering started in October 1998 at an IOF Technical Development Committee meeting in Oslo, Norway. The first public version was published in January 2001, but soon proved to be inadequate. The 2.0.3 version was published in April 2002, and has been the current interface standard for more than ten years.

Information about previous versions of the IOF Interface Standard can be found [here](http://www.soenniksen.dk/iof-datastandard/).

# IOF Interface Standard 3.0

The information technology business evolves rapidly, and ten years is a long time in this perspective. New standards, tools and platforms become available, and usage patterns change.

Also, the sport of orienteering evolves. New event formats and inventions like geographical tracking of competitors are not supported by the 2.0.3 version. The need for a new version of the interface standard was identified in 2010. Two years later, in the end of 2012, the new version was declared official.

The formal specification of the new interface standard can be found at <http://iofdatastandard.googlecode.com/svn/trunk/IOF.xsd>. [TODO: move to <http://www.orienteering.org/datastandard/3.0/IOF.xsd>?]

# XML Schema

The specification of the new version is expressed as an [XML Schema](http://www.w3.org/XML/Schema.html), in contrast to the DTD (Document Type Definition) used previously. Defining information structure using an XML Schema is more powerful than using a DTD. Benefits include:

* Support for data types
* Support for namespaces, allowing for extensions
* Support for validation
* Code generation and validation tools available for all major programming languages and platforms

# Improvements

There are a number of improvements in the 3.0 version compared to the 2.0.3 version. Some of them, but not all, are listed here.

* Usage of standardized data types in XML Schema
  + ISO 8601 dates and times
  + total seconds instead of hh:mm for elapsed times
* Improved data structure for contact details to persons and organisations
  + email address
  + physical address
  + websites
* Support for control card lists (to keep track of rental cards)
* Support for geographical locations (i.e. latitude and longitude) for events and clubs
* More event details
  + event schedules (e g first start, banquet, price giving ceremony)
  + information
  + news
* Support for organisation hierarchy (IOF > international regional federations > national federations > national regional federations > clubs)
* Support for club logotypes
* Improved entry fee support
  + fees depending on entry time and/or competitor age
  + fees for extra competitors in a team
* Support for rogaining-style events (control scores)
* Improved support for team/relay events
  + variable number of members in a team
  + variable number of competitors per relay leg
  + a team may belong to several clubs
  + individual results per relay course and relay leg
  + team results after each relay leg
* Support for race and overall results in the same file for multi-race events
* Improved support and documentation for split times
  + how to handle failing control units
  + additional punches
* Support for Trail-O results
* Support for storage of routes acquired from geographical tracking of competitors

# Example XML file

Here is an example of a XML file that conforms to the IOF Interface Standard version 3.0. It contains the name and start time for an orienteering event.

<?xml version="1.0" encoding="UTF-8"?>

<EventList xmlns="http://www.orienteering.org/datastandard/3.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

iofVersion="3.0"

createTime="2011-07-20T12:16:31+02:00"

creator="Example Software">

<Event>

<Name>Example event</Name>

<StartTime>

<Date>2011-07-31</Date>

<Time>10:00:00+02:00</Time>

</StartTime>

</Event>

</EventList>

A few things to note:

* To avoid confusion and support multilingualism, the [UTF-8](http://www.utf-8.com/) character encoding must be used. XML files must be saved in UTF-8 encoding and also express it explicitly in the XML declaration.
* The [namespace](http://www.w3schools.com/xml/xml_namespaces.asp) http://www.orienteering.org/datastandard/3.0 is used. The namespace is preferably declared as the default namespace in the root element. Additional namespaces can be defined here as well, allowing for extensions.
* The ‘iofVersion’ attribute defines the version used, and must be set to 3.0.
* The ‘createTime’ attribute defines the time when the file was created.
* The ‘creator’ attribute defines the name of the software that created the file.

More example XML files with comments are available at <http://iofdatastandard.googlecode.com/svn/trunk/Examples>. [TODO: move to <http://www.orienteering.org/datastandard/3.0/Examples>?]

# Message definitions

There are ten types of message elements defined. Each XML file must contain exactly one message element, which always is the root element. Each message element is extendable through the Extension sub-element.

|  |  |
| --- | --- |
| CompetitorList | Used to exchange a list of possible competitors, e.g. from a national competitor database to an event administration system. |
| OrganisationList | Used to exchange a list of organisations (clubs, regional associations, etc.), e.g. from a national organisation database to an event administration system. |
| EventList | Used to exchange event fixtures/event calendars, e.g. from a national event database to a website. |
| ClassList | Used to exchange a list of classes, e.g. from a national class database to an event administration system. |
| EntryList | Used to exchange entered competitors for an event, e.g. from an online entry website to an event administration system. |
| CourseData | Used to exchange information about courses and controls for an event, e.g. from a course setting application to an event administration system. |
| StartList | Used to exchange start times for the competitors at an event, e.g. from an event administration system to a website. |
| ResultList | Used to exchange start times for the competitors at an event, e.g. from an event administration system to a website. |
| ServiceRequestList | Used to exchange services (e.g. accommodation, meals) requested by competitors at an event, e.g. from an online entry system to an event administration system. |
| ControlCardList | Used to exchange information about rental cards, e.g. from an electronic punching system vendor to an event administration system. |

Refer to <http://iofdatastandard.googlecode.com/svn/trunk/IOF.xsd> [TODO: move to <http://www.orienteering.org/datastandard/3.0/IOF.xsd>?] for detailed documentation of the message elements.

# Data definitions

There are over 50 data elements defined. Often-used elements include Id, Person, Organisation, Event, Class, Course, Service, ServiceRequest and Fee. A data element holds a number of attributes and elements. Attributes tend to define metadata for the element whereas elements tend to define concrete data, although there is no clear distinction.

Most attributes and elements are non-mandatory. This is an intentional design choice to keep the interface flexible. In some events organisers might not need to keep track of competitor ids, while they are essential for other events. The downside of this approach is that the parties taking part in the exchange need to agree upon what elements and attributes that are mandatory. The rule of thumb is that the software creating the file to exchange should include as much information as possible, even non-mandatory information if known.

Refer to <http://iofdatastandard.googlecode.com/svn/trunk/IOF.xsd> [TODO: move to <http://www.orienteering.org/datastandard/3.0/IOF.xsd>?] for detailed documentation of the data elements.

# Identifier elements

The Id element is widely used to represent a unique identity of an entity, e.g. a person, organisation, event or entry, making synchronization possible. The id should be known and common for both parties. It is strongly advised that ids are official identities issued by a national federation. Internal database identifiers should not be exposed by the Id element.

# Dates and times

Dates and times are expressed in the [ISO 8601 format](http://www.w3.org/TR/xmlschema-2/#dateTime) yyyy-mm-ddThh:mm:ssZ. This format allows inclusion of a time zone identifier. It is strongly encouraged that this feature is used, to facilitate for computer systems running in different time zones to exchange data in a reliable manner. Most programming languages have a means to determine the time zone used on the computer, and the time offset to UTC (Universal Coordinated Time).

|  |  |
| --- | --- |
| 2012-12-26T14:36:07Z | 26th December 2012, 14:36:07 UTC (‘Z’ denotes UTC) |
| 2012-12-26T15:36:07+01:00 | 26th December 2012, 15:36:07 local time, 1 hour ahead of UTC. This is the same time as 2012-12-26T14:36:07Z, but expressed in another time zone. |
| 2012-12-26T08:36:07.9-06:00 | 26th December 2012, 08:36:07.9 local time, 6 hours behind UTC. This is the same time as 2012-12-26T14:36:07.9Z, but expressed in another time zone. Fractional seconds may be used for e.g. finish times when the time resolution is a tenth of a second. |

# Embedding extended information

Information not covered by the IOF Interface Standard can be embedded using the Extensions sub-element that is present in many elements. This allows two parties to use the IOF Interface Standard to exchange e.g. custom result information along with ordinary result lists. The custom information has to be wrapped in elements defined in an XML namespace. These elements are in turn placed inside the Extensions element. For an example, refer to <http://iofdatastandard.googlecode.com/svn/trunk/Examples/ResultList4.xml>. [TODO: move to <http://www.orienteering.org/datastandard/3.0/Examples/ResultList4.xml>?]

# Automatic code generation

One of the advantages of XML Schema is that there exists wide-spread support for automatic code generation. Using an XML schema file as input, code generation tools can build object-oriented code that reflects the structure of the XML schema. This greatly reduces the time needed for creating the boilerplate code for applications.

Search for “xml schema validation” using your favorite search engine to find out more about code generation tools for your programming language.

# Validation

Software developers should always ensure that the XML files produced by their software conform to the interface standard. There are plenty of validation tools available for free on the Internet. Search for “xml schema validation” using your favorite search engine.