MeOS Online Protocol

The purpose of the MeOS Online Protocol (MOP) is to provide an open standard for online results (and also start lists). The protocol is XML based, and the design is aimed at minimizing data transfer, making (almost) real time results possible over a slow Internet connection.

MeOS implements a MOP Server and Melin Software provides a web server implementation (PHP + MySQL) under the Apache License[1] that can be used on your server as-is, or form a basis for a custom implementation.

Setting up the Web Server

The following procedure is a minimal setup of the MOP service on a web server.

File	Description	Remark
`show.php`	Shows the result list in a web browser.	
`functions.php`	Internal utility functions.	
`config.php`	MySQL and MOP configuration.	You must edit this file.
`setup.php`	Run once to setup MySQL tables.	
`update.php`	Invoked by MeOS or other MOP server. Stores results in the database.	
`Zipupdate.php`	Same as above, but supports zipped updates. Requires more PHP functionality.	

1. Copy the files in the table above to your web server, which must support PHP.
2. Edit config.php to match the configuration of your web server's MySQL connection. (Note that this is most likely not the same MySQL server used by MeOS). Also set a password for the MOP protocol. This is the password you enter in the online result service in MeOS. Unless your web connection is encrypted, this password is sent in plain text. This means that the password protection is merely a way to avoid accidental access; it is not a strong protection.
3. Run `setup.php` on the web server. This creates MySQL tables and verifies that your config.php is correct. After this step, you may remove `setup.php` from the web server.
4. Start the Results Online service in MeOS. Enter the URL to your `update.php` script, the password for the MOP protocol, which you defined above, and an id number of the competition (you will overwrite any previous competition on the web server with that id number).
5. Show results by browsing to the `show.php` script with your web browser.

[1] See http://www.apache.org/licenses/LICENSE-2.0

Protocol Details

Data transfer to the web server is based on an XML format formally defined in mop.xsd. Here follows an informal description and some examples.

The file includes the following data

- Competition properties (name, date, organizer, event homepage).
- A list of radio controls (if any). Includes id and name.
- A list of organizations/clubs (if any). Includes id and name.
- A list of classes. In addition to id and name, each class has a specification of which radio controls are used (if any) for each leg, and an order index used to sort classes.
- A list of competitors, with id and name. A competitor has a start time, a running time, a status and a link to an organization and a class. Optionally there is a list of pairs, each pair holding a link to a radio control and a running time to that control.

  If the runner is in a team (relay) or the competition is a multi-day event, there is also a total status (that is, the status after this race, including previous races – in particular the total status is OK if and only if all previous statuses are OK and this race is OK.) and an input time. If the total status is OK, the total time (at a radio or at the finish) is defined as the running time plus the input time.
- A list of teams (if any). A team has an id and a name, and a start time, a running time, a status and a link to an organization and a class. The running time and status are total status of the team and is set once the entire team has finished. If the team participates in a multi-day event, the time and status is the sum of all stages up to and including this stage.

  In addition, the team defines a list of competitors for each leg, for example a patrol is considered as one leg with two competitors on this leg, while a classic relay might have three legs with one competitor on each leg. A competitor may only be referenced once by a team, and the class definition must be consistent.

Notes

All times in the protocol is in tenths of a second. The start time is given in tenths of a second after 00:00:00 (local time) on the (first) day of the event.

The status is a number in range 0 to 99. The following statuses are defined:

Number	Meaning
0	Unknown, running? This is the default status until something is known about the competitor
1	OK. The running time is valid if and only if status is OK.
3	MP (Missing punch)
4	DNF (Did not finish)
5	DQ (Disqualified)
6	OT (Overtime)
20	DNS (Did not start)
99	NP (Not participating)

The protocol does not define any places. This is because including place gives a global dependence between runners in a class, with implies resending a lot of data as soon places changes. Calculate places as follows:

1. Sort the runners on lexicographically on (status, running time), excluding those with status unknown (meaning not yet finished) for results at the finish.
2. Give each competitor with status OK (OK or status unknown for radio controls) the place corresponding to its position in the sorted list, starting with one, but with one exception: if a runner has the same time as its predecessor, give the same place as the predecessor got.

This algorithm is used at the finish, at radio controls. For the total place in a relay or multi-day event, use total status and input time plus running time instead.

The team status and running time is normally the same as the total status and total running time of the competitor on the last leg of the team. However, if the last leg has several competitors (for example a patrol), it is not necessarily the same as the last team member in the list. The team status and running time defines the result of the team.

The MOP XML file has two global types: MOPComplete and MOPDiff. When a MOPComplete file is received, all previous data related to the event is dropped and replaced by the contents of the file. This is the only method to delete data. MOPDiff has the same format, but adds data records or modifies existing records on the remote side.

Examples

Complete competition

The following is an example of a complete (but very small) competition.

```
<MOPComplete xmlns="http://www.melin.nu/mop">
  <competition date="2012-08-12" organizer="OK Linné"
               homepage="http://www.oklinne.nu">Linnéklassikern</competition>
  <ctrl id="70">Radio</ctrl>
  <cls id="1" ord="10" radio="70">H21E</cls>
  <cls id="19" ord="190" radio="">H35</cls>
  <org id="255">Länna IF</org>
  <org id="515">OK Österåker</org>
  <cmp id="5490">
    <base org="515" cls="1" stat="1" st="370800" rt="71480">Jonas Svensson</base>
    <radio>70,27160</radio>
  </cmp>
  <cmp id="23378">
   <base org="255" cls="19" stat="1" st="401400" rt="50740">Hostas Sjögren</base>
  </cmp>
</MOPComplete>
```

From this data, we may conclude that Jonas Svensson (OK Österåker, H21E) started 10:18, 2012-08-21 (local time), reached the radio control after 45:16, and finished with status OK after 1:59:18.

Competition update

If, for some reason, Jonas is later disqualified, the corresponding file can look like:

```
<MOPDiff xmlns="http://www.melin.nu/mop">
  <cmp id="5490">
    <base org="515" cls="1" stat="5" st="370800" rt="0">Jonas Svensson</base>
  </cmp>
</MOPDiff>
```

More examples can easily be generated from MeOS. Use the Online Results service and choose to save all output files to a folder.

Server response

When transferred to a web server, the XML file is sent as raw http post data. The http header has the following variables defined:

Name of variable	Contents
competition	User defined competition id number.
pwd	User defined password.

These variables can be used to identify which for competition data is sent, if the server supports several competitions. The password can be used to verify authorized upload, but unless the server supports encryption, the level of actual security is minimal.

When the remote server has processed the request, it should respond with an XML file, reporting the result of the operation:

```
<?xml version="1.0"?><MOPStatus status="X"></MOPStatus>
```

Defined values of the status code X are:

Status code	Meaning
OK	Request processed successfully.
BADCMP	The specified competition is unknown
NOZIP	Zipped files not supported by this server
BADPWD	The password was not correct
ERROR	General server error