ADVANCED USER'S GUIDE.

race | result 11

race | result

Precision. Passion. German Engineering.

www.raceresult.com

This manual is intended for

users that would like to understand the structure

and advanced solutions of the race result 11 software

in order to be able to create better solutions for special kinds of events.

This manual is not needed when using race result 11 for standard events.

TABLE OF CONTENTS

1. INTRODUCTION

Managing and scoring a complex race is based on three pillars:

- Managing and calculating the race data. The race scoring software needs to make a lot of calculations, like calculating interim times or lap times, calculating ranks, calculating team results, etc.
- Providing this data to different systems and devices
- Presenting this data in an appropriate from, e.g. as a result list, as a race certificate, as text message to the athlete's cell phone, etc.

In chapter 2 you will get to know the structure of the system which will help you understand how data will be exchanged between different devices and components.

Chapters 3 and 4 are very important and will explain the race result data structure and teach you how to calculate the information you need.

Chapter 5 will then introduce different options to present the data in the appropriate format.

Chapter 6 will be a helpful practical guide how to set up your event with a lot of tips and tricks that make your work easier.

If chapters 3 and 4 were not enough for you, chapter 7 will introduce additional data fields and functions not covered before, and chapter 8 will provide answers to frequently asked advanced questions.

2. SYSTEM STRUCTURE

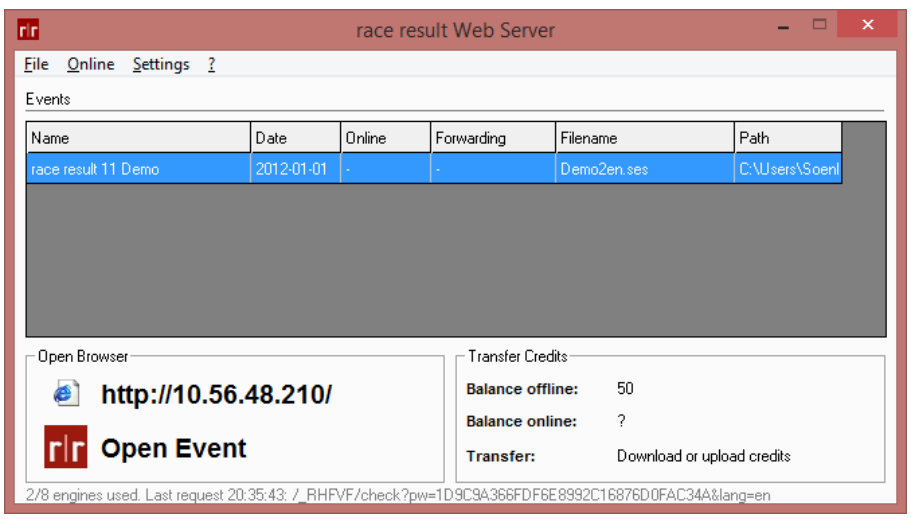
2.1. Working Online and Offline

race result 11 has a server/client based structure that runs both online and offline. Working online means that your race data is stored in the Internet on a race result server and can be accessed from any place with an Internet connection. When working offline, your race data is stored locally on your computer and can be accessed only on that computer and other computers in the local network. Usually, you would work online before and after the event which allows easy online collaboration with other members of the team and also using the results and registration platform my.raceresult.com. During the event, however, it is recommended to work offline so that you still have access to your data even if you lose Internet connectivity.

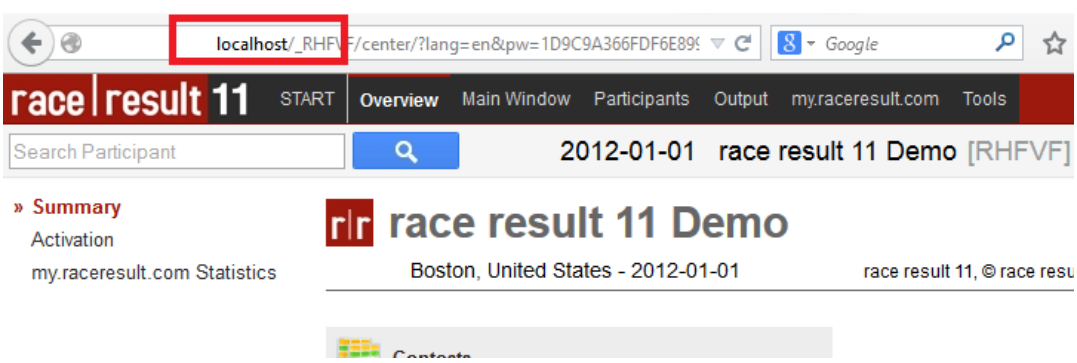
2.2. Browser and Windows Based

Most parts of race result 11 are running in the Internet browser. You would simply log in on www.raceresult.com and access your events. Some parts, however, are Windows based programs, for example the Transponder Module which creates the connection between your event file and your timing system.

When working offline, the small program *race result Web Server* runs as your local web server:

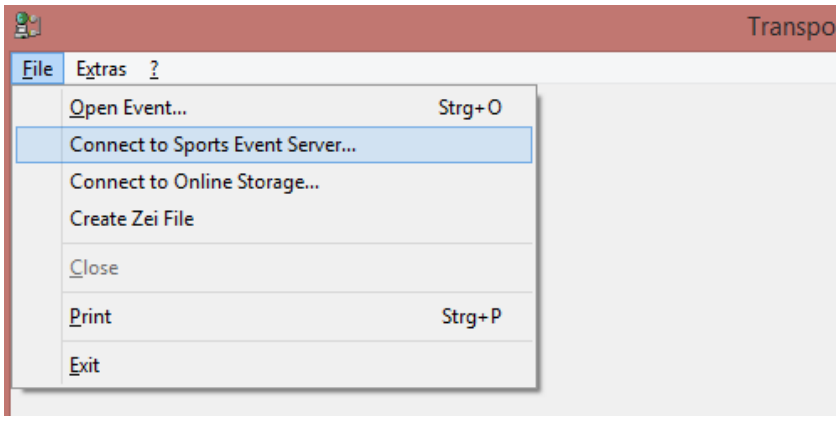


You still access the data through the browser via a http://localhost/.. address:

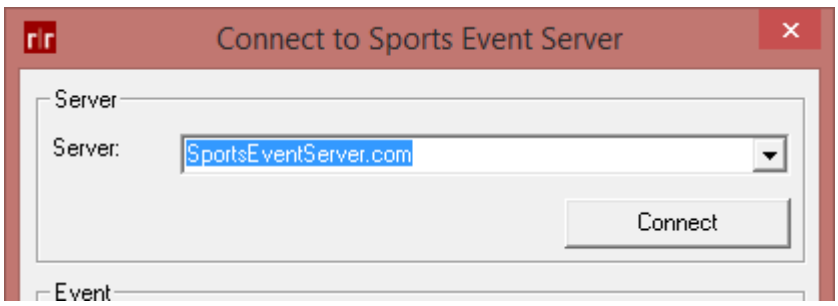


Also all other computers in your local network can access the event by entering the IP address of local server in the browser, e.g. http://192.168.1.100

The Windows based program can work with event files stored both offline and online. In order to open a file stored online, select *File -> Connect to Sports Event Server*.



Then, make sure SportEventServer.com is entered as server and click the *Connect* button.



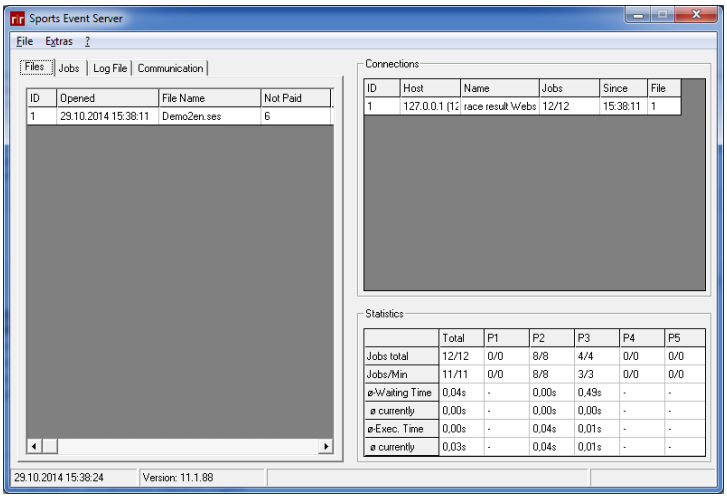
Then enter your customer id and password and choose your event.

In order to open an offline file, click *File -> Open Event* as usual and select the file on your computer.

You can also open a file in your local network (if the computer is not the local server) by using *File -> Connect to Sports Event Server* and enter the IP address of the local server.

2.3. Sports Event Server

The SportsEventServer is the program in the background doing all the data processing. You can access it through the icon in the tray bar:



In detail, the SportsEventServer (SES) does the following:

- As the local server, the SES manages TCP/IP connections to the various programs on various computers in your local network.
- It accepts job request from these programs. A job can either be a write job (change participant data, add a time, change a setting, etc.) or a read job (retrieving settings, performing a data query, etc.). The server prioritizes the job request and processes them.
- At all times, the server ensures data consistency: it always knows which information needs to be calculated or updated.

2.4. Online Forwarding

Using the Online Forwarding is an easy way to provide live results on my.raceresult.com when working offline. If you have an (even unreliable) Internet connection, the SES will forward all write jobs to the SES in the Internet, so that both servers (the local server and the server in the Internet) will do the same and thus have the same data status.

If your Internet connection breaks down, the online forwarding stops, but you still have access to all your data and can simply restart the online forwarding when your connection recovered.

2.5. my.raceresult.com

my.raceresult.com is an Internet platform to provide online registration and results to your participants. It is connected directly to your event file, i.e. whenever you change something in the file, the changes will be reflected on my.raceresult.com (almost) immediately. You only have to decide which information you would like to publish on the platform and you will not have to update the information ever.

3. DATA STRUCTURE AND DATA PROCESSING

Now we are starting with the advanced parts. Remember that race result 11 separates the calculation of data and the presentation of data. This and the next chapter only explain how to set up your event so that the system makes exactly the calculations you need. The presentation layer will be explained later in chapter 5.

So, in order to understand race result 11, you have to understand first how it handles data. Two aspects of handling data have to be separated: making necessary data available (calculation) and retrieving data (filtering, selecting fields, sorting). Assume that you would like to create a list of the three riders that have been fastest in the second lap. First you will set up your event so that it calculates the rank in the second lap. From then on, this information (rank in second lap) is available for all participants. You can use this information like any other field (e.g. year of birth), but the system will make sure that it is always up-to-date even if the underlying results may change. This field can then be used to retrieve the required data, for example:

- Fields: rank_in_2nd_lap, bib, name, time_in_2nd_lap;
- Filter: [rank_in_2nd_lap]<=3
- Records ordered by rank_in_2nd_lap

This also means that the data retrieval is based on the participants, but the participants' data can have different sources:

- Stored data fields e.g. bib number, last name, date of birth
- Assigned fields e.g. name or length of the assigned contest, name of the event, date of the event, age group name
- Derived fields e.g. age (derived from date of birth and event date)
- Stored times e.g. time stamp when passing the 10K split timing point
- Calculated times e.g. time needed from half marathon split to marathon finish.
- Formatted times the time stamp 5 minutes and 3 seconds can be presented in many formats, e.g. 05:03, 00:05:30.0, etc.
- Calculated ranks e.g. rank in second lap, or: finish time rank of all athletes having the same first name (hypothetically)
- Calculated team results, e.g. team rank, team time

If you know data bases, you can think of a query over the table *participants* joined with other tables (event data, contest data, age groups) and some other magic that makes sure that you also have immediate access to all calculated values such as ranks.

Whenever you set up an event, you need to think about, first, which information you would like to store besides the standards fields (e.g. participation in city championship, t-shirt size, etc.) and which information you would like to calculate: which times/points/values are needed? Which ranks are needed for my reports and certificates? Which team results are needed? Start to set up the scorings settings in Main Window -> Advanced Settings accordingly (this chapter) and then set up the reports you need (chapter 5).

In the following, the data sources listed above will be described in detail.

3.1. Stored Data Fields

The stored data fields are pretty straight forward. You enter something and when using the field it will show what you have entered. When creating reports, you can easily select the desired data field in the field wizard:

Besides the standard fields like bib, name, club, etc., the 20 additional text fields (ATF), the 20 additional yes/no fields (AYN) and the 20 additional yes/no text fields (AYNTF) serve to save any kind of additional information. Note that you can access these additional fields either by its name (e.g. ATF5) or by the label you defined (e.g. TShirt). The second option is nice, because you do not have to remember the number of the field, and even if you change the numbering, you do not have to update your reports. In return, you cannot change the label without changing your report.

Note that there are also stored fields that cannot be written in externally, e.g. *Created* and *Modified* (when the record has been created / modified).

3.2. Assigned Fields

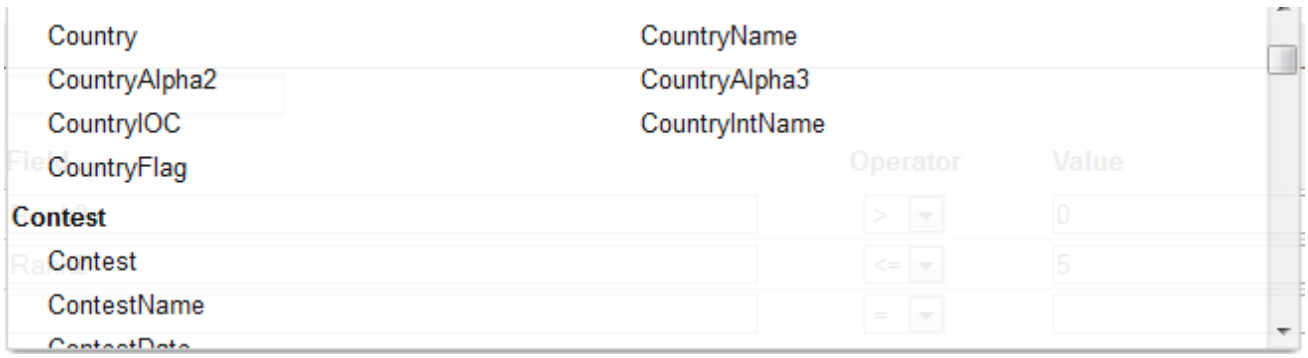
Depending on the contest / age group / nation / country assigned to a participant, you also have access to the other attributes of the selected contest / age group / nation.

You can access the attributes of the assigned contest / age group / country also simply by selecting them in the field wizard:

It works the same with the attributes of the event (of course it is the same event for every participant):

A simple use case for that is a race certificate: instead of putting the event date on the certificate as fixed text, better use the field EventDate so that you do not need to update the certificate next year.

Maybe you are wondering what attributes the assigned country / nation has: these are different representations of the name of the country. If you select Germany as country for a participant, you can show it on a list / certificate as Germany, GER, DE, or Deutschland.



3.3. Derived Fields

Derived fields can be used like any other data field, but they are derived from other fields of the participant, e.g.

Age: Derived from date for birth and event date

Year: year of birth (e.g. 1983), derived from date of birth

Year2: two digit year of birth (e.g. 83), derived from date of birth

LastFirstName: Combines last and first name to (e.g.) Doe, John

3.4. Stored Times

For each participant, an arbitrary number of times can be saved, but you need to define these *Results* first in Main Window -> Results. For example, you define that each participant can have the stored times *Start, 10k split, half marathon split, 30K split, 40K split, finish* by setting up these results accordingly.

Note that a time is no more than a decimal number that usually represents the time in seconds. So a time of 30 minutes would be stored as the value 1800 (seconds). Not necessarily, the value stored has to be a time, it can also present points or any other value that is a number.

To access these stored values, you can use the field *DecimalTimeX* (replace X, e.g. DecimalTime5 for the value in the result with ID 5). There also is a short version *Tx*, e.g. T5.

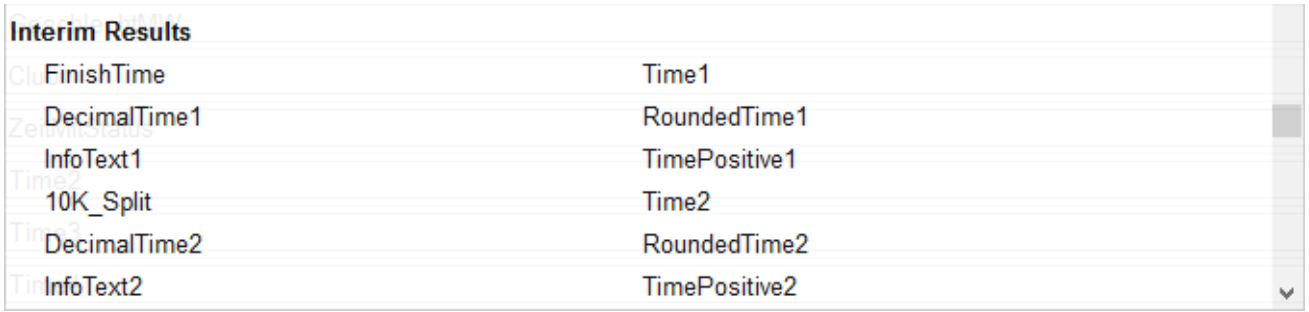
Usually, this short version would be used to calculate formula results (see below), for example using a formula like T5-T4, whereas on reports and certificates you would use the formatted times (0:30:00 instead of 1800).

3.5. Rounded and Formatted Times

The rounded and formatted times are derived from the stored (decimal) times. The rounded times can be accessed via the fields RoundedTimeX or short TRx (replace X, e.g. RoundedTime5 or TR5). If you are showing rounded times on reports, it is recommended to make calculations using rounded times as well, otherwise rounding discrepancies may occur.

The rounded times are also the basis for the formatted times. The system uses the rounded value to build a time string like 1:23:45.67. This formatted time can be accessed using the fields TimeX (e.g. Time5).

All fields can simply be selected in the field wizard:



In this list you can see that you can also access the formatted time via the name, e.g. *10K_Split*. It has the same advantages and disadvantages like in case of the additional text fields: when you change the ID, you do not have to change your reports, but when you change the name, you have to.

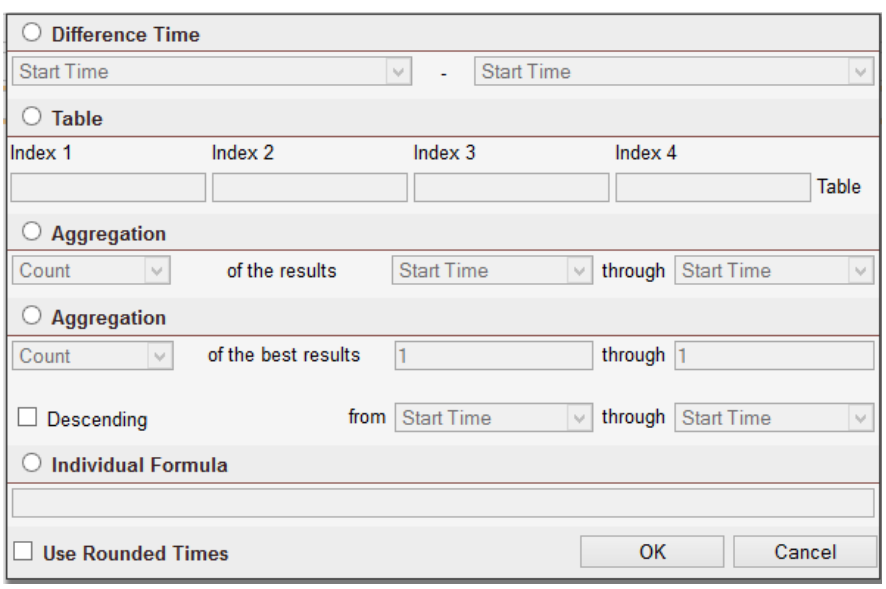
The screenshot also shows that each result has two more fields: *InfoTextX* is a text field that usually contains additional information from the timing system (e.g. hits, signal strength, etc.), and TimePositiveX is a derived field which is 1 if the time value is greater zero, and 0 otherwise.

Also note that for rounding and formatting times, the settings from Main Window -> Results will be used. If a rounding / a time format is not specified there, the rounding / time format from the assigned contest will be used. This way, you can define a default time format for each contest, and for some results you will define a different time format, e.g. "*s*" for points.

3.6. Calculated Formula Results

When defining a formula for a result in Main Window -> Results, no values can be saved in this result, but instead the SportsEventServer will automatically calculate the values according to your formula. The formula can be any formula that returns a number (since results can only contain numbers).

The formula assistant (click on the hand symbol) helps you to set up easy formulas and to learn the syntax:



The easiest kind of formulas are difference times. Example: when you save the time stamps after lap 1, 2, 3.. in results 11, 12, 13, .. you can set up results with formulas T11-T0, T12-T11, T13-T12, … to calculate the time of lap 1, 2, 3, …

Other formulas can build the sum/average/minimum/maximum over several results or look up a value in a table. Finally, you can also build completely individual formulas using the expressions and functions described in the next chapter. Very often the function *iif()* will be used. For example the formula

```
iif(T21>0; T21; T22)
```

will return the value of result 21 if it is greater zero, and the value of result 22 otherwise.

Many formulas use only times of the athlete, however, you can also access the participant's data, e.g.

```
iif([Sex]= "f"; T21; T22+10)
```

If the participant is female, the formula will return the value of result 21. Otherwise it will return the value of T22 and add 10 seconds (see next chapter to learn what the formula returns for men who do not have a time in result 22).

The formula can also access the ranks and team scores. Assume you would like to calculate points like this: 1st place: 10 points, 2nd place: 9 points, … 10th place: 1 point; others: 0 points. For this you can use this simple formula:

```
max(11-[Rank1]; 0)
```

The formulas can be nested and thus get as complicated as necessary. You can imagine that you can calculate virtually anything, similar like in Excel.

3.7. Finish Result

Another important concept regarding results is the *Finish Result*. The finish result is an attribute of a contest and can be defined in Main Window -> Contests. Selecting one result as finish result has the following effects:

- The field *Finished* will be 1 if the participant has a positive time in the finish result, i.e. the field corresponds to the field *TimePositiveX* where X is the ID of the finish result.
- The fields *DecimalTime*, *RoundedTime* and *Time* (without a number at the end) will return the (rounded) decimal time value and formatted time of the selected finish result.

Note that the finish result can refer to different results for participants in different contests. This will turn out to be very helpful later.

By default, also the calculation of ranks is based on the finish result:

3.8. Ranks

After defining for example the rank with ID 4, three additional fields can be used:

- Rank4 the position of the participant according to the rank definition, e.g. 47
- Rank4p the position with a period at the end, e.g. 47.
- MaxRank4 the number of participants in the same group, e.g. 495.

MaxRank helps to create an output like: *47 of 495 in age group M50*.

Note that defining a rank only provides these additional fields. Creating a report with this rank is a separate step and will be explained in chapter 5.

Ranks can be defined entirely flexibly. Theoretically, you can define a rank that

- only considers athletes named John,
- groups them into those who are born in an even year and those who are born in an odd year,
- and sorts and positions them by the rounded time in lap 2.
- If the time in lap 2 is not decisive, the participants shall be positioned by the length of their last name.

Yes, this is a very hypothetical case, but it shows that there are virtually no limits in what you can do.

The ranks will be defined in Main Window -> Ranks and their settings are based on three main pillars: filter, grouping, sorting.

Filter: In most cases, you would like to exclude certain participants from the rank, for example those that have not finished. These will not receive a rank (more precisely: the rank field will contain the value -1). For the standard ranks, the filter is therefore simply *Finished* (based on the finish result, see above), but the filter can be any other expression, e.g.

- [FirstName]= "John" theoretical example from above
- [Finished] AND [AYN1] participant needs to have finished and the AYN field needs to be set
- T5 > 3 the value in result 5 needs to be greater than 5

Grouping: the fields selected here group those participants fulfilling the filter into several groups. Two participants will only be in the same group if they have identical values in all fields. For example, a typical gender rank would be grouped by the fields *Contest* and *Sex*, so that there are groups like 5K male, 5K female, 10K male, 10K female. For each of these groups there will be a 1st, 2nd, 3rd, … place.

Sorting: the sorting can be defined either in the rank definition (turn off option *Order According to Contest*) or in the contest settings (turn on the before mentioned option). The selected order fields define how the participants shall be ordered and then ranked. If the first field is not decisive, the second field will be considered, and so on. For each field, you can also define if the participants shall be ordered ascending or descending. For example for lap races you will need descending: more laps -> better position.

Finally, there is an option *Ties Possible* which defines what happens if all sorting fields are not decisive: if the option is turned off, the participants will be ranked randomly. If the option is on, two participants will have the same rank and then there will be a gap, e.g. 1st, 2nd, 3rd, 3rd, 5th, ..

The name of the rank is on the one hand helpful to remember what a rank is intended for, on the other hand the name can also be used to access the field (same as additional text fields and results). If you name a rank *OverallRank,* you can also access the field *OverallRankp* which adds a period to the number (note that the fields with period return a text whereas the fields without period return a number. In consequence, the fields with period cannot be used to sort a list).

3.9. Team Scores

Whenever several participants build a team which will be compared with other teams, you need a team score. A team score will determine which participants belong to the same team, calculate team results (e.g. sum of the participants' finish times) and rank the teams among other teams. Again, the team score will only provide additional data fields (such as team rank, team time, number of team members, etc.) which can be used to create appropriate lists, race certificates, etc.

The team scores can be defined in Main Window -> Team Scores. Note that you can create several team scores if needed.

When setting up the team score proceed in the following steps:

1. Think about who builds a team. Which fields do team members have in common? In many cases this will be *Contest* and *Club* – the team members have to have run the same contest and need to be in the same club. You can also add the field *Sex* if teams can have either only women or only men.
   Enter these fields in the *Team Aggregation* settings.
2. Define the minimum and maximum number of participants per team and enter these in the *Number of Participants* settings. In many cases, the teams will have a fixed size (e.g. 3 participants per team), so make sure to enter the same value for minimum and maximum.
   If a team can have any number of women, make sure to enter 0 for *Min. Number of Women*.
3. Use the *Filter* setting to define who will be considered for the team score. The filter may be
   `[Club]<> ""`
   so that participants without a club name will be ignored.
4. What happens if the number of participants of one club is larger than the maximum team size? Depending on the option *Max. No. of Teams* the team score will create several teams for a club or it

will associate the other participants of the same club with the team, but they won't be scored. Think of a team score for school classes: all children shall be part of the team, but only the best 5 shall be scored (i.e. their finish times will be summed up for the team time).

5. Think of how the teams will be grouped before the ranking. Enter the appropriate fields in the *Grouping* settings. Usually, this is a subset of the *Team Aggregation* fields, e.g. only *Contest*: there will be a 1st, 2nd, 3rd, .. team for the 5K race and another 1st, 2nd, 3rd, .. team for the 10K race.
   If you added Sex to the *Team Aggregation* fields you probably want to add it to the Grouping as well. Otherwise the male and female teams will be ranked together, for example on the three first ranks there may be male teams and a female team may have the 4th rank.
6. Using the settings in the *Results* section you can define which times will be used to build the team times. Simply, select one or several results and determine if the team time will be the sum, average, minimum, maximum or number of times.
   Note, these results also determine how the teams will be ranked. If you select the finish time as *Result 1* and sum these up, the teams will be ranked by the sum of this time, either ascending or descending. If result 1 is not decisive, result 2 and then result 3 will be considered. If all results are not decisive, two teams can have the same rank if the option *Ties possible* is activated.

Among others, the team score provides the following new fields:

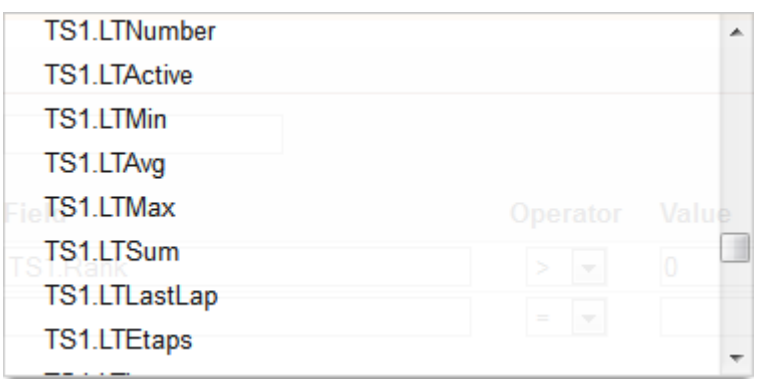
TSx.Rank	rank of the team
TSx.MaxRank	highest rank given within the group. Can be used two show "rank 5 of 23".
TSx.Time1	formatted time of the team according to the *Result 1* setting
TSx.DecimalTime1	decimal time of the team according to the *Result 1* setting
TSx.Position	All team members will be numbered from 1 to [team size]. This field returns that number. The order by which the team members are being numbered depends on the *Order Participants* setting.
TSx.Number	number of team members of that team.
TSx.TeamIndex	If several teams can be built from one club, the first one will have team index 1, the second team index 2, and so on.

3.10. Team Scores with Lap Mode

The Lap Mode is a special mode of a team score designed for team lap races like 24h races. There is only one timing point per lap and only one rider will be on the track at a time. Thus, a lap time may be the time stamp of one athlete minus the last timestamp from the previous athlete.

When activating the Lap Mode in Main Window -> Team Scores, the team score will take all times from the team, sort them and calculate the lap times. The team will be scored first by the number of laps and second by the time needed.

The team score will then provide a lot of additional fields:



For further information, refer to the race result Online Help and the template "Team Lap Race".

3.11. Background Information: Real Time Scoring Engine

On the previous pages you have read that race result 11 can be used to make a lot of complex calculations that are almost as flexible as Excel. For a race, it is not only important that you can make any kind of scoring, but also that these calculations will be performed very fast with virtually no delay. During the race you may need instant real time information without any delay in order to be able to show it on a live screen or even in TV graphics.

For this reason, the SportsEventServer in the background performs almost every calculation in real time, i.e. whenever some data is changed (e.g. a new lap time added), the server will recalculate and update all depending data (e.g. number of laps, fastest lap, slowest lap, rank of this rider and other riders). Therefore, the server analyzes the dependencies in the event file and thus knows which information may need to be updated if some data is being changed. The dependency tree may look like this:

```
RESULT 1: Finish
|__RESULT 13: AfterRunning
   |__RESULT 0, 23

RESULT 2: Swimming
|__RESULT 11: AfterSwimming
   |__RESULT 0, 21

RESULT 3: Bike
|__RESULT 11: AfterSwimming
|  |__RESULT 0, 21
|__RESULT 12: AfterBike
   |__RESULT 0, 22

RESULT 4: Running
|__RESULT 12: AfterBike
|  |__RESULT 0, 22
|__RESULT 13: AfterRunning
   |__RESULT 0, 23
```

```
RESULT 11: AfterSwimming
|__RESULT 0, 21

RESULT 12: AfterBike
|__RESULT 0, 22

RESULT 13: AfterRunning
|__RESULT 0, 23

RANK 1: TotalRank
|__RESULT 1: Finish
   |__RESULT 13: AfterRunning
      |__RESULT 0, 23

RANK 2: MFRank
|__RESULT 1: Finish
   |__RESULT 13: AfterRunning
      |__RESULT 0, 23

RANK 3: AGRank
|__RESULT 1: Finish
   |__RESULT 13: AfterRunning
      |__RESULT 0, 23
```

In this example, the system understands that the *Total Rank* depends on the result *Finish,* which itself depends on the result *AfterRunning* and so on.

This example is fairly simply. If you think of an event like the Tour de France, you can imagine that an event file can have hundredths of results and dozens of ranks with long dependency trees. There are results for each day, special sprint scores, overall scores, and so on.

Real Time Limits

Note that not everything can be calculated in real time:

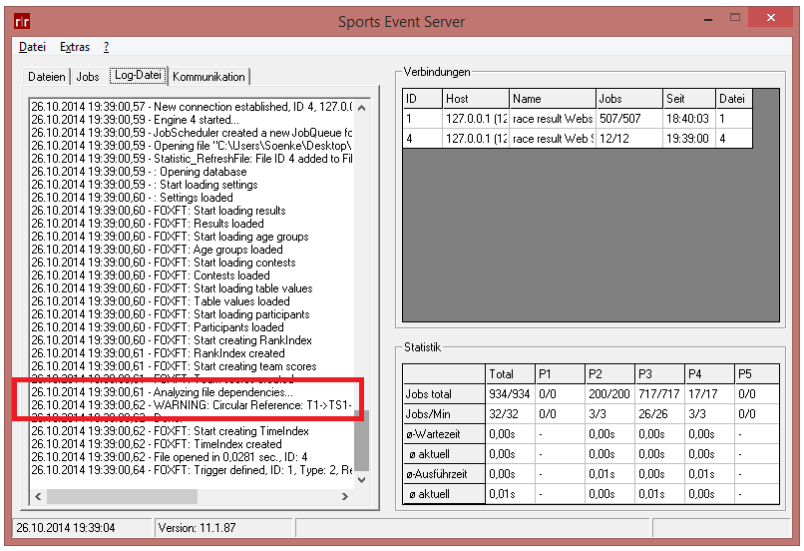
- Team scores will be updated in real time only if the real time option is activated in the settings
- A formula result will not be calculated in real time if it depends on a rank or a team score.
- Everything that depends on something that is not being calculated in real time cannot be calculated in real time either.

If something is not being calculated in real time, it will simply be calculated when needed: if you open the result list based on a non-real time team score, the system will calculate the team score first before creating the list what will create an additional delay of a few milliseconds or seconds depending on the size of the event.

Circular References

When you set up a complex race you can quickly make a tiny mistake which creates a circular references. For example the definitions *A:=B+C+1* and *B:=A+1* create a circular reference. If C has been updated, A needs to be updated. In return, B needs to be updated and then A needs to be update again. This will create an endless loop so that the system would never stop working unless it detects the circular reference.

The SportsEventServer detects even long complex circular references and then refuses the calculation since something must be wrong in your definition. However, it shows the detection of the circular reference only in the log file:



If something is not working as expected, checking for a circular reference may be helpful.

4. EXPRESSIONS AND FUNCTIONS

4.1. Expressions and Functions

In the last chapter we have already used expressions (called them formulas) without actually understanding what they are and how they work. Expressions use some operators and functions on some data to generate some new data. For example the expression

```
max(T1; T2) + 1
```

uses the data fields T1 and T2 to generate a new information: the maximum of those two added to 1.

Operators and functions are actually the same, but operators always have two parameters and their syntax is different. The expression

```
T1+T2
```

uses the operator "+", but theoretically it could also be a function like

```
add(T1;T2)
```
[Note that the *add* function does not exist in race result!]

In case of operators, the parameters (called operands) will be left and right of the operator. In case of functions, the parameters will be in round brackets after the function name separated by a semicolon.

You will find a complete list of all operators and functions in the race result Online Help. Here, the most important ones shall be discussed:

4.2. Operators

=	equal (returns true or false)
>	greater than (returns true or false)
<	less than (returns true or false)
<=	less or equal (returns true or false)
>=	greater or equal (returns true or false)
<>	not equal (returns true or false)
AND	logical And (returns true or false)
OR	logical OR (returns true or false)
XOR	logical XOR (returns true or false)
+	addition
-	subtraction
*	multiplication
/	division
%	modulo
:	time operator. The operator will multiply the left operand with 60 and add the result to the right operand, i.e. *a:b* is equivalent to 60*a+b. This special operator makes it possible to write times intuitively, e.g.

```
iif(T1>1:29:00; "more than 1 hour 29 minutes"; "less than")
```

& concat strings

The last operator & is very important for lists and certificates. Imagine you would like to show last name, first name and nation in brackets on a list. Then, you can use this expression:

```
[LastName] & ", " & [FirstName] & " (" & [NationIOC] & ")"
```

In this example you can also see that fields need to be in square brackets when using them in expressions, whereas fixed texts will be in double quotes. The only fields that do not have to be in brackets are the decimal and rounded fields T1, T2, T3, .. and TR1, TR2, TR3, … This way it is easy and intuitive to create formula results like T22-T21

Note that an operator expression like T12-T11 will only be calculated if both operands (T11 and T12) are available. If T12 is NULL (not entered / does not exist), the result of T12-T11 is NULL as well. In some cases, however, you may want to make the calculation: assume result 32 can contain a bonus/malus which shall be added if entered: T12-T11+T32. If T11 and T12 are available, the formula shall be evaluated, no matter if T32 is available or not. For that you can use the nz() function which returns 0 if its parameter is NULL (not available):

```
T12-T11+nz(T32)
```

4.3. Functions

These are the most important functions. Please find a complete list of all available functions in the race result Online Help.

nz(a)	returns 0 if a is NULL, and a otherwise
max(a;b)	returns the maximum of a and b
min(a;b)	returns the minimum of a and b
ucase(a)	converts all characters of a to upper case
lcase(a)	converts all characters of a to lower case
left(a; b)	returns the first *b* characters or a
right(a; b)	returns the last *b* characters or a
iif(a; b; c)	if a is true, b will be returned, c otherwise
switch(a;b;c;d;..)	if a is true, b will be returned; otherwise: if c is true, d will be returned; and so on.
choose(n; a;b;c;d;..)	will return the *n*th parameter (a if n=1, b if n=2, and so on)
format(n; s)	formats the time of n seconds according to the time format s, e.g. format(T5; "mm:ss.k")

4.4. User Defined Fields

User defined fields can make your life a lot easier. They are very similar to the derived fields like FirstLastName (see above), but they are not defined by the system, but instead by you. They can be defined in Main Window -> User Defined Fields/Fct.

When you need one expression in many different parts of your file, you should define it as user defined field. For example, when creating a new event file, a user defined field *DisplayName* will be created using this expression:

```
trim([Title] & " " & [Firstname] & " " & [Lastname])
```

It is the default way to show a name on a result list: on all result lists the name column refers to the user defined field *DisplayName*. The big advantage is that you can change this user defined field and in consequence all the result lists will show the name in the new format. You do not need to update every single result list.

4.5. User Defined Functions

User Defined Functions are very advanced. You can define not only your own derived fields, but also your own functions in Main Window -> User Defined Fields/Fct. Assume you would like to change the upper/lower case of words, so that the first letter is in upper case and the other letters in lower case. For this you could define this function:

```
RightCase(x)      ucase(left([x];1)) & lcase(mid([x]; 2))
```

You can use the new function like any other function now, e.g.

```
RightCase("raCEreSuLt")       will return: Raceresult
```

Your functions can also have several parameters. For example we can define the *add* function:

```
add(x;y)     [x]+[y]
```

The names of the parameters can be defined freely. When using them in the definition of the function, they have to be in square brackets.

5. PRESENTATION OPTIONS

By now you should have an idea how to generate most of the data you need for scoring your event. In chapter 7 we will cover some advanced fields and functions, but for now this should be sufficient. Having the necessary data, it is now about bringing it into the right form using lists and certificates.

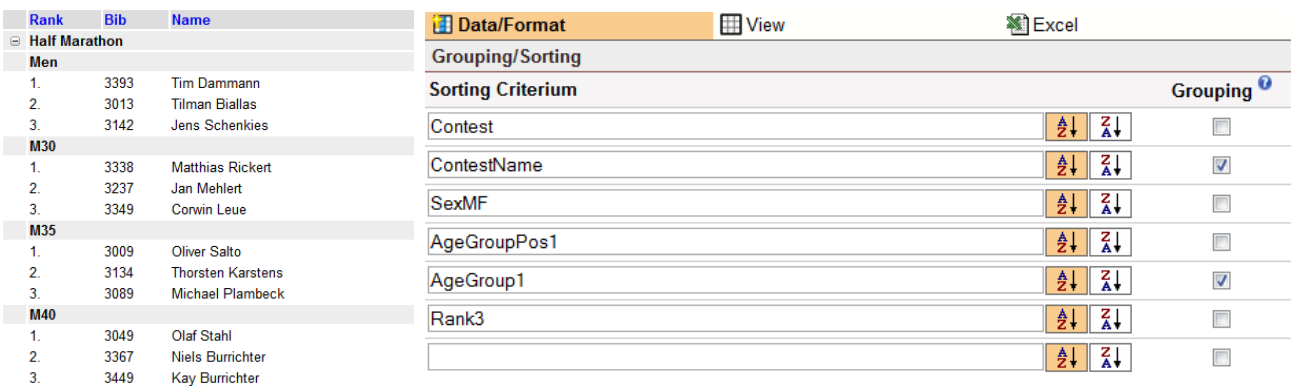
We assume that you have already familiarized yourself with lists and certificates. You know that a list serves to put many records on one page, whereas a certificate will have one record per page with free placement of data, pictures and other elements. You have explored the settings of lists and saw that you define these through sorting fields (which can also create group headers), data fields (which define what information /columns to show on the list) and filters (which define which records to show on the list). You have also checked out the *Designer* which serves to place and format the elements on your race certificate.

In this advanced users guide we are now going to cover advanced topics for lists and certificates.

5.1. Lists

List Sorting / Grouping

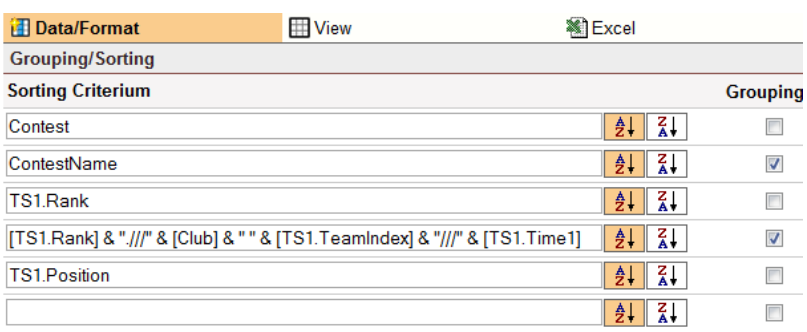
The settings in the section *Grouping/Sorting* define how the records on the list will be sorted. If the first field (in the first line of the settings) is not decisive, the second sorting field will be considered, and so on. When turning on the option *Grouping* a group header will be inserted every time the value of the field changes.



Rank	Bib	Name
Half Marathon		
Men		
1.	3393	Tim Dammann
2.	3013	Tilman Biallas
3.	3142	Jens Schenkies
M30		
1.	3338	Matthias Rickert
2.	3237	Jan Mehlert
3.	3349	Corwin Leue
M35		
1.	3009	Oliver Salto
2.	3134	Thorsten Karstens
3.	3089	Michael Plambeck
M40		
1.	3049	Olaf Stahl
2.	3367	Niels Burrichter
3.	3449	Kay Burrichter

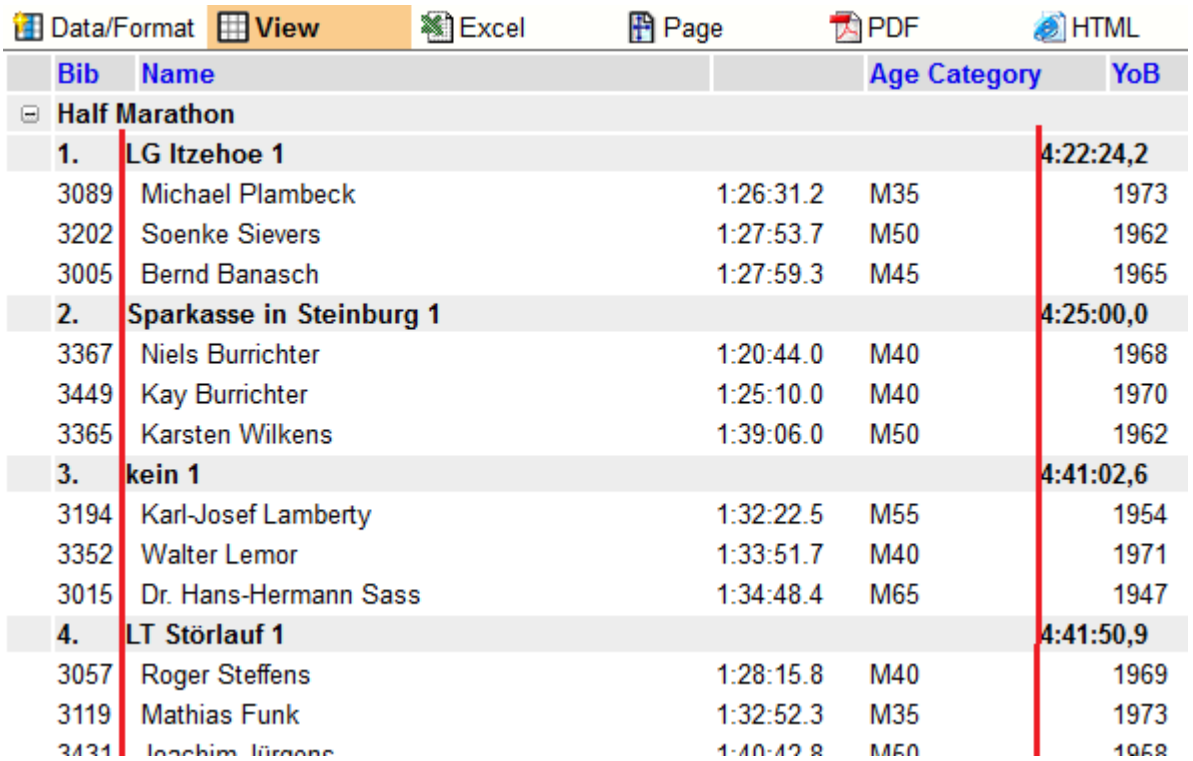
In this example, there are two groupings: the first for the contest and the second for the age group. Also note that the list is sorted first by the field *Contest* (the ID of the contest) and then grouped by the *ContestName*. This way the contests will be ordered by the contest ID (instead of by the contest name), but the contest name will be in the group header.

Using different fields/expressions for the sorting and grouping is very commonly used. For example consider the standard team results:



The group header will show several data points (team rank, club name, team time). As a result the expression in the 5th line cannot be used for sorting (as a text it would return this order: 1st, 10th, 11th, …, 2nd, 21st, 22nd, ..), so that the field *TS1.Rank* (team rank) will be used first for sorting.

Maybe you are wondering what the three slashes (///) in the group header are used for. This is a small hack which creates columns within the group headers:



	Bib	Name		Age Category	YoB
	Half Marathon				
	1.	**LG Itzehoe 1**			**4:22:24,2**
	3089	Michael Plambeck	1:26:31.2	M35	1973
	3202	Soenke Sievers	1:27:53.7	M50	1962
	3005	Bernd Banasch	1:27:59.3	M45	1965
	2.	**Sparkasse in Steinburg 1**			**4:25:00,0**
	3367	Niels Burrichter	1:20:44.0	M40	1968
	3449	Kay Burrichter	1:25:10.0	M40	1970
	3365	Karsten Wilkens	1:39:06.0	M50	1962
	3.	**kein 1**			**4:41:02,6**
	3194	Karl-Josef Lamberty	1:32:22.5	M55	1954
	3352	Walter Lemor	1:33:51.7	M40	1971
	3015	Dr. Hans-Hermann Sass	1:34:48.4	M65	1947
	4.	**LT Störlauf 1**			**4:41:50,9**
	3057	Roger Steffens	1:28:15.8	M40	1969
	3119	Mathias Funk	1:32:52.3	M35	1973
	3431	Joachim Jürgens	1:40:42.8	M50	1958

Note that the *Page Break* option offers a lot of option to insert page breaks (for PDF output) before a group header or to repeat the group header on a new page.

Also, in the advanced settings (accessible via the double arrow >>) you can define colors for the group header:

Half Marathon

1.	**LG Itzehoe 1**		**4:22:24,2**
3089	Michael Plambeck	1:26:31.2	M35
3202	Soenke Sievers	1:27:53.7	M50
3005	Bernd Banasch	1:27:59.3	M45
2.	**Sparkasse in Steinburg 1**		**4:25:00,0**
3367	Niels Burrichter	1:20:44.0	M40
3449	Kay Burrichter	1:25:10.0	M40
3365	Karsten Wilkens	1:39:06.0	M50

The advanced settings also offer dynamic formatting, explained below.

Multi-Line Lists

Multi-line lists are an easy way to put more information on a list. One record (participant) can have up to 30 lines on the list. Therefore, simply select a line greater 1 for a field/column of your list.

Rang	Stnr.	Jhg.	Name	Ort		
Elite / Amateure						
1.	3	1984	FLÜCKIGER Lukas	Leimiswil		
Rd.Zeit:			*5:47 (9.)*	*5:50 (10.)*	*5:54 (8.)*	*5:48 (3.)*
Total:	0:46 (2.)		6:34 (6.)	12:25 (9.)	18:19 (9.)	24:07 (8.)
2.	2	1985	WILDHABER Marcel	Galgenen		
Rd.Zeit:			*5:47 (7.)*	*5:50 (8.)*	*5:54 (2.)*	*5:48 (2.)*
Total:	0:46 (3.)		6:34 (5.)	12:24 (8.)	18:18 (5.)	24:07 (5.)
3.	16	1983	ZAHNER Simon	Dürnten		
Rd.Zeit:			*5:47 (5.)*	*5:50 (6.)*	*5:54 (9.)*	*5:48 (7.)*
Total:	0:46 (1.)		6:33 (1.)	12:24 (5.)	18:19 (7.)	24:07 (7.)

Maybe you would like to have different column widths for the fields in the different lines like in the screenshot above: the lap times in the second and third line have different column positions than the basic information above. To do so, you have two options:

- You can enter the exact column position in the extended field settings (click on the double arrow >>).
- You can set a column offset. All lines will have the same columns, but for example in the second line the field will be moved one or several columns to the right, i.e. the previous column has a greater column span.

By the way, note that the system automatically determines the column widths in your list depending on the content (unless you set specific column positions). Therefore, the system first determines the needed width for each column. If the page is not wide enough for all the data (for example because there are some very long club names), it has to shorten some columns. For this, the system calculates the standard deviation of all columns and reduces the width of those columns width the highest standard deviation. This means, that columns containing for example a time (all times have the same length) would not be reduced in width as this would cut the time in every line. Instead the system will reduce the width of, for example, the club name, so that only a few outliers will be cut.

Dynamic Formatting

Dynamic formatting is way to format different records differently. You can change the color, background-color and font style (bold, italic, underlined). For example, you could print all men in blue, and all women in red.

Therefore, you need to enter an expression in the extended field settings (double arrow button >>) like this:

```
iif([Sex]="f"; "C(#FF0000)"; "C(#0000FF)")
```

The dynamic formatting can be defined either for the entire line or only for certain columns. For further details please read the article in the race result Online Help.

Entering Complex Filters

The filter settings for a list offer an easy definition of your filter, for example:

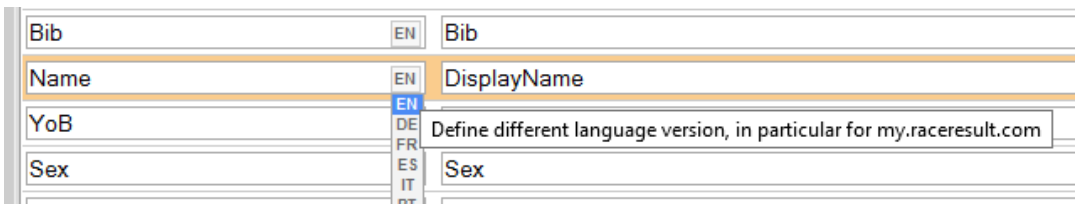
Conjunction	Field	Operator	Value
	Rank3	>	0
and	Rank3	<=	3
and		=	

If you need to enter a complex nested filter, the GUI may be tedious. In that case simply enter your filter in the first line like this:

```
iif(([Contest]=2 AND T2>5) OR ([Contest]=3 AND T2=5); 1; 0)  =  1
```

Multi-Language Reports

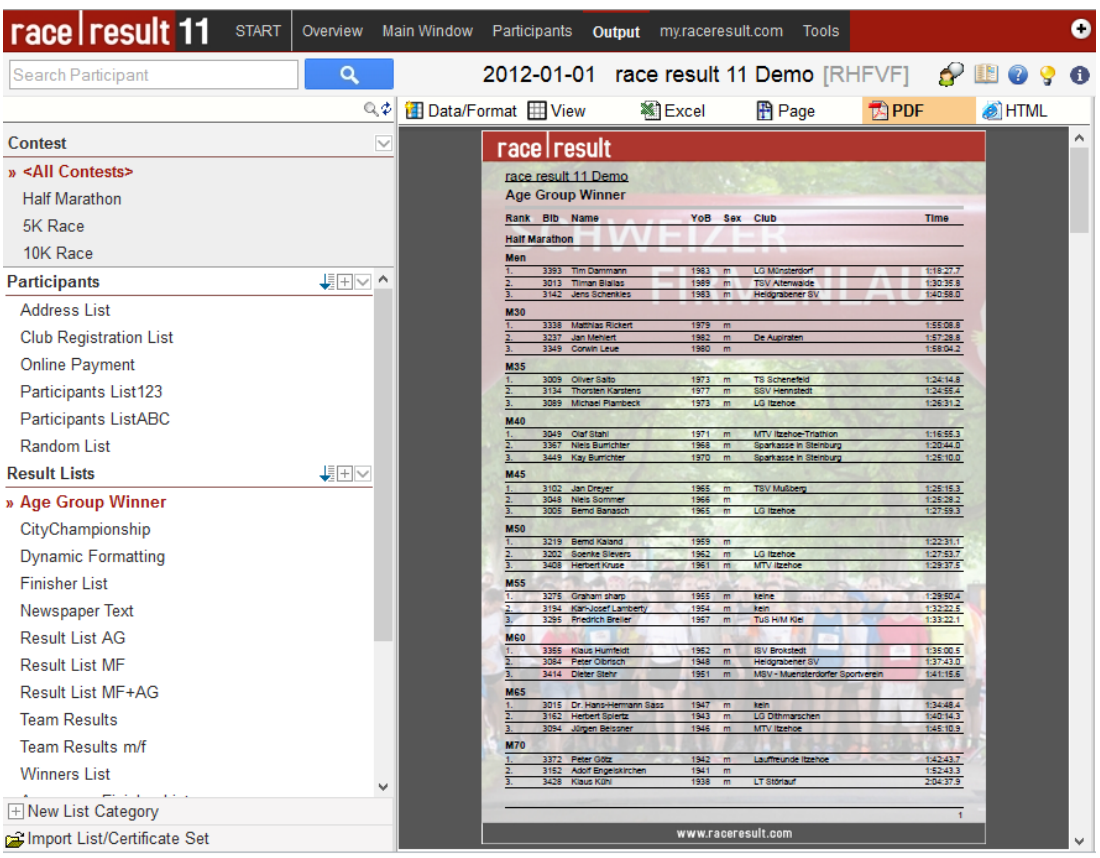
When publishing a result list on my.raceresult.com you may would like to have for example the column headers in different languages. Therefore, you will find a small language box at the end of the label fields where you can enter your texts for different languages:



Depending on the language chosen on my.raceresult.com, the accordant text will be shown. If you haven't entered the text in a specific language, the system will show the English version. If an English version is not available either, the system will use the language entered first.

List Background

The settings in the *Page* tab can be used to change the layout of your report, for example you can define the page size and format, page margins, header and footer texts, header and footer pictures. If this does not fulfill all your needs you can also put a certificate in the background of your list, which allows you to position texts and pictures freely like this:



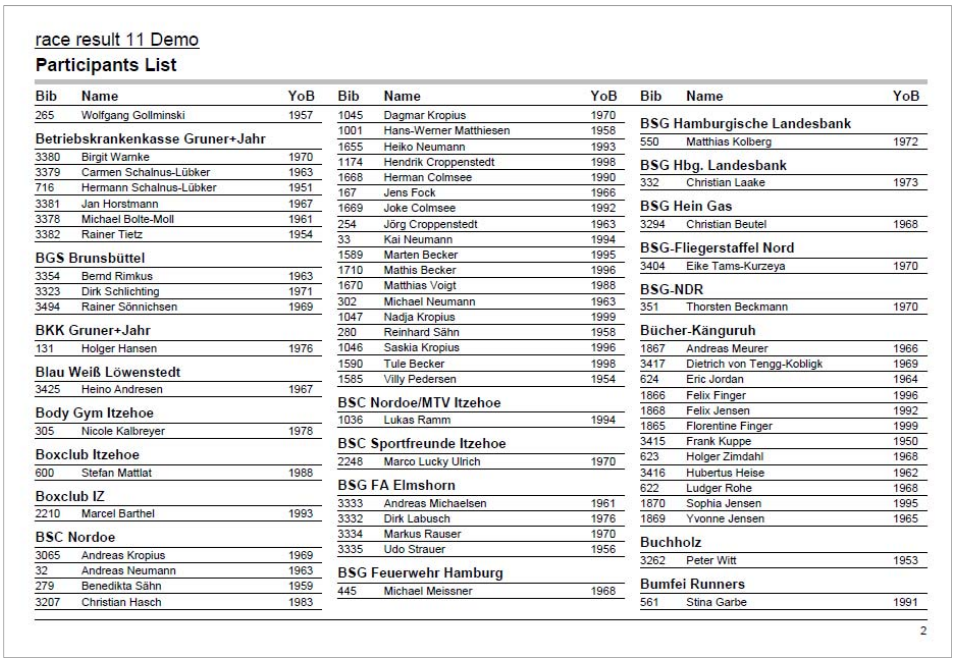
This option gives you unlimited flexibility in designing your reports.

Cover Sheet, Back Sheet

Similarly, you can put an additional cover sheet at the beginning of your report and an additional back banner page at the end of the report. Therefore, set up the page as certificate in the Designer and then select it in the page settings of your list.

Multi Column Reports

In some cases it can be helpful to split the page into several columns:



race result 11 Demo

Participants List

Bib	Name	YoB
265	Wolfgang Gollminski	1957
Betriebskrankenkasse Gruner+Jahr		
3380	Birgit Warnke	1970
3379	Carmen Schalnus-Lübker	1963
716	Hermann Schalnus-Lübker	1951
3381	Jan Horstmann	1967
3378	Michael Bolte-Moll	1961
3382	Rainer Tietz	1954
BGS Brunsbüttel		
3354	Bernd Rimkus	1963
3323	Dirk Schlichting	1971
3494	Rainer Sönnichsen	1969
BKK Gruner+Jahr		
131	Holger Hansen	1976
Blau Weiß Löwenstedt		
3425	Heino Andresen	1967
Body Gym Itzehoe		
305	Nicole Kalbreyer	1978
Boxclub Itzehoe		
600	Stefan Mattlat	1988
Boxclub IZ		
2210	Marcel Barthel	1993
BSC Nordoe		
3065	Andreas Kropius	1969
32	Andreas Neumann	1963
279	Benedikta Sähn	1959
3207	Christian Hasch	1983
1045	Dagmar Kropius	1970
1001	Hans-Werner Matthiesen	1958
1655	Heiko Neumann	1993
1174	Hendrik Croppenstedt	1998
1668	Herman Colmsee	1990
167	Jens Fock	1966
1669	Joke Colmsee	1992
254	Jörg Croppenstedt	1963
33	Kai Neumann	1994
1589	Marten Becker	1995
1710	Mathis Becker	1996
1670	Matthias Voigt	1988
302	Michael Neumann	1963
1047	Nadja Kropius	1999
280	Reinhard Sähn	1958
1046	Saskia Kropius	1996
1590	Tule Becker	1998
1585	Villy Pedersen	1954
BSC Nordoe/MTV Itzehoe		
1036	Lukas Ramm	1994
BSC Sportfreunde Itzehoe		
2248	Marco Lucky Ulrich	1970
BSG FA Elmshorn		
3333	Andreas Michaelsen	1961
3332	Dirk Labusch	1976
3334	Markus Rauser	1970
3335	Udo Strauer	1956
BSG Feuerwehr Hamburg		
445	Michael Meissner	1968
BSG Hamburgische Landesbank		
550	Matthias Kolberg	1972
BSG Hbg. Landesbank		
332	Christian Laake	1973
BSG Hein Gas		
3294	Christian Beutel	1968
BSG-Fliegerstaffel Nord		
3404	Eike Tams-Kurzeya	1970
BSG-NDR		
351	Thorsten Beckmann	1970
Bücher-Känguruh		
1867	Andreas Meurer	1966
3417	Dietrich von Tengg-Kobligk	1969
624	Eric Jordan	1964
1866	Felix Finger	1996
1868	Felix Jensen	1992
1865	Florentine Finger	1999
3415	Frank Kuppe	1950
623	Holger Zimdahl	1968
3416	Hubertus Heise	1962
622	Ludger Rohe	1968
1870	Sophia Jensen	1995
1869	Yvonne Jensen	1965
Buchholz		
3262	Peter Witt	1953
Bumfei Runners		
561	Stina Garbe	1991

2

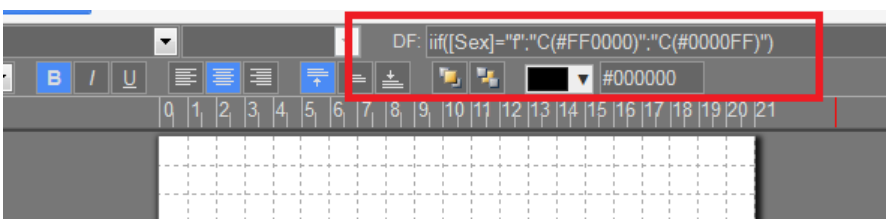
Simply define the number of columns and the column spacing in the *Page* settings of your list.

5.2. Certificates

Certificates are pretty straightforward: you set them up in the Designer (Tools -> Designer) by placing data fields, fixed text fields and pictures on the page. The certificate can either be complete and include the background image (for example to publish it online) or it only contains the data fields and will be printed on the pre-printed sheets.

Dynamic Formatting

Dynamic formatting can be used similar to lists. It can be defined in the top right corner:

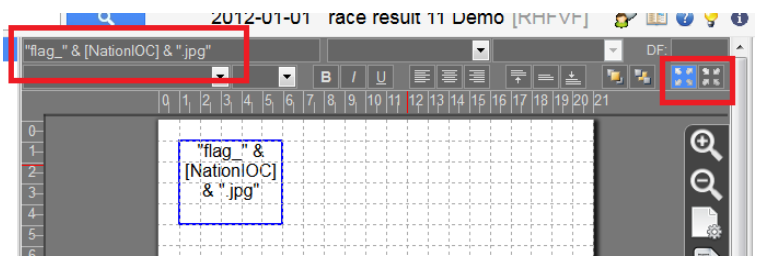


Data Dependant Images

Using data dependent images, you can use different images for different participants on the same certificate. Assume you would like to show the flag of the participant's nationality on the certificate. The flags are saved as *flag_XXX.jpg* where XXX is the name of the country in IOC format.

Then, add a data dependent image and use this expression:

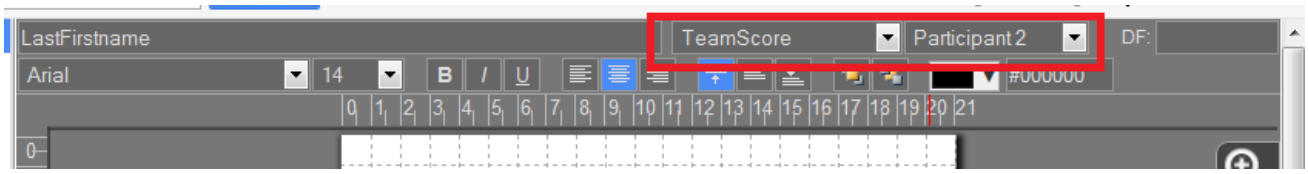
```
"flag_" & [NationIOC] & ".jpg"
```



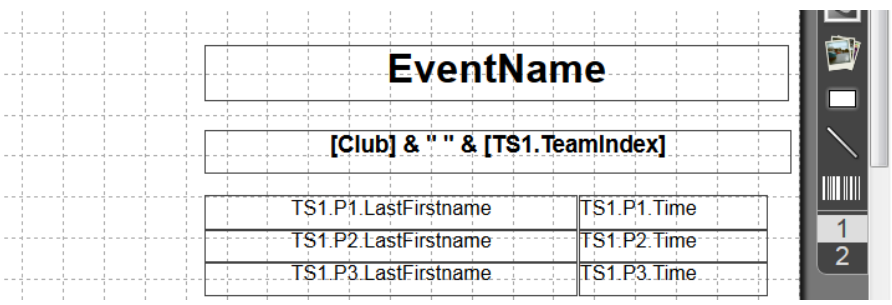
If the pictures do not have the same dimension you may want to turn of the stretch mode and instead scale the image maintaining the image ratio (see top right corner of the screenshot).

Team Certificates

Certificates for teams work exactly the same like a normal certificate. Instead of the individual rank and time you would show the team's rank and time. If you would like to show also information of the other team members, you can simply enter the field/expression in the top left corner and then select the team score and the participant.

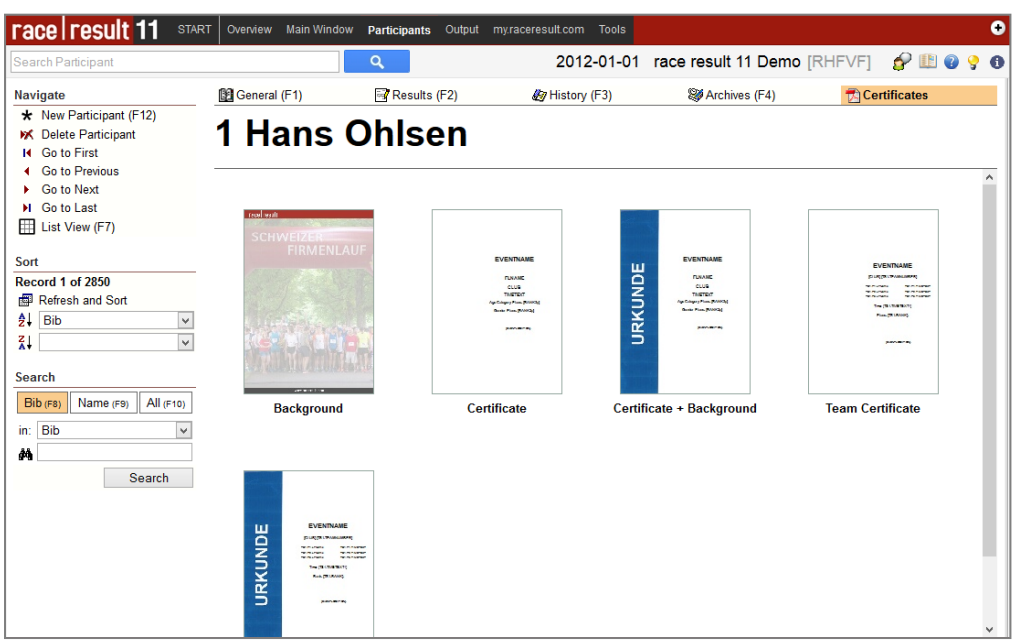


For example, to show the last name of the second participant of the team (according to team score 1), the system will use the field TS1.P2.Lastname. In the preview, you can see that selecting a team score/participant only helps you to set up your certificate, but internally the field name will be used.



Printing Certificates

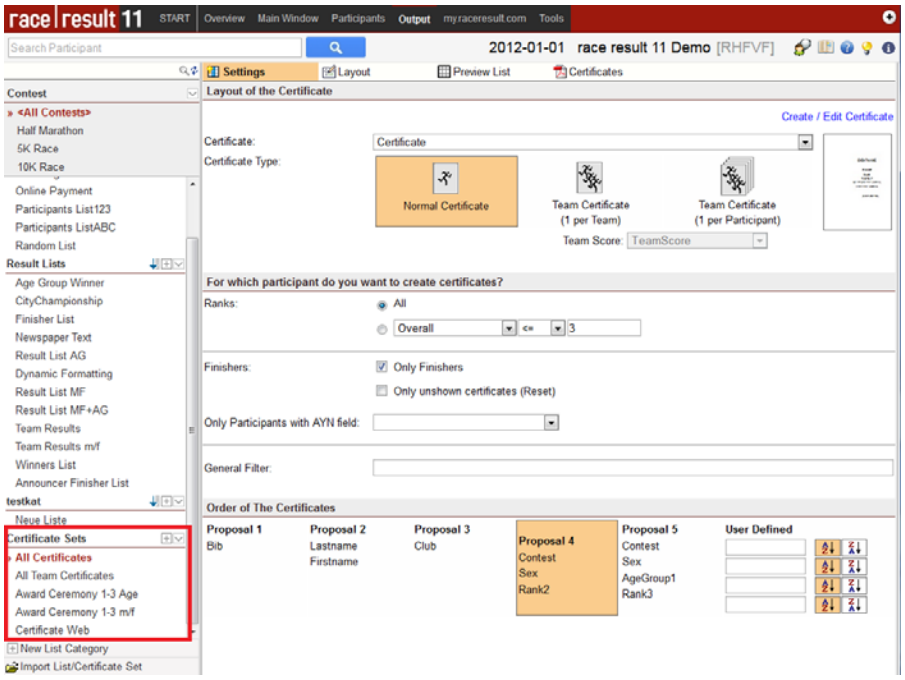
The certificates can be accessed and printed in the Participants Window through the *Certificates* tab:



In most cases, however, you need to print several certificates at once which can be done using a Certificate Set in the Output Window:

Certificate Sets

A certificate set is simply a set of certificates for certain participants in a certain order and using a specific design (certificate). The certificate sets can be accessed in the bottom left corner of the Output Window:



When creating a new event file, some standard certificate sets will be set up automatically, for example to print the certificates for the first three women and men per contest for the award ceremony.

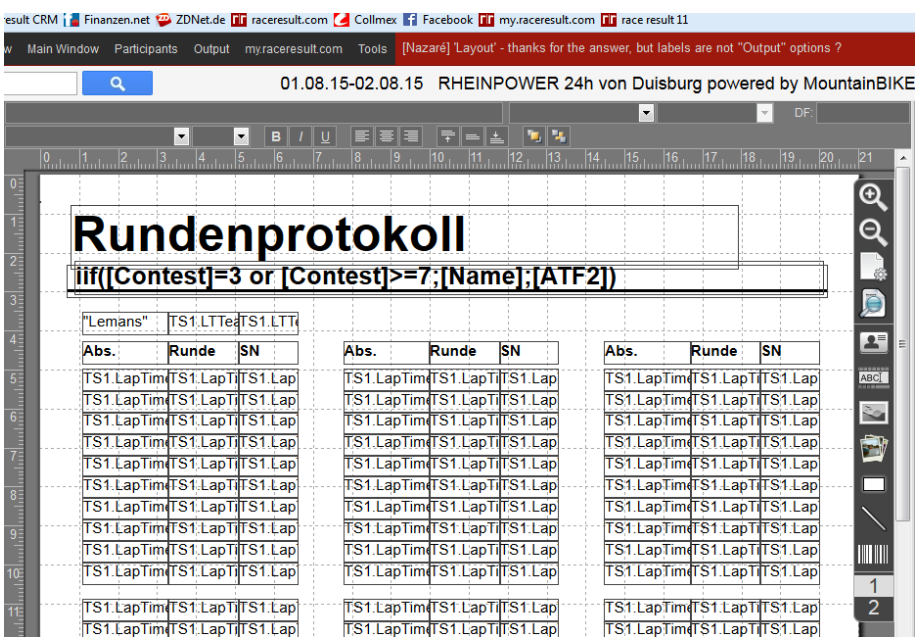
Publishing Certificates

Certificates can be published easily on my.raceresult.com. Therefore, go to the my.raceresult.com settings -> Publish -> Certificates and simply check those certificate set you would like to publish.

Certificate Sets will be used to publish certificates, because this way you can define both, the design/certificate you would like to publish and for whom you would like to publish the certificate. Only those participants fulfilling the filter in the certificate set will be able to download the certificate from my.raceresult.com.

What Else Certificates Can Do

Besides certificates, the certificates functionality can be used for many different applications. For example, it can be used to show the lap times for a team lap race:



Other options are stickers or labels, or printing on race bibs or envelopes, etc.

5.3. Presentations

The Presenter program (Windows based) gives you extensive possibilities to present results on a screen. This can be either a standard result list displayed page by page on a video wall or a special list having only one page and showing for example a leader board.

Page by Page Presentation

These presentations are straightforward as they are so similar to a normal result list. For example, consider the following results from a 7h ski lap race (3 riders per team, one of them on the track at a time, make as many laps [mountain up in the lift and down on the ski] as possible in 7 hours):

TeamRace Meiringen-Hasliberg 2014 - Teams+Fahrer TOP3

Startnr.	Name	Runden	Etappen	Schnell	Mittel	Langsam	Lemans	Summe
Damen								
1.	**d'smilies :-) (9)**						**7:05:05,5**	**35 Runden**
27	Capaul, Kathrin	12	6	11:16,6	12:07,2	12:43,8		2:25:26,8
26*	Capaul, Franziska	15	7	11:19,2	12:02,9	12:46,3		3:00:44,4
25	Stoffel, Katja	8	4	11:25,1	12:05,2	12:50,2	02:12,1	1:36:42,0
2.	**SchwurzchungHB77 (1)**						**7:07:30,2**	**35 Runden**
2	Vögtli, Sarah	11	5	11:26,8	12:00,2	12:29,5		2:12:03,1
1*	Vögtli, Anna-Lena	12	6	11:40,0	12:16,9	13:11,7		2:27:23,3
3	Nägeli, Diana	12	6	11:30,7	12:09,4	12:50,4	02:09,9	2:25:53,7
3.	**Gipfistirmerinnä (2)**						**7:13:15,3**	**35 Runden**
5*	Furrer, Karoline	13	6	11:25,7	12:09,6	13:27,5		2:38:04,8
6	Furrer, Sandra	11	6	11:27,0	12:27,0	13:14,6		2:16:58,0
4	Furrer, Martina	11	6	11:26,0	12:21,7	12:58,8	02:13,2	2:15:59,0
Herren								
1.	**Ds Dritt simer Bissig ohni iisä Teamcaptain dits J (56)**						**7:06:58,5**	**39 Runden**
168*	Ammann, Roland	13	6	10:36,6	10:58,7	11:42,8		2:22:44,1
166	Kohler, Sandro	11	6	10:45,2	11:01,3	11:18,7		2:01:15,1
167	Ammann, Andreas	15	6	10:22,7	10:46,4	11:05,7	01:22,0	2:41:37,1
2.	**Skiclub Schwarzsee (70)**						**7:09:24,0**	**39 Runden**
209*	Lötscher, Sven	12	6	10:44,9	11:12,8	12:26,3		2:14:33,8
210	Thalmann, Dominik	13	6	10:27,7	10:53,5	11:19,6		2:21:36,1
208	Thalmann, Benjamin	14	6	10:31,3	10:50,0	11:17,8	01:33,0	2:31:40,9

The presentation is running on a large TV in the team area where the teams can check their positions any time. It is going through the list page by page and starting over from the top after the last page.

Leader Board / Live Presentations

Leader board presentations can be nice for both, the announcer commenting on the race as well as for a video wall. As example consider this leader board:

Rangliste Team Men

Rg.	SN.	Team	Inline Wechsel	Swim Wechsel	MTB Wechsel	Laufen	Gesamt Diff.
1.	716	JuraDefi	17:54,1 (2.) 00:35,1 Joris Dubail	21:52,6 (2.) 00:25,3 Oriet Gilian	1:14:30,3 (1.) 00:27,7 Blanchard Tristan	45:33,3 (1.) Jeanbourquin Patrick	**2:41:18,9** -
2.	701	Gilgen Door Systems	19:08,2 (3.) 00:42,2 Giorgetta Daniele	28:46,9 (15.) 00:40,5 Renner Jörg	1:20:42,4 (2.) 00:22,6 Ryffel Lorenz	50:00,5 (3.) Burri Hans	**3:00:23,6** *+19:04,7*
3.	711	Tomahawks	20:01,1 (5.) 00:31,7 Woodtli Marc	23:01,6 (7.) 00:43,8 Tschanz Stefan	1:30:18,2 (3.) 00:36,6 Lüthi Stefan	55:06,9 (5.) Bühler Manfred	**3:10:20,1** *+29:01,2*
4.	707	kraftwerk-domaspo	25:27,6 (14.) 04:05,2 Lauri Christian	22:45,2 (6.) 00:51,3 Vitis Amir	1:41:30,4 (10.) 00:50,4 Sutter Stefan	47:59,0 (2.) Leuenberger Niklaus	**3:23:29,4** *+42:10,5*
5.	712	Bier gewinnt	25:23,8 (13.) 00:39,4 Kühni Marc	26:02,3 (11.) 00:45,5 Spieler Roman	1:32:06,0 (5.) 00:30,0 Gerber Philipp	1:03:14,6 (8.) SANTSCHI Oliver	**3:28:42,0** *+47:23,1*

Rangliste Team Women

Rg.	SN.	Team	Inline Wechsel	Swim Wechsel	MTB Wechsel	Laufen	Gesamt Diff.
1.	810	Mt. Pilatus Power	26:48,1 (3.) 00:41,8 Gander Martina	25:41,1 (1.) 00:51,0 Stucki Sarah	1:53:49,4 (3.) 00:30,9 Schaerer Simone	55:55,4 (2.) Wenger Iris	**3:44:18,0** -
2.	807	Bikini Girls	26:16,7 (2.) 01:05,0 Aebi Nicola-Sandra	27:13,2 (4.) 00:38,7 Stocker Ines	1:45:52,1 (1.) 00:32,3 Saner Rita	1:06:59,6 (4.) ROTHENBÜHLER Stefanie	**3:48:37,9** *+4:19,9*
3.	811	BGH Alumni +	24:20,4 (1.) 00:38,3 Wehrli Daniela	36:40,6 (11.) 00:53,2 Knecht Arnika	2:04:05,1 (7.) 00:27,2 Hofer Christina	52:50,5 (1.) Hüsler Barbara	**3:59:55,6** *+15:37,5*
4.	804	Hoch 2 Power	28:04,1 (4.) 00:29,8 Kummer Sandra	27:34,6 (5.) 00:32,3 Bichsel Colette	1:51:55,5 (2.) 00:30,7 Ruef Nina	1:14:44,8 (10.) Kopp Nicole	**4:03:52,1** *+19:34,0*
5.	803	Söll emal cho!	30:30,2 (8.) 00:42,5 Marti Nicole	27:02,8 (3.) 00:47,3 Niggli Selina	1:58:04,6 (5.) 00:40,0 Mösch Florence	1:11:24,5 (7.) Liniger Regula	**4:09:12,1** *+24:54,0*

Rangliste Team Mixed

Rg.	SN.	Team	Inline Wechsel	Swim Wechsel	MTB Wechsel	Laufen	Gesamt Diff.
1.	517	Zaboo Team Pro 1	18:38,8 (6.) 00:35,2 Sumi Patrick	23:15,7 (12.) 00:35,5 Meyer Valérie	1:12:40,6 (1.) 00:21,3 Dähler Friedrich	45:07,6 (2.) Oldani Lukas	**2:41:15,0** -
2.	560	R'ADYS ZABOO	19:07,2 (7.) 00:35,5 Rechsteiner Reto	23:14,1 (11.) 00:27,5 Brechbühl Andrea	1:18:30,2 (3.) 00:31,8 Zahnd Simon	46:10,1 (3.) BIERI Matthias	**2:48:36,7** *+7:21,7*
3.	616	Go for it	19:59,9 (12.) 00:28,7 Ackeret Jeanine	21:05,8 (5.) 00:31,7 Müller Sebastian	1:25:12,8 (14.) 00:26,5 Rentsch Manuel	45:00,1 (1.) Ackeret Christian	**2:52:45,7** *+11:30,7*
4.	518	Zaboo Team Pro 2	21:08,6 (24.) 00:40,8 Meyer Rico	25:23,1 (30.) 00:34,4 Beaudet Amélie	1:19:00,6 (4.) 00:27,3 German Benno	48:06,7 (4.) Botha Joca	**2:55:21,8** *+14:06,8*
5.	703	Backyard Legends	17:68,2 (2.) 00:32,3 Friedli Yannick	20:29,4 (1.) 00:30,5 Roller Marc	1:22:20,8 (5.) 00:26,8 hängärtner yves	57:38,7 (30.) SCHÄR Yvonne	**2:59:06,8** *+17:51,8*

Rangliste Couple

Rg.	SN	Team	Inline Wechsel	Swim Wechsel	MTB Wechsel	Laufen	Gesamt Dif.
1.	321	Team Mahu	17:59,0 (2.) 00:27,2 Berger Marc	21:14,4 (3.) 00:30,7 Matter Brügger Sibylle	1:21:56,2 (2.) 00:22,8 Berger Marc	56:39,5 (17.) Matter Brügger Sibylle	**2:59:10,0** -
2.	384	R'ADYS / ZABOO BIKES	18:39,5 (3.) 01:48,9 Bürgi Yvonne	30:33,1 (86.) 00:30,6 Bürgi Yvonne	1:14:00,3 (1.) 00:49,4 Spaeth Sandro	54:09,9 (8.) SPAETH Sandro	**3:00:31,9** *+1:21,8*
3.	306	Les paresseux	19:16,4 (6.) 00:30,8 Koller Martin	23:41,0 (11.) 00:37,7 Schramek Diana	1:25:28,7 (6.) 00:26,3 Koller Martin	53:13,6 (6.) Schramek Diana	**3:03:14,8** *+4:04,7*
4.	410	Reto&Susu	19:59,3 (14.) 00:27,2 Aschenberger Reto	23:26,9 (10.) 00:36,6 Gries Susanne	1:27:43,2 (9.) 00:21,2 Aschenberger Reto	54:32,9 (10.) Gries Susanne	**3:07:07,7** *+7:57,7*
5.	398	VS-Express	19:34,6 (9.) 00:31,3 Willisch Marco	25:17,6 (27.) 00:44,0 Ruffiner Rachel	1:22:57,9 (3.) 00:21,1 Willisch Marco	1:03:54,1 (40.) RUFFINER Rachel	**3:13:20,9** *+14:10,8*

This event is a multisport event (inline, swim, bike, run) with couples and teams of 4. The presentation shows the 4 categories via the Split Screen option of the Presenter. For each category, it only shows the leading

teams and refreshes the information every second so that the announcer can see everything in almost real time. The lists are standard lists with three lines per record and a filter like Rank>0 AND Rank<=5

Note that some pictures and some colors can make the presentation look much better:

2009 UCI CYCLO-CROSS WORLD CHAMPIONSHIPS

HOOGERHEIDE THE NETHERLANDS

Final Results Men Elite

Rank	Bib No.	Name	YoB	Nation	Laps	Time	Gap Top	Gap Prev
1.	11	ALBERT Niels	1986	BELGIUM	10	1:02:24,2	-	-
2.	15	STYBAR Zdenek	1985	CZECH REPUBLIC	10	1:02:46,1	+0:21,9	+0:21,9
3.	8	NYS Sven	1976	BELGIUM	10	1:03:02,6	+0:38,4	+0:16,5
4.	9	WELLENS Bart	1978	BELGIUM	10	1:03:34,8	+1:10,6	+0:32,2
5.	29	MOUREY Francis	1980	FRANCE	10	1:03:47,7	+1:23,5	+0:12,9
6.	10	PAUWELS Kevin	1984	BELGIUM	10	1:03:47,9	+1:23,7	+0:00,2
7.	14	VANTHOURENHOUT Sven	1981	BELGIUM	10	1:03:48,1	+1:23,9	+0:00,2
8.	27	ZAHNER Simon	1983	SWITZERLAND	10	1:03:48,1	+1:23,9	+0:00,0
9.	30	CHAINEL Steve	1983	FRANCE	10	1:03:48,1	+1:23,9	+0:00,0
10.	12	VANTORNOUT Klaas	1982	BELGIUM	10	1:03:48,2	+1:24,0	+0:00,1
11.	22	FONTANA Marco Aurelio	1984	ITALY	10	1:03:48,4	+1:24,2	+0:00,2
12.	2	AL Thijs	1980	NETHERLANDS	10	1:03:48,5	+1:24,3	+0:00,1
13.	21	FRANZOI Enrico	1982	ITALY	10	1:03:49,0	+1:24,8	+0:00,5
14.	32	LOPEZ Jonathan	1986	FRANCE	10	1:03:50,7	+1:26,5	+0:01,7
15.	26	HEULE Christian	1975	SWITZERLAND	10	1:03:53,3	+1:29,1	+0:02,6
16.	3	DE KNEGT Gerben	1975	NETHERLANDS	10	1:04:14,8	+1:50,6	+0:21,5
17.	17	DLASK Petr	1976	CZECH REPUBLIC	10	1:04:15,2	+1:51,0	+0:00,4
18.	19	AUSBUHER Kamil	1975	CZECH REPUBLIC	10	1:04:15,3	+1:51,1	+0:00,1
19.	13	VERVECKEN Erwin	1972	BELGIUM	10	1:04:15,3	+1:51,1	+0:00,0
20.	1	BOOM Lars	1985	NETHERLANDS	10	1:04:15,4	+1:51,2	+0:00,1
21.	6	VAN IJZENDOORN Eddy	1985	NETHERLANDS	10	1:04:15,9	+1:51,7	+0:00,5

RHEINPOWER 24h von Duisburg powered by MountainBIKE 2014

LIVE TIMING - Results Class

Bib	Name	Laps	Last	Avg	Min	Max
4er-Team m						
1. RSV Radsportverein Coesfeld			68 Laps	24:16:28		
4007-1	HÜWE Marco	17	16:21	17:04	16:21	17:42
4007-2	PÖTER Marcel	17	17:14	17:32	17:06	18:25
4007-3	BREULMANN Christopher	16	17:49	17:57	17:03	18:24
*4007-4	KRÖGER Ralph	18	17:30	31:12	16:13	4:26:54
2. RS Kraichgau			67 Laps	24:17:44		
4179-1	GEIGER Uwe	17	17:23	18:12	17:19	19:45
4179-2	NEUWEILER Jens	17	17:12	17:38	16:43	18:40
*4179-3	BRECHT Manuel	17	18:16	31:47	17:17	4:16:06
4179-4	GARTMANN Stefan	16	18:10	18:00	16:52	19:52
3. MSC-Wiesenbach			66 Laps	24:09:38		
4145-1	DUKEK Klaus	16	17:31	17:58	17:15	18:53
*4145-2	SCHWARZER Roland	17	17:02	17:29	16:35	18:15
4145-3	CHARLY Mair	17	17:58	32:43	17:11	4:18:15
4145-4	SASSE Simon	16	17:10	17:54	16:33	18:45
4. www.liquid-life.de			65 Laps	24:13:07		
*4176-1	KAISER Leon	23	17:06	29:05	16:01	4:20:30
4176-2	DRÖGE Daniel	10	18:26	18:04	17:22	19:30
4176-3	BRAMBRING Dirk	17	24:16	18:30	16:57	24:16
4176-4	KOCH Alexander	15	17:32	17:57	17:08	19:01
5. Team Hüttenzauber			64 Laps	24:08:00		
*4120-1	HÜTING Fabian	16	17:49	18:32	17:32	19:29
4120-2	HÜTING Marvin	17	18:28	18:22	17:25	19:34
4120-3	FUNCKE Michael	15	17:49	18:15	17:19	18:57
4120-4	DAHMEN Markus	16	19:36	34:05	17:41	4:15:28
6. Team Siemens Turbo 4.4			64 Laps	24:08:11		
4255-1	GUTEWORT Gordon	14	18:33	19:24	18:23	20:29
4255-2	STEIGMANN Matthias	16	17:57	34:05	16:59	4:31:48
4255-3	WALDE Daniel	16	17:15	17:57	17:06	19:05
*4255-4	KALB Dieter	18	17:52	17:57	17:15	18:53

Bib	Name	Laps	Last	Avg	Min	Max
7. Berufsfeuerwehr Aachen1			64 Laps	24:10:16		
4044-1	HILBURGER Ingo	17	17:47	18:36	17:27	19:48
4044-2	RITZERFELD Rene	17	17:16	18:27	17:10	20:04
*4044-3	KOCH Silvan	18	17:37	31:30	16:48	4:13:21
4044-4	RODER Stefan	12	18:46	19:27	18:13	20:38
8. BSB Fanta 4			63 Laps	24:01:50		
4237-1	THORWARTH Matthias	14	20:09	20:21	18:42	21:28
*4237-2	THIEL Heiko	17	17:13	32:41	16:40	4:30:35
4237-3	THIEL Dominik	16	18:28	18:15	17:03	18:59
4237-4	WASSERZIEHER Matthias	16	17:11	17:55	17:11	19:23
9. SIG Koblenz			63 Laps	24:07:29		
4215-1	HAGEMEISTER Stefan	16	18:03	18:47	17:40	20:24
4215-2	SCHNEIDER Felix	15	17:07	33:43	16:22	4:19:57
*4215-3	ÜBING Gordon	16	18:40	20:01	18:40	21:35
4215-4	REINART Marc	16	18:36	18:38	17:25	19:54
10. Deloitte			63 Laps	24:14:55		
4259-1	SPAHN Michael	19	18:04	31:31	17:31	4:25:38
*4259-2	GOLDGRUBER Florian	20	18:00	18:40	17:55	20:30
4259-3	HÜPEN Holger	4	18:51	19:40	18:51	20:39
4259-4	SIMAK Jan	20	18:48	19:01	17:53	20:23
11. www.liquid-life.de / Neumann Brilon			62 Laps	24:02:59		
*4201-1	KLEIN Pascal	18	17:12	32:07	17:09	4:31:04
4201-2	KLUGE Silvio	14	18:02	18:44	17:50	19:43
4201-3	SCHIRBEL Jörg	13	20:34	20:39	19:47	21:37
4201-4	MITCHEL Ryan	17	17:31	18:27	17:17	20:19
12. Moerser TV Triathlon 1			62 Laps	24:04:15		
4174-1	BULLERMANN Tim	15	18:45	18:41	17:50	19:10
*4174-2	BECKER Jan-Matti	17	18:28	33:31	17:20	4:31:03
4174-3	EGGEN Max	16	18:51	19:02	17:20	19:54
4174-4	REINECKE Dominik	14	18:47	19:12	18:16	20:22

01.11.2014 www.sportservicehamburg.de 16:45:30

Vorankünder

Uhrzeit	W	Finisher	w	SNR	Name	Top	Gap	Zeit	AK	Alter	Verein	MW-PI	SE	AK-PI
21:20:59	A-211-A	392 | 627 im Ziel		A83	Michael KONZE (GER)	+1280:58	+0:01	16:21:39	M40	43	Rad-Treff Borchen e.V.	398.		147. M40
21:20:58	A-211-A	393 | 627 im Ziel		A724	David HASLAUER	+1280:57	+1:58	16:21:40	MHK	19	VKB-Bank	397.		46. MHK
21:19:00	A-211-A	391 | 627 im Ziel		A748	Paolo CAPRINI (ITA)	+1278:59	+0:04	16:19:27	M50	53	Tirreno bike	396.		32. M50
21:18:56	A-211-A	390 | 627 im Ziel		A507	Roland DOBRINDT (GER)	+1278:55	+0:01	16:19:25	M40	49	Wurzelhopser	395.		146. M40
21:18:55	A-211-A	389 | 627 im Ziel		A651	Gaute RISNES (NOR)	+1278:54	+0:38	16:19:24	M40	44	Team Telemark	394.		145. M40
21:18:17	A-211-A	388 | 627 im Ziel		A643	Marc VAN DER VOSSEN (NED)	+1278:16	+1:44	16:18:42	M30	33	Noordbikers	393.		173. M30
21:16:33	A-211-A	387 | 627 im Ziel		A396	Jan-Olaf KOSCHINSKI (GER)	+1276:32	+0:06	16:16:47	M40	47	DPD-No limits	392.		144. M40
21:16:27	A-211-A	386 | 627 im Ziel		A435	Vladimír PALÚCH (SVK)	+1276:26	+0:00	16:16:47	M50	56	The Slovak Snail Highlanders	391.		31. M50
21:16:27	A-211-A	384 | 627 im Ziel		A575	Filippo ALTOBELLI (ITA)	+1276:26	+0:59	16:16:42	M40	41	G.S. Amoruso	390.		143. M40
21:16:27	A-211-A	385 | 627 im Ziel	W	A290	Michaela TUMOVÁ (CZE)	+193:02	+67:59	16:16:43	WHK	23	Prštice	8.		4. WHK
21:15:28	A-211-A	383 | 627 im Ziel		A443	Silvio GRAUPNER (GER)	+1275:27	+28:48	16:15:49	M30	38	Prost Rico	389.		172. M30
20:46:40	A-211-A	344 | 627 im Ziel		A545	Zoltán FÜLÖP (HUN)	+1246:39	+0:00	15:46:57	M30	31	Bringabanda SC	388.		171. M30
20:46:40	A-211-A	343 | 627 im Ziel		A323	Jirí CECHURA (CZE)	+1246:39	+1:29	15:46:57	M50	49	Alltraining.cz	387.		30. M50
20:45:11	A-211-A	340 | 627 im Ziel		A177	Lukas PÖHL	+1245:10	+0:00	15:45:24	M30	31	TSD	386.		170. M30
20:45:11	A-211-A	342 | 627 im Ziel		A634	Marc HULSHOF (NED)	+1245:10	+0:00	15:45:32	M40	49	WV de Meteoor	385.		142. M40
20:45:11	A-211-A	341 | 627 im Ziel		A176	Georg LÖSCHL	+1245:10	+0:33	15:45:25	M30	35	TSD	384.		169. M30
20:44:38	A-211-A	339 | 627 im Ziel		A288	Michal KADLEC (CZE)	+1244:37	+0:13	15:44:58	M40	41	Podolí	383.		141. M40
20:44:25	A-211-A	337 | 627 im Ziel		A701	Dieter SCHULZKI (GER)	+1244:24	+0:03	15:44:36	M40	49	CSV-MTB-Team	382.		140. M40
20:44:22	A-211-A	338 | 627 im Ziel		A623	Chris DREVEL (NED)	+1244:21	+0:03	15:44:45	M40	44	Le Champion	381.		139. M40
20:44:19	A-211-A	335 | 627 im Ziel		A362	Javier SIMON (ESP)	+1244:18	+0:02	15:44:32	M40	41	APLbike	380.		138. M40
20:44:17	A-211-A	336 | 627 im Ziel		A648	Mattijs VINKESTEIJN (NED)	+1244:16	+0:01	15:44:33	M30	33	Zwaar 2002	379.		168. M30

01.11.2014 16:48:49

6. RACE SETUP GUIDE

The best way to learn how to set up complex event files is to learn from examples. That is why we provided different examples in the support section on our raceresult.com website. This section will describe some general design guide lines that proved to be helpful in the past.

6.1. Choose IDs wisely

When you set up results or ranks, choose your IDs wisely. Make sure that IDs are intuitive (e.g. lap 1 in result 21, lap 2 in result 22, lap 3 in result 23, and so on – instead of lap 1 in result 17). Also make sure to leave gaps in our IDs, maybe you would like to add something in between later.

6.2. Use Field Names

You can access times/ranks/additional fields via the ID (e.g. Rank3) or via its name (e.g. AgeGroupRank). The set up of your lists and certificates will be easier to understand later if you use the names instead of the IDs. Therefore, make sure to use a short yet concise name that you will not change later.

6.3. Use User Defined Fields

Whenever you need an expression like

```
Choose([TS1.Sex]; "Men"; "Women"; "Coed")
```

create a User Defined Field. You will definitely need this expression in several reports/certificates later, so use a User Defined Field from the beginning.

6.4. Finish Result

It has been proofed helpful using ID 1 for the finish time result. If you do so, you always know that the most important time will be in result 1 and thus at the very top, for example in the Results tab of the Participants Window.

Only if you have events with different numbers of laps per contest or with several finish lines at different locations, it is very helpful to have several finish results. For example:

ID	Name	Time Format	Rounding	Formula	Location
1	AfterLap1		no rounding	T11-T0	
2	AfterLap2		no rounding	T12-T0	
3	AfterLap3		no rounding	T13-T0	
11	TimeStamp1		no rounding		
12	TimeStamp2		no rounding		
13	TimeStamp3		no rounding		
21	Lap1		no rounding	T11-T0	
22	Lap2		no rounding	T12-T11	
23	Lap3		no rounding	T13-T12	

In this case, there are three results for up to three detections (11, 12, and 13). The results 21, 22, and 23 calculate the time per lap, whereas result 1, 2 and 3 calculate the time from the gun start until finishing lap 1/2/3.

If, for example, contest 1 has two laps and contest 2 three laps, you can simply select “AfterLap2” as finish result for contest 1, and “AfterLap3” as finish result for contest 2.

Instead of using different finish results, there are the following alternatives:

a) (bad alternative) Remove results 2 and 3, rename result 1 to “FinishTime” and use the formula `TMax(11;13)-T0` . This way, the finish time will simply be the time after finishing the last lap what

creates two problems: First, you have to change the rank settings so that someone with more laps is better than someone with less laps. Second, if someone runs an additional lap for fun, he would not have the correct finish time.

b) (good alternative) Remove results 2 and 3, rename result 1 to "FinishTime" and use the formula `choose([Contest];T12;T13)-T0` . This is exactly what selecting different finish results for each contest would do in the background, but you have more options. Assume in contest 2, women run only two laps, but men three. Then you could use this formula:
```
choose([Contest]; T12; iif([Sex]="f";T12;T13)) - T0
```

For the beginner, we would recommend to simply select different finish results (complicated enough!), but the advanced user may appreciate the additional flexibility of option b).

7. ADVANCED DATA FIELDS AND FUNCTIONS

7.1. Position Fields

Ranks (chapter 3.8) are a handy concept since they are defined once and can then be used everywhere – on a list, in a certificate, in a text message. However, sometimes you may want to create a numbered list quickly without creating a rank definition.

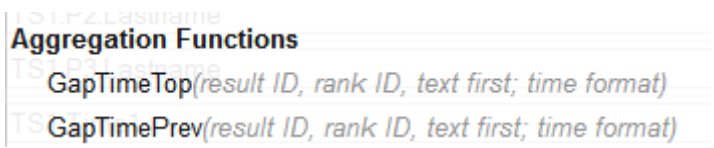
Let's assume you need a result list of all participants named John. Simply add a filter to the list and use the field *Position* instead of the rank:

Rank	Bib	Name	YoB	Sex	AgeCat	Club	Time
Half Marathon							
1	3482	John Treyke	1974	m		kein	1:43:38.0
2	3485	John Ramlow	1950	m		AGS Itzehoe	1:59:37.4
3	3468	John Berndt	1955	m		Dentaltechnik Itzehoe	2:14:41.6
5K Race							
4	1323	John Witt	1995	m	Juniors B	Klasse 4c GS Edendorf Ursula Paarmann	26:32.8

Looks good, but it should start at 1 again for the 5K race. For that, you can use the field *Position1* which will start over at 1 when the highest grouping level changes the value. *Position2* would start over at 1 when the first or second highest grouping level changes the value.

7.2. Gap Times

To show the gap between a participant and the winner or the previous athlete, you can use the two functions GapTimeTop() and GapTimePrev():



The rank defines who the winner / previous rider is (winner overall, winner gender, winner age group). *Text first* defines what to show for the first participant (does not have a faster athlete). The *time format* parameter defines how to format the gap time.

The expression

```
GapTimeTop(1; 2; "--"; "+mm:ss")
```

may have this result:

Rank	Bib	Name	YoB	Sex	AgeCat	Club	Time
Half Marathon							
Men							
1.	3049	Olaf Stahl	1971	m	M40	MTV Itzehoe-Triathlon	1:16:55.3 --
2.	3393	Tim Dammann	1983	m	Men	LG Münsterdorf	1:18:27.7 +01:32
3.	3367	Niels Burrichter	1968	m	M40	Sparkasse in Steinburg	1:20:44.0 +03:48
4.	3219	Bernd Kaland	1959	m	M50		1:22:31.1 +05:35
5.	3009	Oliver Salto	1973	m	M35	TS Schenefeld	1:24:14.8 +07:19
6.	3134	Thorsten Karstens	1977	m	M35	SSV Hennstedt	1:24:55.4 +08:00
7.	3449	Kay Burrichter	1970	m	M40	Sparkasse in Steinburg	1:25:10.0 +08:14
8.	3102	Jan Dreyer	1965	m	M45	TSV Mußberg	1:25:15.3 +08:20

7.3. Top and Prev Times

The functions GapTimeTop and GapTimePrev are actually based on the Top and Prev Times: the fields

```
RankXTopY and RankXPrevY
```

return the time in result Y of the winner / the previous athletes according to rank X. So the field Rank2Top1 may return the (decimal) time of the first man/woman. If you add an "R" the field will return the rounded value (Rank2Top1R / Rank2PRev1R), and if you add the word "Text" the field will return the formatted time (Rank2Top1Text / Rank2Prev1Text).

So, instead of *GapTimeTop(1, 2, "--"; "+mm:ss")* you could write also:

```
iif([Rank2]=1; "--"; format(TR1-[Rank2Top1R]; "+mm:ss"))
```

Frequently, RankXTopY is used in formulas calculating points. For example, the formula *550 - 250 * [Time] / [TimeOfFirst]* (minimum 0 points) could be implemented in race result like this.

```
max(0; 550-250*TR1/[Rank2Top1R])
```

7.4. Bunch Times

Bunch times are common in cycling: as long as the gaps between the riders in a field are less than, for example, 1 second, all of them receive the time of the first rider in the field.

There are two ways to do this in race result: one is easy to use but does not run in real time calculation, the other does but is more complicated.

Easy Option

The field

```
RankXPrevYRepeatedZ
```

will return the bunch time of result Y. The previous rider is defined by rank X and the gap between two riders needs to be less or equal Z seconds. If you add an "R" the field will return the rounded value, and if you add the word "Text" the field will return the formatted time.

```
Rank2Prev1Repeated1Text
```

Efficient Option

In order to calculate the bunch times efficiently, it should be set up as result in Main Window -> Results like this:

ID	Name	Time Format	Rounding	Formula	Location	
1	FinishTime		no rounding			
10	Bunch		no rounding	iif(T1-TPrev(2;1)<2; TPrev(2;10); T1)		
			no rounding			

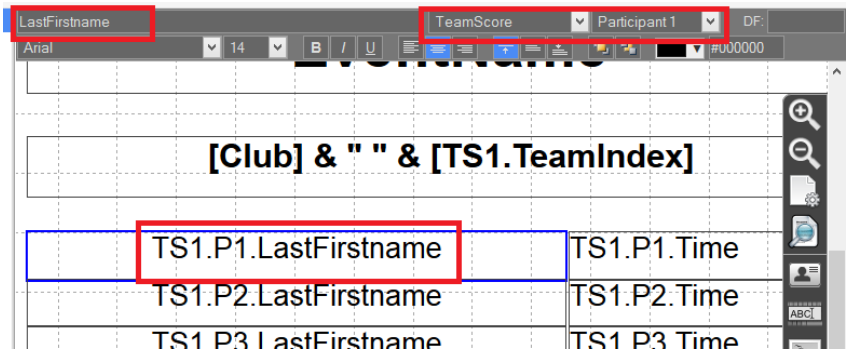
TPrev(X; Y) also returns the time of the previous rider from result Y. The previous rider is defined by rank X. So the time gap to the previous rider is

```
T1-TPrev(2;1)
```

If this is less than 2 seconds, the formula takes recursively the bunch time of the previous rider until the gap is >= 2 seconds. If the gap is >= 2 seconds, the time from result 1 will be used.

7.5. Team Scores: Accessing Other Participants

When creating a race certificate for a team, you can access the data of all members of the team:



As you can see, you can access the data of the yth member of the team according to team score x via TSx.Py.

You can use this to turn the detailed team results ..

Bib	Name		Age Category	YoB
Half Marathon				
1.	**LG Itzehoe**			**4:22:24,2**
3089	Michael Plambeck	1:26:31.2	M35	1973
3202	Soenke Sievers	1:27:53.7	M50	1962
3005	Bernd Banasch	1:27:59.3	M45	1965
2.	**Sparkasse in Steinburg**			**4:25:00,0**
3367	Niels Burrichter	1:20:44.0	M40	1968
3449	Kay Burrichter	1:25:10.0	M40	1970
3365	Karsten Wilkens	1:39:06.0	M50	1962
3.	**kein**			**4:41:02,6**
3194	Karl-Josef Lamberty	1:32:22.5	M55	1954
3352	Walter Lemor	1:33:51.7	M40	1971
3015	Dr. Hans-Hermann Sass	1:34:48.4	M65	1947
4.	**LT Störlauf**			**4:41:50,9**
3057	Roger Steffens	1:28:15.8	M40	1969
3119	Mathias Funk	1:32:52.3	M35	1973
3431	Joachim Jürgens	1:40:42.8	M50	1958

.. into a short version with one line per team, still showing the last names of the athletes.

Data/Format | View | Excel | Page | PDF | HTML

Rank	Club	Rider1	Rider2	Rider3	Time
Half Marathon					
1	LG Itzehoe	Plambeck	Sievers	Banasch	4:22:24,2
2	Sparkasse in Steinburg	Burrichter	Burrichter	Wilkens	4:25:00,0
3	kein	Lamberty	Lemor	Sass	4:41:02,6
4	LT Störlauf	Steffens	Funk	Jürgens	4:41:50,9
5	Lauffreunde Itzehoe	Lassen	Wierauske	Eckhardt	4:47:24,5
6	kein	Grünbauer	Wend	Bade	4:54:05,4
7	LG Dithmarschen	Kreft	Hein	Petersen	4:54:28,3
8	Heidgrabener SV	Matthiesen	Olbrisch	Schenkies	4:55:04,9
9	LT Stoerlauf	Papert	Bohnsack	Djurovac	4:55:17,3

How did we do this?

Fields/Columns

Column Heading	Data	Alignment	Line	B	I	U
Rank EN	TS1.Rank		1	B	I	U
Club EN	Club		1	B	I	U
Rider1 EN	TS1.P1.Lastname		1	B	I	U
Rider2 EN	TS1.P2.Lastname		1	B	I	U
Rider3 EN	TS1.P3.Lastname		1	B	I	U
Time EN	TS1.Time1		1	B	I	U
EN			1	B	I	U

Filter

Maximum Number of Records:

Conjunction	Field	Operator	Value
	TS1.Rank	>	0
and	TS1.Position	=	1

We use TS1.Px to access the three names of the athletes and show them in one line. Without any other modification the report would look like this:

Rank	Club	Rider1	Rider2	Rider3	Time
Half Marathon					
1	LG Itzehoe	Plambeck	Sievers	Banasch	4:22:24,2
1	LG Itzehoe	Plambeck	Sievers	Banasch	4:22:24,2
1	LG Itzehoe	Plambeck	Sievers	Banasch	4:22:24,2
2	Sparkasse in Steinburg	Burrichter	Burrichter	Wilkens	4:25:00,0
2	Sparkasse in Steinburg	Burrichter	Burrichter	Wilkens	4:25:00,0
2	Sparkasse in Steinburg	Burrichter	Burrichter	Wilkens	4:25:00,0
3	kein	Lamberty	Lemor	Sass	4:41:02,6
3	kein	Lamberty	Lemor	Sass	4:41:02,6
3	kein	Lamberty	Lemor	Sass	4:41:02,6
4	LT Störlauf	Steffens	Funk	Jürgens	4:41:50,9
4	LT Störlauf	Steffens	Funk	Jürgens	4:41:50,9
4	LT Störlauf	Steffens	Funk	Jürgens	4:41:50,9

It would still show one line per participant. So we have to add the filter TS1.Position = 1. TS1.Position is the internal numbering of the team (1 => first team member, 2 => second team member, …), so the filter makes sure that only one record per team will be shown.

7.6. Team Lap Race: Additional Fields

When using a team score with lap mode (see chapter 3.10) for example for a 24h mountain bike race, you may need information about the participant who did the 25th lap of the team. For this you can use TSx.Ly.., For example

```
TS1.L25.Lastname
```

returns the last name for the athlete that did the 25th lap of the team according to team score 1. After *TS1.L25.* any field can be used.

7.7. Team Gap Times

For team scores, there is no function like *GapTimeTop*, so we have to calculate gap times manually. Therefore the fields

```
TSx.DecimalTimeTopY and TSx.DecimalTimePrevY
```

can be used which return the yth team time of the first team / previous team according to team score X. A possible expression for the team gap time would thus be:

```
iif([TS1.Rank]=1; "-"; format([TS1.DecimalTime1]-[TS1.DecimalTimeTop1]; "+mm:ss"))
```

7.8. Aggregation Functions

First of all note that the following aggregation functions are not very fast and should be used rarely.

Assume you would like to show the number of finishers on a race certificate. For this you can use the DCount() functions which has a filter as first and only parameter:

```
DCount("[Finished]")
```

If you would like to show the number of finishers that have the same first name you can use this expression:

```
DCount("[Finished] AND [FirstName]='" & [FirstName] & "'")
```

Besides DCount, there are the functions DSum, DMin, DMax, DAvg that calculate the sum/minimum/maximum/average of a field, and DFirst that returns a field of the first participant fulfilling a expression. If you would like to show the average age of all finishers, you can use this expression:

```
DAvg("[Age]"; "[Finished]")
```

Why the double quotes around the field names? Because the parameter is the name (instead of the content) of the field that the function shall analyze!

8. FREQUENTLY ASKED ADVANCED QUESTIONS

How to overwrite a formula result manually?

The answer is: you can't. But you can create a result for the manual time. Assume your formula result is result 11. Change the ID to 21 and add another result "Manual Time" with ID 22. As result 11, you can create a formula result with formula

```
iif(T22>0; T22; T21)
```

11	ManualOrFormulaResult		no rounding	iif(T22>0;T22;T1)		
21	FormulaResult		no rounding	[your formula]		
22	Manual		no rounding			

Speed or Time per km

Make sure to enter the contest length in meters in Main Window -> Contests. Then you can use these expressions:

```
format(T1 / ([ContestLength]/1000); "mm:ss") & " min/km"
```

```
format(3.6*[ContestLength]/T1; "s.k") & " km/h"
```

Relay numbers

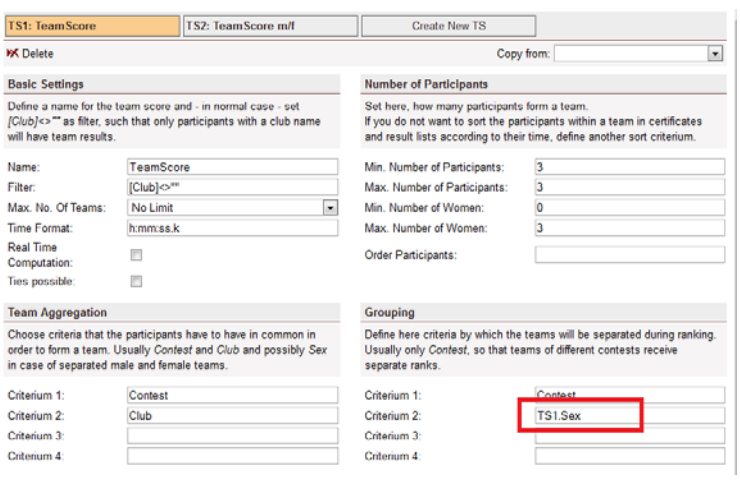
The bib number in race result has to be numerical. If you have relay numbers like 11-A, 11-B, 11-C, 12-A, 12-B, 12-C, .. you can proceed as follows:

- Use bib 111 for 11-A, 112 for 11-B, 113 for 11-C, 121 for 12-A, 122 for 12-B, and so on
- Create a user defined field *RelayBib* (see chapter 4.4) using this expression:
  `([Bib]\10) & "-" & choose([Bib]%10; "A"; "B"; "C")`
- Use *RelayBib* instead of *Bib* on all lists / certificates.

Team Score with Automatic Male/Female/Coed Detection

Teams can be male, female or coed. These three categories will be scored separately, i.e. there is a first place for the male teams, a first place for the female teams and a first place for the coed teams. The field TSx.Sex automatically determines if the team is male (TSx.Sex=1), female (TSx.Sex=2) or coed (TSx.Sex=3).

This field can then be used to group the team score:



You can group your team results report for example by this expression:

```
choose([TS1.Sex]; "Male Teams"; "Female Teams"; "Coed Teams")
```