

# **JRaceman Users Guide**

**by Jim McBeath**



# **JRaceman Users Guide**

by Jim McBeath

Copyright © 2002-2007 Jim McBeath

A Free Meet/Race/Regatta Management Program

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License, Version 2 of June 1991 and no other version, the same license as is used for the JRaceman software. A copy of the license is included in the appendix titled "Software License". The GPL is also available on-line [<http://www.gnu.org/copyleft/gpl.html>].







# Table of Contents

## How To Use This Guide

### Part I. Overview

#### Chapter 1. Introduction

1.1. Features 11

1.2. Goals 11

#### Chapter 2. Key Concepts

2.1. Definitions 13

2.2. The Database 14

2.2.1. Relational Database 14

2.2.2. Schema 14

2.2.3. ID Fields 15

2.3. Progressions 16

2.4. Scoring 16

#### Chapter 3. Managing JRaceman

3.1. System Requirements 17

3.2. Installing Java 17

3.3. Installing JRaceman 17

3.4. Unpacking the JRaceman Source 18

3.5. Running JRaceman 18

3.5.1. On Windows 18

3.5.2. On Unix 19

3.6. Upgrading JRaceman 19

3.7. Client/Server 20

3.8. Username and Password 20

### Part II. Using JRaceman

#### Chapter 4. Preparing for a Meet

4.1. Creating a New Database 22

4.2. Entering Meet Data 22

4.3. Creating a Schedule 22

4.4. Entering Teams and People 23

4.5. Entering Registrations 23

4.6. Entering Entries 23

4.7. Assigning Lanes 23

4.8. Scratch Meeting 23

4.9. Printing Lane Sheets 24

#### Chapter 5. Running a Meet

5.1. Handling Scratches 25

5.2. Entering Race Results 25

5.3. Managing Progressions 26

5.4. Generating Scores and Awards 26

5.5. Generating Web Reports 26

#### Chapter 6. Producing Reports

6.1. Report Format 27

6.2. Web Reports Directory 27

#### Chapter 7. Managing the Database

7.1. Creating Backups 28

7.2. Checking Your Data 28

7.3. Sharing Data by Import and Export 28

#### Chapter 8. Web Registration

8.1. Set Up 29

### Part III. User Interface

#### Chapter 9. Main Window

9.1. Input Tabs 31

9.2. Separator Bar 31



9.3. Output Tabs	31
9.4. Status Line	31
Chapter 10. User Interface Elements	
10.1. Terms	32
10.2. Tool Tips and Tool Prompts	32
10.3. Using Menus	32
10.4. Using Tabs	33
10.5. Choice Lists	33
10.6. Scroll Bars	33
10.7. Split Panes and Separator Bars	34
10.8. Tables	34
10.9. Sheets	35
10.9.1. Function Keys	35
Chapter 11. Menus	
11.1. The File Menu	37
11.2. The Database Menu	37
11.3. The Tabs Menu	37
11.4. The Query Results Menu	37
11.5. The Report Menu	38
11.6. The Help Menu	38
Chapter 12. Batch Mode	
Chapter 13. Top Level Input Tabs	
13.1. By Event	41
13.1.1. Selecting an Event	41
13.1.2. Tasks	41
13.2. Database	42
13.3. Debug	42
13.4. Meet Setup	43
13.5. Plan Setup	43
13.6. Reports	43
13.7. Site Setup	44
13.8. Sport Setup	44
13.9. Team Setup	44
Chapter 14. Input Tabs	
14.1. Areas	46
14.1.1. Query and Edit Tabs	46
14.1.2. Table	46
14.2. Awards	46
14.3. Backup	47
14.3.1. Creating Backups	47
14.3.2. The Quick Backup Button	48
14.3.3. Recovering A Backup	48
14.4. Challenges	49
14.4.1. Query and Edit Tabs	49
14.4.2. Table	49
14.5. Checks	49
14.6. Competitions	50
14.6.1. Query and Edit Tabs	50
14.6.2. Export	51
14.6.3. Table	51
14.7. ComplanRules	51
14.7.1. Query and Edit Tabs	51
14.7.2. Table	52
14.8. Complains	52
14.8.1. Query and Edit Tabs	53
14.8.2. Export	53
14.8.3. Table	53
14.9. ComplanStages	53



14.9.1. Query and Edit Tabs	54
14.9.2. Table	54
14.10. ContextOptions	54
14.10.1. Query and Edit Tabs	54
14.10.2. Available Options	55
14.10.3. Table	55
14.11. Debug Change ID	56
14.12. Debug EditField	56
14.13. Debug ForeignKeys	56
14.14. Debug Generate	57
14.15. Debug Properties	57
14.16. Debug SQL	57
14.16.1. Buttons	57
14.16.2. Commands	58
14.17. Entries	58
14.17.1. Query and Edit Tabs	58
14.17.2. Sheet	59
14.17.3. Person Entries	59
14.17.4. Table	60
14.18. Events	60
14.18.1. Query and Edit Tabs	60
14.18.2. Sheet	61
14.18.3. Table	61
14.19. Exceptions	62
14.19.1. Query and Edit Tabs	62
14.19.2. Table	62
14.20. Finish	63
14.20.1. Editing Results	63
14.20.2. Data Entry Options	63
14.20.3. Enter Places By Lane	64
14.20.4. Reports	64
14.20.5. Table	65
14.21. Genders	65
14.21.1. Query and Edit Tabs	65
14.21.2. Table	66
14.22. LaneOrder	66
14.22.1. Query and Edit Tabs	66
14.22.2. Sheet	66
14.22.3. Table	66
14.23. Lanes	67
14.23.1. Query and Edit Tabs	67
14.23.2. Sheet	67
14.23.3. Table	68
14.24. Levels	68
14.24.1. Query and Edit Tabs	69
14.24.2. Table	69
14.25. Meets	69
14.25.1. Query and Edit Tabs	69
14.25.2. Export	70
14.25.3. Table	71
14.26. Options	71
14.26.1. Editing Options	71
14.26.2. Options	71
14.26.3. Table	72
14.27. People	72
14.27.1. Query and Edit Tabs	72
14.27.2. Sheet	73
14.27.3. Table	74



14.28. Progressions	74	
14.28.1. Query and Edit Tabs	74	
14.28.2. Available Progressions	75	
14.28.3. Table	77	
14.29. Progress	77	
14.29.1. Automatic or Manual Progression	78	
14.29.2. Automatic Progressions	78	
14.29.3. The Auto Progress Action Buttons	78	
14.29.4. Manual Progressions	79	
14.29.5. Table	79	
14.30. Races	79	
14.30.1. Query and Edit Tabs	79	
14.30.2. Sheet	80	
14.30.3. Table	80	
14.31. Registration Fees	80	
14.31.1. Query and Edit Tabs	81	
14.31.2. Sheet	81	
14.31.3. Table	81	
14.32. Registrations	81	
14.32.1. Query and Edit Tabs	82	
14.32.2. Sheet	82	
14.32.3. Table	82	
14.33. Reports.General	83	
14.33.1. Format Choices	83	
14.33.2. Selection Choices	83	
14.33.3. Report Styles	84	
14.34. Schedule	85	
14.34.1. Creating Races	85	
14.34.2. Adding Special Items	86	
14.34.3. Sheet	87	
14.34.4. Using Auto-Increment	87	
14.34.5. Creating a Report	88	
14.34.6. Table	88	
14.35. Score	88	
14.35.1. Reports	88	
14.35.2. Table	89	
14.36. Scoring	89	
14.36.1. Systems Sheet	89	
14.36.2. Query and Edit Tabs	89	
14.36.3. Sheet	90	
14.36.4. Table	90	
14.37. Seeding Lists	90	
14.37.1. Query and Edit Tabs	90	
14.37.2. Sheet	91	
14.37.3. Table	91	
14.38. Seeding Plans	91	
14.38.1. Query and Edit Tabs	91	
14.38.2. Sheet	92	
14.38.3. Table	92	
14.39. SimplanRules	92	
14.39.1. Query and Edit Tabs	92	
14.39.2. Table	92	
14.40. Simplans	93	
14.40.1. Query and Edit Tabs	93	
14.40.2. Export	93	
14.40.3. Table	94	
14.41. SimplanStages	94	
14.41.1. Query and Edit Tabs	94	



14.41.2. Table	95
14.42. Sites	95
14.42.1. Query and Edit Tabs	95
14.42.2. Table	95
14.43. Stages	96
14.43.1. Query and Edit Tabs	96
14.43.2. Table	96
14.44. Teams	96
14.44.1. Query and Edit Tabs	97
14.44.2. Sheet	97
14.44.3. Export	98
14.44.4. Table	98
14.45. Transfer	98
14.45.1. Sending Data	99
14.45.2. Receiving Data	99
Chapter 15. Output Tabs	
15.1. Help	100
15.1.1. The Help Tab	100
15.1.2. The Help Menu	100
15.1.3. Help File Navigation	100
15.2. Messages	101
15.3. Query Results	101
15.4. Report	101
Part IV. Appendices	
Appendix A. Software License	
A.1. GNU General Public License	103
Appendix B. Debugging and SQL	
B.1. SQL Primer	108
Appendix C. Mckoi Database	
Appendix D. Index	



# How To Use This Guide

Welcome to JRaceman. You will probably want to read different parts of this Guide depending on how much experience you have with JRaceman and what you are trying to do.

If you are reading this online in JRaceman, you can always get back to this page by selecting the Welcome command from the Help menu.

- If you have not yet used JRaceman, or this is the first time you have used it, you should read (or at least skim) the Overview part so that you understand what JRaceman is about.
- If you are trying to do something in JRaceman for the first time, look in the Using JRaceman part for a chapter or section that looks related to what you want to do. This part is intended to answer questions of the form “How do I do ... ?”.
- If you want to learn in more detail how a specific part of the interface works, see the User Interface part. This part is intended to answer questions of the form “How do I use ... ?”.
- If you are reading this online from JRaceman and want more information on how to use the Help tab, see the Help section, also accessible with the Help On Help command in the Help menu.

More information can be found in these documents:

- JRaceman README [[../README.html](#)]
- JRaceman Tutorial [[../tutorial/index.html](#)]
- JRaceman API Reference [[../api/index.html](#)] (if you have unpacked the source [[../README.html#source](#)])



# Part I. Overview

This Part describes what JRaceman is, defines the key concepts used throughout the program, and explains some of the underpinnings. You should at least skim this Part when first using JRaceman, and read it in more detail when you want to gain a better understanding of JRaceman's capabilities.



# Chapter 1. Introduction

JRaceman is a meet and race (as in track race) or regatta manager written in Java and distributed under the GNU General Public License (GPL).

For the latest updates to JRaceman, visit the JRaceman home page [<http://www.alumni.caltech.edu/~jimmc/jraceman>].

## 1.1. Features

- Licensed under GPL, freely redistributable with source.
- 100% Java, runs on Windows, Mac and Linux/Unix.
- Self-contained relational database (Mckoi Database [<doc/userguide/Mckoi.html>]).
- Runs in standalone (single-user) mode, or in a client/server configuration to allow multiple clients to access the database simultaneously.
- Checks data for errors and restrictions (such as max entries per person).
- Automatic Progression (lane assignment).
- Supports non-scoring competitors (such as Internationals in a National competition).
- Supports group events (such as relay races or team boats).
- Calculates Individual and Team scores.
- Supports multiple customized per-place point scoring systems.
- Produces HTML reports: Schedule Reports, Entries Reports, Lane Reports, Results Reports, Progress Reports, Score Reports, Award Reports.
- Fast web reports to maintain a web site during a meet.
- Prints labels for awards.
- Export/Import capability to support distributed data entry.
- Interfaces to FinishLynx [<http://www.finishlynx.com>] and Omega automated finish-line systems.
- Integrated on-line web registration module.
- On-line help with built-in browser.
- Includes support for USACK Sprint Canoe/Kayak competitions, progressions, scoring and awards.

## 1.2. Goals

The goals of JRaceman are:

1. to be easy to use;



2. to run on Windows, Mac, Unix and Linux;
3. to be available for free, not requiring the purchase of any software;
4. to be freely redistributable so that anyone can fix bugs or make improvements for the benefit of all.

To satisfy these goals, Jraceman:

1. is designed with a hierarchical set of tabs, has built-in on-line help, includes a tutorial wizard, and comes with documentation that can easily be printed (PDF) or viewed on-line (HTML);
2. is implemented in 100% Java, so will run on any machine that can run Java;
3. is distributed without charge, includes a free database, and does not require any additional packages or programs to run (other than Java, which is free);
4. is distributed under the GNU General Public License (GPL) to ensure that it remains free and to encourage users to contribute their improvements.



## Chapter 2. Key Concepts

This chapter describes some of the key concepts you need to understand in order to effectively use JRaceman.

You should read the Definitions section to make sure you understand the terminology used by Jraceman; this is essential because JRaceman uses these terms extensively.

You may defer reading the other sections in this chapter until later; they present a few more concepts and are useful for gaining a better understanding of how JRaceman works, but are not essential for your initial use of JRaceman.

### 2.1. Definitions

JRaceman deals with eleven main types of data divided into two groups. There are eight data types relating to the events you manipulate, and three data types which cover the participants in events.

The eight main Event Data Types are:

- A Meet is a meet, regatta or other event at which competitors collect in order to compete in different ways. A Meet is a collection of Events.
- A Competition is one sport within a Meet. For example, at a track meet, High Jump and 100-meter run are each a Competition.
- An Event is a Competition qualified by Level (typically age) and Gender. At a track meet, 100-meter run for Boys 12-13 could be an event, and 100-meter run for Girls 12-13 would be a different event.
- A Stage is the type of a Round. Typical Stage names are Heat, Semifinal, and Final.
- A Round is a set of all Races of the same Stage within an Event, where the results of each Round are used to determine which competitors advance to the next Round. The advancement is controlled by the Progression which has been selected for that Event.
- A Section is one of the Races within a Round. Each Section of the Round is given a sequential number within the Round.
- A Race represents the actual running of the event. Each Race is within one Round of an Event. The last Race of an Event is typically a Final, and the winners of that Race earn points or awards. A Race has a Round number and a Section number and is associated with an Event.
- A Lane is one slot in a race. In races where the competitors are individuals, a Lane corresponds to the use of one lane or starting position for one Race. In group races, such as relay races, there are multiple Lanes (one for each person) which taken together correspond to the use of one lane or starting position for one Race.

The three main Participant Data Types are:

- A Team is a collections of People competing together. In addition to being a means of identifying people, points earned in events are summed for all members of a team.
- People are the individuals who make up a Team.
- A Group is a collection of People competing together in an Event, such as the four people in a four-man relay race, or the four people in a team boat in a regatta. The members of a Group are usually on the same Team, but not always.



## 2.2. The Database

This section describes the database as used by JRaceman.

### 2.2.1. Relational Database

JRaceman uses a relational database (the Mckoi database) to store all of its data. A relational database contains a collection of tables. Each table has rows and columns. Each column has a name and a data type. Each row represents one data record. The intersection of a row and a column is a field, which is one value within one record.

Within each table, one or more columns are defined as primary key columns. The value of the primary key fields of each row in a table must be unique within that table. This is called the primary key constraint. All primary key fields must not be null. A database that maintains this constraint ensures entity integrity. In JRaceman, all of the tables use a single-column primary key called "id". Thus, within one table, all of the "id" values must be unique.

A table may also have columns with values that relate to the values in other tables (hence the name relational database). In the simplest case, a column of one table contains only values which are taken from the set of values in a column of another table. This column is then called a foreign key column. In JRaceman, many of the tables have foreign key columns pointing to the "id" column of another table. For example, the People table has a column called "teamId" which is a foreign key referring to the "id" column of the Teams table. The value in the "teamId" column of the People table must be one of the values from the "id" column of the Teams table; the database will not allow it to be any other value (other than blank). Having the database apply this constraint on allowable values (making sure that all of the non-blank foreign keys refer to a valid row in the referenced table) ensures referential integrity.

### 2.2.2. Schema

The collection of tables, with the column definitions and primary and foreign key constraints, is called a schema.

The schema in JRaceman consists of the following tables, each of which is documented in more detail with the editing tab for that table.

- Areas - Defines the areas with a Site which are used for Events.
- Challenges - Defines the Challenge category names.
- Competitions - Defines the types of Events.
- Complains, ComplanRules and ComplanStages - Defines complex progression plans available for a Meet.
- ContextOptions - Options which are applied based on context-specific values for host, web context, or Meet.
- Entries - Entries of People into Events, with information about Groups.
- Events - Defines the Events to be contested at a Meet.
- Exceptions - Defines the reasons why contestants can be disqualified from an Event.
- Genders - Defines the names to use for gender-specific Events.
- LaneOrder - Defines the lane ordering preference.
- Lanes - Defines which Entries are in which Races, plus lane assignments, finishing times and positions, and other lane-specific data.



- Levels - Defines the different levels of competition, typically age groups.
- Meets - Defines the meets or regattas at which Events take place.
- Options - Generic options.
- People - Defines the people competing in Events.
- Progressions - Defines the progressions that determine how competitors advance from one round of a race to the next.
- Races - Defines the difference race that make up an Event.
- RegistrationFees - Registration rates for Meets based on number of Events entered.
- Registrations - Registrations of People for a Meet, with information about payment and signed waivers.
- ScoringSystems and ScoringRules - Defines the point scoring systems available to use for a Meet.
- SeedingPlans - Defines the seeding plans available for seeding the lane assignments when drawing lanes for an Event.
- SeedingLists - Defines the seeding lists associated with seeding plans, allowing specifying a list of entrants to be seeded in an Event.
- Simplans, SimplanRules and SimplanStages - Defines the simple progression plans available for a Meet.
- Sites - Defines the sites at which Meets take place.
- Stages - Defines the stages, such as Heat or Final, for the difference rounds of an Event.
- Teams - Defines the Teams of People competing in Events.

### 2.2.3. ID Fields

In JRaceman, every table has a single-column primary key called "id". As with any key in a relational database, the value of this field must be unique among all rows within the table.

When creating a new record in any editing tab, the ID field is editable, allowing you to fill in a value if you desire. When editing an existing record, however, the ID field is not editable.

Normally you do not need to enter a value for the ID in a new record; the program automatically generates one for you based on the type of record, and sometimes based on the data in other fields of that record.

In most cases, the tab supplies an ID prefix, which is a string such as "ID". The program takes this ID prefix and adds a number to it to create a unique ID value such as "ID1". As each new record is created, the next number is used with the ID prefix, generating a sequence such as "ID1", "ID2", "ID3", etc. If a record is deleted, that ID will be reused later.

Some tabs use a different technique to generate a unique ID, as documented for those tabs. In these cases, the program may silently modify that unique ID by adding a numeric suffix or by changing the numeric suffix so that the ID string is unique.

If you choose to enter an ID value for a new record, the program will check to see if your ID value is unique when you save the record. If it is not, the program will automatically add a numeric suffix, or modify an existing numeric suffix, if given, to produce a unique ID. In this case, you will be asked to accept this new ID before the record is saved.



## 2.3. Progressions

Progression refers to the manner in which the winners of early rounds in a race move on to the later rounds. For example, in an Event with Heats, Semi-Finals, and Finals, the Progression for the Event may specify that the first-place winners of each Heat go directly to the Final round, the second-place through sixth-place winners of the Heats go to the Semi-Finals, and the seventh-place and later finishers of the Heats are out of the Event. The same Progression would specify which finishers in the Semi-Final round move on the Final round. The Progression also specifies the lane assignments for the finishers who are advancing.

The set of rules which apply to a specific Event is referred to as the Progression Plan for that Event. A set of Progressions Plans taken together form a system of Progressions. For example, one Plan may determine how finishers advance from Heats through Semi-Finals to Finals when there are two Heats; another Plan in the same system may determine how finishers advance when there are three Heats; and another may determine it when there are four Heats.

If an Event has only one round, then there is no Progression for that Event.

JRaceman is capable of automatic or manual Progressions. In an automatic Progression, you specify the Progression system to use for an Event. JRaceman picks the Plan to use based on the number of Entries in that Event. After entering the data for a Round, a single button-press tells JRaceman to use the automatic Progression to do the lane assignments for the following Rounds.

When using manual progression, you must enter the lane assignments for each round yourself.

You can create a Custom automatic progression using Simplans or Complans. You can also use the manual progression tools to make changes to the lane assignments created by an automatic or custom progression.

## 2.4. Scoring

JRaceman is set up to calculate scores for a meet. Scoring is based on the finishing place in the final round of an Event. Points are awarded to finishers in each Event, then added up based on Team to get totals for each team.



## Chapter 3. Managing JRaceman

This chapter describes the administrative details of setting up and maintaining JRaceman. Much of this information is also presented in the README.html [../README.html] file.

### 3.1. System Requirements

Item	Required	Why
Processor	300 MHz	Starting JRaceman will take about one minute on a 300MHz processor. Faster processors will of course give better results.
Memory	64 MB	Graphical Java programs tend to use a lot of memory. With anything less than 64MB, your computer will run considerably slower due to swapping. 128MB or more is recommended.
Disk	20 MB, plus up to 250 MB to install Java	If you already have Java installed, installing JRaceman will take about 10MB, and creating the database will take about 5MB. Larger databases will require more space. Installing Java takes 100MB or more, depending on which version you install, and whether or not you install the documentation.
Java	JRE 1.5	JRaceman uses Swing and other features of Java which are not supported in earlier versions.
Operating System	Windows, Linux, or any other OS that supports Java 1.5	For a list of platforms on which Java is available, visit Sun's Java Platform Ports [ <a href="http://java.sun.com/cgi-bin/java-ports.cgi">http://java.sun.com/cgi-bin/java-ports.cgi</a> ] page.

### 3.2. Installing Java

JRaceman is a 100% Java program, and requires Java in order to run. To install Java on your system, download JRE 5.0 [[http://java.sun.com/javase/downloads/index\\_jdk5.jsp](http://java.sun.com/javase/downloads/index_jdk5.jsp)] (also known as 1.5) from Sun's Java web site [<http://java.sun.com/>] and follow those installation directions.

### 3.3. Installing JRaceman

JRaceman is delivered in a self-extracting JAR file. You must have Java running on your system in order to extract it.

To install JRaceman from the self-extracting JAR file (.jar extension), double-click on the JAR file, or run the command



```
java                                -jar                                jraceman-N_N_N.jar
```

This will open a dialog allowing you to specify a destination directory, and it will create a directory called `jraceman-N_N_N` in the specified directory, where *N\_N\_N* is replaced by the JRaceman version number.

To install the web registration module, see the Web Registration chapter.

## 3.4. Unpacking the JRaceman Source

If you want to build JRaceman yourself, or just look at the source code, you will have to unpack the source files from the `src.zip` archive file with the commands

```
cd                                jraceman-N_N_N;  
unzip                             src.zip
```

This will create a `src` directory with all of the Java files, as well as adding a few other files (such as a Makefile) to your `jraceman` directory.

Similarly, if you want to look at the Javadoc-generated API documentation for the Java classes, you must first unpack those html files from the `apidoc.zip` archive file with the commands

```
cd                                jraceman-N_N_N;  
unzip                             apidoc.zip
```

This will create the directory `doc/api` with all of the API documentation in it. Alternatively, you can unpack `src.zip` and then run the command

```
make                                doc
```

to run javadoc and create the API documents from the source files.

The JRaceman kit does not include the source to the Mckoi database. You can download the Mckoi database source from the Mckoi web site [<http://www.mckoi.com/database>], or from the same location from which you downloaded JRaceman. You do not need the Mckoi source to compile JRaceman; the `mckoidb.jar` file included in the JRaceman kit is sufficient.

The JRaceman kit does not include the source to the `jRegistryKey` package. You can download the `jRegistryKey` source from the `jRegistryKey` web site [[http://www.beq.ca/tech/Products/jreg\\_key.shtml](http://www.beq.ca/tech/Products/jreg_key.shtml)], or from the same location from which you downloaded JRaceman. You do not need the `jRegistryKey` source to compile JRaceman; the `jRegistryKey.jar` file included in the JRaceman kit is sufficient.

## 3.5. Running JRaceman

If you have not already done so, you must install Java.

### 3.5.1. On Windows

On Windows, CD to the main JRaceman directory and run

```
java                                -jar                                jraceman.jar
```

or set up a shortcut to `jraceman.jar` in the main JRaceman directory, with the Start-In directory of the shortcut set to



the JRaceman directory.

JRaceman uses a DLL on Windows to access the registry (`jRegistryKey.dll`). The DLL is in the main JRaceman directory; either this directory must be in your PATH, or your working directory must be the JRaceman directory. Assuming you don't change your PATH to include the JRaceman directory, then if you are running from the command line, you must CD to the JRaceman directory before running `jraceman.jar`; if you are setting up a shortcut, you must set the Start-In directory (or working directory) of the shortcut to the JRaceman directory.

### 3.5.2. On Unix

On Unix, run

```
java -jar jraceman.jar
```

If the java binary is not in your PATH, you can explicitly specify the path to it. For example, if you have installed JDK1.3 into `/usr/local/jdk1.3`, you can run `jraceman` with the command

```
/usr/local/jdk1.3/bin/java -jar jraceman.jar
```

On Solaris 2.6 or later, you can directly run `jraceman.jar`.

## 3.6. Upgrading JRaceman

When you install a new version of JRaceman, you may need to upgrade your database. The new version of JRaceman may use new tables or columns, or, less likely, the format of the underlying Mckoi database may have changed.

Before upgrading to a newer version of JRaceman, be sure to back up your database with the old version in case there are database format changes. In particular, you should create an export file from your database (using the Export command in the Database menu) which can be used to import your data into a new database if required.

When you first run a new version of JRaceman, it checks to see if the database needs to be upgraded. If so, it opens a dialog asking you if you want to upgrade. Typically you will have to upgrade in order to run the new version of JRaceman.

You can force JRaceman to run an upgrade. After backing up your database and installing the new version of JRaceman, run this command:

```
java -jar jraceman.jar -db database -upgrade
```

This will add all necessary new tables and columns to your database. If you want to see what new tables and columns will be added without actually adding them, use the **-checkupgrade** command line option rather than the **-upgrade** command line option.

With an older database, you may receive a message something like this:

```
The state store for this database doesn't exist. This means the database version is pre version 1.0.
Please see the README for the details for converting this database.
```

This message comes from the Mckoi database engine, and indicates that the underlying Mckoi database format has changed. The README file referred to in that message is the Mckoi README file, which tells how to perform the Mckoi database upgrade. Rather than doing this, you should probably export from your old database and import into a newly created database.

If the underlying Mckoi database format has changed, the simplest upgrade path is to create an export file with the



old version of JRaceman (using the Export command in the Database menu), then create a new empty database with the new version of JRaceman, and finally import the previously created export file into the new database with the new version of JRaceman (using the Import command in the Database menu).

Some versions of JRaceman may have special upgrade instructions, which will be in the HISTORY file.

## 3.7. Client/Server

For smaller races, the simplest way to run JRaceman is to use the default database with a single user. This is how JRaceman runs if you do not specify any special command line options.

For larger races, you may want to allow multiple people to access the database simultaneously. In this situation, you can run a JRaceman database server, after which multiple clients can run and connect to that server.

To start a JRaceman server, run JRaceman on your server machine with this command line:

```
java -jar jraceman.jar -db database -runserver
```

This will start a JRaceman database server listening on TCP port 9157. After the server starts, you can start one or more JRaceman clients this this command:

```
java -jar jraceman.jar -dbhost servername
```

where servername is the name of the machine on which the JRaceman database server is running. When you want to shut down the server, first exit from all of the clients and then run this command:

```
java -jar jraceman.jar -dbhost servername -sql SHUTDOWN
```

If, when installing JRaceman on Windows, you left the "Install Shortcuts on Desktop" selected checked, there will be shortcuts to start the server, stop the server, start the client, and start in standalone mode. You can look at the Properties on these shortcuts to see how they work. For a multi-user system, copy the start-client shortcut and change the Properties to pass in the servername of the JRaceman host computer.

## 3.8. Username and Password

By default, JRaceman uses a built-in username of "jraceman" and password of "jraceman". For single-user systems and trusted networks, this is usually acceptable. For increased security, particularly when running a JRaceman database server, you can specify another user and password. To do so, use the -user and the -password options the first time you run JRaceman for a new database:

```
java -jar jraceman.jar -db database -user username -password password
```

After this, when running a client (or when running in single-user mode), you can either specify the user (and optionally the password) on the command line in the same manner, or you can simply start JRaceman and fill in the user name and password when it presents you with the dialog for that purpose.



## Part II. Using JRaceman

This Part is intended to help you learn how to use JRaceman. It is organized by task to allow you to find out how to use JRaceman to do a specific job. You should look in this Part when you know what you want to do, but you are not sure what part of JRaceman to use to do it.



## Chapter 4. Preparing for a Meet

This chapter describes how to use JRaceman to prepare for a meet in advance.

### 4.1. Creating a New Database

The first time you use JRaceman for a meet, you must create a new database. If you start JRaceman with no command line options, it will look for a database in its default location. Alternatively, you can use the **-db *database-path*** command line option to specify the location of your database. In either case, if the database does not exist JRaceman will ask you if you want to create it. Say yes, and your new database will be created with all necessary tables but no data.

At some point you should visit the Backup tab in the Database tab to set up a backup directory. Once this has been set up, you can easily create a backup snapshot of the database by pressing a single button in the Backup tab, or by pressing the Backup button in the menu bar.

After creating a new database, you must set up the Sport definitions. The easiest way to do this is to load a standard set of sport data. JRaceman includes a sport definition file for USACK sprint canoe/kayak races. To load this file, open the Database menu and select the Import JRaceman Data File command. Navigate to the directory in which you installed JRaceman, then from there down into the data directory. Select the file called `usack-sports.txt` and press the Open button. This will import the USACK sport definition file.

If you already have a database, you may want to add your meet to that database. This has the advantage that much of the data may already be there, such as the Site and Team data. Or you can dump some or all of the data from your old database into a text file, possibly edit the text file to leave only the data you want to transfer, then import it into your new database.

### 4.2. Entering Meet Data

After you have created your database you can start entering data about your Meet. First you must define the Site at which your Meet will be held, and possibly the Areas within that site. This is done in the Site Setup tab.

If your race course is non-standard, or for some reason you want lanes to be assigned in some other order than starting at 1 and counting up, you can specify the order of lane assignment (for initial lane draws and for Simplan and Complan progressions) using the Lane Order tab in the Site Setup tab.

After the Site is defined, you can start on the Meet. Use the Meets tab to define the Meet, then use the Events tab to enter all of your Events. Remember that an Event consists of multiple Races.

### 4.3. Creating a Schedule

After your Events are defined you can start working on your race schedule. In order to do this, you must have some idea of how many people are entering each event in order to determine how many races there will be. You can estimate this number for now in order to create your preliminary schedule. Later, after you have the Entries data and thus have an exact count, you can easily go back and modify the schedule if necessary.

Select the Schedule tab in the Meet Setup tab to work on your schedule. Using that tab you can select one Event at a time, enter the estimated number of Entries for that Event, and create the Races for that many entries. Once all of the Races are created, you can assign numbers to each Race, order them by race number, and assign a date and time for each Race. For details on how to do this, see the section for the Schedule tab.



## 4.4. Entering Teams and People

Before you can enter data for Entries (which tell who is racing in each Event), you must enter data for each Team and each Person who is competing. You do this using the tabs in the Team Setup tab.

## 4.5. Entering Registrations

After you have entered the People, you can use JRaceman to register them for a specific meet by using the Registration tab. This optional step allows you to see who has not yet paid or not yet turned in a signed waiver form. If these are not concerns for your meet, or if you are using another system rather than JRaceman to manage this data, then you can ignore registration.

If you are using web registration, you can have JRaceman automatically calculate the registration fees based on the number of entries for a person. To set this up, you must enter data into the Registration Fees tab.

## 4.6. Entering Entries

After you have entered the Teams and People you can enter the Entries data. The Entries tab is in the Meet Setup tab.

There is one Entries record for each combination of Person and Event. For Group events, such as team boats or relay races, there is still a separate Entry for each person in that group. The Entries within a Group are identified by setting the same Group value on all of those Entries for that Event. The Group values themselves are arbitrary, and can be reused for multiple Events.

After you have entered all of the Entries, you may need (or wish) to go back to the Schedule tab to redo the number of Races in some Events, scratch some Events, or otherwise fix up the Schedule.

## 4.7. Assigning Lanes

You may wish to assign lanes before your scratch meeting. The easiest way to do this is to use the Draw First Round for Matching Events command from the Schedule menu in the Schedule tab. In order to do this, you must already have created the Races for the Events. You may want to do this using the Create Races for Matching Events command from the Schedule menu in the Schedule tab.

## 4.8. Scratch Meeting

You may wish to assign lanes and print out Lane reports for your scratch meeting. To do this, see the Printing Lane Sheets section below. When printing Lane Sheets and Schedules for the scratch meeting, you probably want to set the Preliminary toggle so that the reports include the PRELIMINARY watermark in the background.

After completing all of the previous sections, you have entered all of data for the Meet, Teams and People, Events and Entries. At the scratch meeting just before the meet, you may have to modify some of this data. You can make these changes using only the tabs you have already used, or you can use the By Event tab.

To add, delete, or change Entries for a particular Event, use the By Event tab, select the Event of interest, and select the Entries/Progress task. That screen allows you to add, delete, and change Entries for that Event, and to change the group, alternate, and lane assignments for each Entry.



There are two ways you can scratch an Entry, depending on whether or not you want the Entry to appear as a Scratch in the reports for the Race. You can delete the Entry, in which case it will not appear anywhere, or you can mark it as a Scratch, in which case it will appear in reports as a scratch. You might want the Entry to appear as a scratch if you have already done lane assignments and are not planning on reassigning; showing the scratch explains an otherwise-empty lane.

To delete an Entry, use the By Event tab as described above. To mark an Entry as scratched, select the Event in the By Event tab and select the Entries/Progress task. Click on the checkbox in the Scratched column for the Entry to be scratched.

You can also use the Entries tab to add, edit and delete Entries. In particular, if you will be changing a number of Entries for one person, The Person Entries tab in the Entries tab is an easy way to do that.

If adding or removing an Entry changes the number of Races required for that Event, you can make that change. After changing the Entries in the By Event tab, select the Create Races task. This will display the number of Entries for that Event. Press the Create Races button. If the new number of Races is less than the existing number of Races, some of the existing Races are deleted. If the new number of Races is greater than the existing number of Races, new Races are created. If Races were deleted, you need not do anything to your Race Schedule, it will just have a few holes in it where those Races used to be. If, however, new Races were created, you will need to add those new Races to your Schedule using the Schedule tab. In this case, you will probably find the Auto-Increment features of the Schedule tab useful.

## 4.9. Printing Lane Sheets

Once the scratch meeting has finished, you can print up the final schedule and the Lane Sheets. Use the Schedule tab to print up a schedule.

Before printing the Lane Sheets, you must make sure you have assigned lanes for all races. Note that assigning lanes is a separate step from creating the Races, which you have already done during the process of creating the schedule. You can manually assign lane numbers, or you can have JRaceman automatically assign lane numbers.

To automatically assign lane numbers for all Races, use the Schedule tab as described above. To assign lane numbers for one events, you can use the Schedule tab and enter that Event number into the selection box on the left side of the form, then use the Draw First Round command, or you can use the By Event tab, select the Entries/Progress task, and then press the Draw First Round button.

To manually assign lane numbers for the first round, the simplest approach is to use automatic lane assignment, after which you can manually change those lane assignments using the Entries/Progress task in the By Event tab.

Once you have completed your lane assignments, you are ready to print out the Lane Sheets. This is done in the Reports tab. You can print up all of the Lane Sheets at once by selecting your Meet on the right side of the tab and selecting the desired report format on the left side. Press the Generate button to create the report into the Report tab. You can then use one of the commands in the Report menu to print the report directly or to save it to a file.



## Chapter 5. Running a Meet

This chapter describes how to use JRaceman while the meet is in progress. Although things can move pretty quickly while the races are going on, once you have done the initial preparation as discussed in the previous chapter, doing the data entry for the race results is actually pretty easy.

By the start of the races, you should have already printed out all of your Lane Sheets for the first-round races. Thus your tasks on Race Day will be:

- Handle race-day scratches.
- Enter the results for races as they finish.
- Manage the progression of Events with multiple rounds.
- Generate score summaries and lists of award winners.

### 5.1. Handling Scratches

If you need to scratch an Event or a Race during a meet, you can do so by setting the "scratched" flag in the Event or Race. You should also set the comment for the scratched Event or Race so that it will appear on the schedule the next time you print it. To set these values for Events, use the Events tab. To set these values for Races, use the Schedule tab. When you reprint the revised schedule, you probably want to use the "Revised" annotation on it.

There are three ways to scratch a competitor from an Event, depending on when it happens. If the scratch happens before the lane assignments have been printed up and distributed, such as if it happens at the scratch meeting, you can simply delete the Entry for that person in that Event. You can do this from the Entries/Progress task in the Entries tab.

If the scratch happens after the lane assignments have been distributed but before the first race of the Event, you can set the "scratched" flag in the Entry for that competitor. The competitor's name will still be printed in the reports for that Event, but with a line through the name. You can do this from the Entries/Progress task in the Entries tab.

If a competitor is scratched from an Event after having raced in at least one Race in that Event, you should enter that as an exception in the Results task of the By Event tab. Select that tab, then select the Event and Race and select the Results task. For the Entry to be scratched, change the Exception value to SCR for scratched. The SCR indication will appear in the results column for that competitor in that Race only. Results of other Races are not affected.

### 5.2. Entering Race Results

The By Event tab provides the primary interface for managing the various steps in an Event. Select the desired Event, then select the appropriate task.

If you are manually entering results into JRaceman, use the By Event tab and the Results task. If you are using the FinishLynx automated finish-line system, use the Transfer tab; if you are using the Omega finish-line system, use the Results task in the By Event tab and select the Import Race Results command from the By Event menu. In either case you can see the results of the race in the ByEvent/Results tab, where you can sort by any column to confirm that you have the right data. If scores are not being auto-updated, press the Recalculate Scores button to calculate the scores for all lanes in the selected race.

After all results data for a race have been entered and the score has been calculated, you can print out a Results Sheet for the race from the Reports task of the By Event tab, or from the Reports tab. Select the "Results" report on the Selected Race and press the Generate button to generate a report into the Report tab. You can use the commands in the Report menu to print the report or to save it to a file.



## 5.3. Managing Progressions

For multi-round races, use the Entries/Progress task of the By Event tab to manage the progression to the next round. After the results data have been entered for all of the sections of the current round, the Progress button in the Progress task of the By Event tab will become enabled. Press this button to automatically progress finishers to the following rounds. To manually enter the progression for an Event, use the sheet in the Progress task of the By Event tab.

In order for automatic progression to work, you must have specified a Progression for the Event. This is normally done as part of the pre-meet preparation, but if you forgot earlier you can set it now using the Event tab.

After progressing to the next round, print the Lane Sheets for those Races using the Reports task in the By Event tab, or use the Reports tab, just as you did when printing up the original Lane Sheets.

## 5.4. Generating Scores and Awards

The last stage of the process is to generate scores and calculate awards. You can do this at any time, either while the races are running or after they are all done.

Use the Score tab to generate score reports. The report will show the current point total for all Events which have been completed and scored. Choose the desired report format and press the Generate button to generate a report into the Report tab. You can use the commands in the Report menu to print the report or to save it to a file.

Use the "Awards" report format in the Reports task or the Reports tab to generate a report that lists all of the people who are to receive medals for an Event. Use the Award Labels tab to generate labels for awards for those people. You can generate these reports and labels as time permits after each completed Event. The Award Labels tab includes a Blank Label Count field that allows you to print additional labels onto a sheet which has been partially used already.

## 5.5. Generating Web Reports

Don't forget to make your results available on the Web! You can easily generate HTML files for your web site by using the Web reports. You can wait until the end to do this, or you can do it while the Meet is in progress in order to provide up-to-the-minute results on your web site. See the chapter on Web Reports for details.



## Chapter 6. Producing Reports

Reports are produced by pressing the appropriate buttons in various tabs. You can quickly get to any tab with a report button by clicking on Report menu item in the Tabs menu. Selecting any of the displayed tab names will take you to that tab. The buttons to generate reports are always labeled with text that includes either the word **Report** or the word **Generate**.

### 6.1. Report Format

JRaceman produces reports in HTML format. You can view these reports directly in the Reports tab in JRaceman, or you can save the report to a file and view it with your favorite browser. To save a report, use the Save HTML As command in the Report menu.

The HTML produced by JRaceman includes some special markers which are used by JRaceman to control printing. These markers are ignored by standard browsers, so the results of printing a page from a standard browser may not be the same as printing the original report directly from JRaceman. To print a report directly from JRaceman, use the Print command from the Report menu.

### 6.2. Web Reports Directory

JRaceman includes commands to make it easy to maintain a web site with up-to-the-minute race results. To use this capability, you must set up a Web Reports Directory. The Web Reports Directory is specified as one of the fields in a Meet record.

Once you have set up the Web Reports Directory, you can use Web Reports. Using Web Reports is easy. All you have to do is select a Web Report and press the Generate button, everything else is automatic:

- Web Report files are written directly into the Web Reports Directory hierarchy.
- Index files are automatically generated and maintained as you add or regenerate reports.
- A top-level index file `index.html` is maintained in the top Web Reports directory, from which a user can navigate to all of the information in the Web Reports directory hierarchy.

You can find the tabs with Web Reports by clicking on the Report command from the Tabs menu. All of the tabs with Web reports are indicated by the word **Web** in that list of tabs. After going to that tab and selecting the desired Web report, pressing the Generate button does all the rest.



## Chapter 7. Managing the Database

This chapter describes miscellaneous things you might want to do with the data in your database.

### 7.1. Creating Backups

As with any program that uses a database, it is good practice to make backup copies of your data to protect you against power failures or system and program crashes as well as human errors. JRaceman makes it easy to do frequent backups with the push of a button. The Backup tab has the fields and buttons to allow you to do this. Once you have set up a backup directory using that tab, you can use the Backup button in the menu bar to quickly make a backup of the database.

### 7.2. Checking Your Data

JRaceman can check the data in the database for correctness and consistency. The commands in the Checks tab perform these checks. You should run these checks occasionally to make sure there are not errors in your data which could lead to problems or confusion later.

Each Check command displays the errors it finds in the Results tab. If you find an error in your data using a Check command, it is usually possible to double-click on a row in the Results tab and be taken directly to the editing form for that data. In some cases you may be able to look at the data to understand the cause of the error being reported, then use the tab of your choice to correct the error. If the data can not be corrected using one of the standard tabs, you can always edit the data using debugging mode.

### 7.3. Sharing Data by Import and Export

JRaceman is designed to allow you to share your data with other people. You can export your data to text files, and you can import data from text files into your database. Here are some of the things you can do with this capability:

- Allow each club to enter their own data and submit it to you via a text file.
- Share your results of your meet with participating clubs.
- Make text backup files of your database.



## Chapter 8. Web Registration

This chapter describes how to set up and use Web Registration for your database.

### 8.1. Set Up

TODO



## Part III. User Interface

This Part is intended to help you learn more of the details about specific parts of JRaceman. It is organized based on the layout of the user interface to allow you to find out how to do more with each part of the interface. You should look in this Part when you want to manipulate a specific part of the user interface, but you don't remember exactly how to do it.



## Chapter 9. Main Window

The main JRaceman window consists of the following areas:

- A menu bar at the very top of the window.
- Input Tabs in the top half of the window just under the menu bar.
- A separator bar between the Input Tabs and the Output Tabs.
- Output Tabs in the bottom half of the window.
- A status line at the very bottom of the window.

### 9.1. Input Tabs

The top half of the JRaceman window consists of a set of tabs to access different categories of data. Within each of the top-level tabs there are subtabs for more specific access to one kind of data.

### 9.2. Separator Bar

The Separator Bar between the Editing Tabs and the Output Tabs can be used to control the amount of space allocated to each set of tabs.

- To change the portion of the window allocated to the Editing Tabs versus the Output Tabs, drag the separator bar up or down to the desired location.
- To allocate all of the space to one set of tabs, click on one of the arrows in the separator bar to move the bar in the direction of the arrow. To restore the display of both windows, click the other arrow in the separator bar.
- To exchange the positions of the two sets of tabs, double-click on the separator bar.

### 9.3. Output Tabs

The bottom half of the JRaceman window consists of a set of tabs for displaying various kinds of output.

- The Messages tab displays status and information messages.
- The Query Results tab displays the results of searches from any of the Input tabs.
- The Report tab displays reports generated by the program.
- The Help tab displays help information.

### 9.4. Status Line

At the very bottom of the window is the status line. This line displays messages from the program, such as noting that a record was saved. Messages displayed in the Status Line are also recorded in the Messages tab, so you can see old status messages by selecting the Messages tab.



## Chapter 10. User Interface Elements

This chapter describes in general terms how to use the different elements that are used in the graphical user interface of JRaceman.

### 10.1. Terms

The following terms are used in this chapter:

click	Pressing and quickly releasing one of the buttons on the mouse.
drag	Moving the pointer while a button is pressed.
mouse	The physical device which you move in order to make the pointer move on the screen.
panel	A collection of components on the screen which are visible at one time. JRaceman includes many panels, only two (one input and one output) of which are visible at one time. You control which panels are visible by selecting the appropriate tabs. Each panel is controlled by one tab (which may be contained within another panel).
pointer	The icon on the screen which moves according to how you move the mouse. Also known as the cursor, it is usually a small arrow pointing up and slightly to the left, but can change depending on what the program is doing.
press	Pressing and holding down one of the buttons on the mouse.
tab	The small labeled area which, when clicked, makes visible the associated panel.

### 10.2. Tool Tips and Tool Prompts

A Tool Tip is a small box with text that may appear when you move the pointer over some component in the window and let it rest there for a moment. The Tool Tip text gives information about that component to help you know what it does. When you move the pointer, the Tool Tip box will disappear.

A Tool Prompt is text which appears in the Status Line at the bottom of the window when you move the pointer over some component in the window. As with a Tool Tip, the Tool Prompt text gives information about that component to help you know what it does. When you move the pointer, the Tool Prompt text will disappear and the previous contents of the Status Line will be restored.

Tool Tips are most commonly used in JRaceman on buttons and tabs. Tool Prompts are most commonly used in JRaceman on menus.

### 10.3. Using Menus

The main window includes a menu bar at top of the window. To see the contents of a menu, move the pointer over the menu keyword in the menu bar and press or click the left mouse button. The menu will open up below the pointer, after which you can move the pointer to the desired menu item. If you pressed the mouse button, select the desired menu item by releasing the mouse button when the pointer is over that item. If you clicked the mouse button, select the desired menu by clicking the mouse button again when the pointer is over that item.

You can see all of the menus with a single mouse press or click; you don't need to press or click the button again when moving to a different menu, just move the pointer to that menu and it will open in place of the previous menu.



Some menus contain additional menus. These additional menus are indicated by a small right-pointing arrowhead on the right side of the menu item. When you move the pointer to one of these items, another menu will open up to the right of the already-open menu. You can then move the pointer into this new menu in order to select an item from it.

Some menu items have a label which ends with an ellipsis, like this: ... This indicates that selecting that menu item will open a dialog box with additional questions. No other action will be taken for these commands until specified by accepting the dialog. Menu items which do not end with an ellipsis may take immediate action when selected.

## 10.4. Using Tabs

Tabs are a mechanism for allowing multiple panels of components to share the same space in a window. You control which of those panels is visible by selecting the tab for that panel. You can think of the panels as a stack of papers, one of which is on top of the stack. You can see the panel on top of the stack, but you can't see the panels which are underneath; to see one of the panels underneath, you need to raise it to the top of the stack, at which point the previous top panel will no longer be visible.

In JRaceman, the tabs are just below the menu bar in the main window. Select a tab by moving the pointer to that tab and clicking the left mouse button. The panel for that tab will appear.

JRaceman uses hierarchical tabs; in other words, many of the panels themselves contain a set of panels controlled by tabs. When this kind of panel is made visible by clicking on its tab, another set of tabs appears just below the previously selected tab. You can then select one of those tabs to see that sub-panel. JRaceman uses three levels of tabs, so you may have to click three times (once at each level of tabs) to get to the desired panel.

To simplify finding the right panel, there is a Tabs menu which display the tabs in "upside-down" order from how they appear in the main window. In the main window, the third row of tabs ususally includes entries such as Edit, Query, Sheet, and Export. The Tabs menu includes these items as second-level menus within the Tabs menu, allowing you to navaigate directly to one of these panels without having to remember the first and second-level tabs required to see the desired tab for your panel.

## 10.5. Choice Lists

A Choice List is a component that allows you to select from a fixed number of choices. Choice Lists are used extensively in JRaceman for data fields that refer to other tables. Examples of where Choice Lists are used in JRaceman:

- The Team field in the Query and Edit tabs in the People tab.
- The Event field in the By Event tab.

When you are entering data in a tab and a Choice List in that tab does not contain the data you want, you can switch to the tab where that type of data is defined, add a new record with the desired data, then switch back to the original tab. When you return to the original tab, the Choice List will be updated to include the newly added data.

## 10.6. Scroll Bars

When the data being displayed in a panel is too large to be visible all at once, scroll bars are added to allow the contents of the panel to be moved such that any part of it can be made visible. If the panel is too wide to fit, a horizontal scroll bar appears on the bottom of the panel; if too tall, a vertical scroll bar appears on the right of the panel. You control what part of the data is visible by manipulating the scroll bar.

The vertical scroll bar to the right is used to move the data vertically; the horizontal scroll bar below is used to move the data horizontally. The discussion below about how to control the scroll bar assumes a vertical scroll bar that



moves the data up and down. The horizontal scroll bar works in exactly the same manner, but with "up" and "down" (or "above" and "below" ) replaced by "left" and "right", respectively.

A Scroll Bar contains two arrows and a scrolling region. The scrolling region has an active area called the thumb. The size of the thumb as a fraction of the size of the scrolling region indicates the amount of data visible at once as a fraction of the total amount of data. Thus, if the thumb takes up half of the scrolling region, then you can see half of the data at once; if the thumb takes up a tenth of the scrolling region, you can see a tenth of the data at once. The smaller the thumb, the more data there is.

To move the data by a small amount, click on the up or down arrow in the scroll bar. This will move the thumb by a small amount up or down, which will move the visible data in the other direction by a small amount (the viewing region moves through the data in the same direction as the thumb, so the visible data on the screen moves in the other direction).

To move the data by one page (i.e. by an amount equal to the height of the displayed data), move the pointer to the scrolling region above or below the thumb and click the left mouse button. This will move the thumb up or down by one thumb-length, which will move the visible data in the other direction by one page.

To move the data by an arbitrary amount, move the pointer on to the thumb, then press the left mouse button and drag the pointer up or down to move the thumb up or down. The data will move as you move the pointer. Release the button when you reach the desired point in the data.

## 10.7. Split Panes and Separator Bars

A Split Pane allows two smaller areas to be viewed at the same time side by side or above and below. Between the two smaller areas is a separator bar. The separator bar is used to control how much of the viewing area is used for each of the two smaller areas. You can allocate all of the view to one, all to the other, or anywhere in between.

A pane which is split into top and bottom parts has a horizontal separator bar between the two parts; a pane which is split into side by side parts has a vertical separator bar between the two parts. The discussion below about how to control the separator bar assumes a horizontal separator bar which controls the size of the areas above and below. The vertical separator bar works in exactly the same manner, but with "up" and "down" (or "above" and "below" ) replaced by "left" and "right", respectively.

To move the separator bar, move the pointer to the stippled part of the bar, press on the left mouse button, and drag the mouse up or down.

To hide the upper area and show the maximum amount of the lower area, click on the little up-arrow at the left end of the separator bar. This will move the separator bar all the way up. To return to the split configuration showing both areas, click on the little down-arrow at the left end of the separator bar. Similarly, when the display shows both areas, click on the down-arrow to move the separator bar all the way down, hiding the lower area and showing the maximum amount of the upper area. Click on the up-arrow to return to the split configuration.

To swap the positions of the upper and lower areas, double-click anywhere on the stippled part of the separator bar.

## 10.8. Tables

A table is a rectangular area with rows and columns, used to display data. The intersection of a row and a column is called a cell. Each column is labeled with a column name.

If there are too many rows of data to fit in the viewing area, a vertical scroll bar appears on the right side of the table. You can scroll through the data rows using this scroll bar. The column labels do not scroll, so are always visible at the top of the table.

To sort the data on a column, move the pointer into the label for that column and click the left mouse button. A small arrow will appear in that column indicating that the data is sorted on that column. To change the direction of



the sort in the column, click again. Each time you click, the direction of the sort reverses. The direction of the arrow in the column label indicates the direction of the sort.

The sorting algorithm is a stable sort, which allows you to sort on a secondary column. To sort on a secondary column, click on that column first to sort it, then click on the primary sort column. The sorting indicator will only appear in the primary sort column, but rows with the same primary column value will remain sorted according to the secondary column.

To rearrange the columns in a table, move the pointer into the label for the column to be moved, then press the left mouse button and drag the column to its new location. As you move the column to the left or right, the other columns will move aside to make room for it. When the column is in the desired location, release the mouse button.

## 10.9. Sheets

A sheet is a table in which some or all of the data can be edited, similar to a spreadsheet. A panel with a sheet in it allows you to edit the information about multiple records on one page, to simplify entering or editing a group of records.

To edit a set of records, press the "Search" button. A list of all matching records will appear just below the button.

As with any table, you can sort the sheet by any column by clicking on the header label for that column. See the section on tables above for more information on sorting and rearranging columns.

You can directly edit the value of the fields in any column with a header label in a **bold font**. To edit a value, click on the field, type in the new value (or select it from the drop-down list of choices), and press Enter or Tab. The new value is stored in the database when you press Enter. You can re-sort the table after making changes by clicking on the column header again.

To abort editing a field, press Escape.

To add a new record, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. If you have sorted on a column, the new row may be added at the beginning of the sheet rather than the end. In most cases, all fields in the new row will be blank. If there are Search fields in the tab above the Sheet, the selected values in the Search fields will often be set into the newly added row. After you have filled in values for all required fields an ID will be generated and the values in that row will be stored in the database.

Until you fill in all required fields in a new row, the record is not permanent; if you push the Search button again, the unstored row will be lost. Clicking on any column header will sort all rows, including the new rows, even when they are not yet stored. You can easily find new rows by looking for rows with no ID value.

Not all data for a record is necessarily displayed in the Sheet. To edit the remaining fields in the record, you must use the Edit tab for that record. To edit a record, click on that row in the Sheet, then push the Edit button. This will switch to the Edit tab for the selected row. In that tab, you can edit any field for that record, or delete the record. To get back to the editing Sheet, select the Sheet tab again.

### 10.9.1. Function Keys

You can use some of the function keys on your keyboard to help fill in data values in the sheet more quickly. To use one of these function keys, select the cell to be filled, but do not edit it. If clicking on the cell puts it into edit mode, press Escape to exit edit mode.

- F3 - copy the data from the cell above into this cell and move to the next cell to the right. By pressing F3 repeatedly, you can easily copy all fields of the row above into the current row.
- F4 - copy the data from the cell above into this cell and move to the next cell below. By pressing F4 repeatedly,



you can easily fill a column with one value.

- F6 - take the data from the cell above into this cell, increment it, enter the incremented value into this cell, and move to the next cell below. The increment value is normally 1. An editing sheet tab may include a box allowing you to specify the increment amount, such as the Schedule tab. By pressing F6 repeatedly, you can easily fill a column with a set of unique numbers. If the cell above contains a string value, this function key behaves identically to F4.



## Chapter 11. Menus

This chapter describes the JRaceman menus.

### 11.1. The File Menu

The File menu provides program-wide commands.

- Exit - After getting confirmation, exits from JRaceman.

### 11.2. The Database Menu

The Database menu provides commands to manipulate the database.

- Export Standard Data to JRaceman Data File - prompts for an output file and writes all of the data JRaceman knows about to that output file in a format suitable for reading in later or into another database using the Import command.
- Import JRaceman Data File - prompts for an input file such as is produced by an export command and reads the contents of that file into the database.
- Import SQL File - prompts for an import file containing SQL statements and executes those statements.
- Check Upgrade - checks to see what changes need to be made to the database to be compatible with this version of JRaceman.
- Upgrade - makes the changes to the database to be compatible with this version of JRaceman.

### 11.3. The Tabs Menu

The Tabs menu provides an alternate navigation route to the tabs.

- Query - lists all of the bottom-level tabs which are Query tabs.
- Edit - lists all of the bottom-level tabs which are Edit tabs.
- Sheet - lists all of the bottom-level tabs which are Sheet tabs.
- Export - lists all of the bottom-level tabs which are Export tabs.
- Report - lists all of the bottom-level tabs which contain Report buttons.

### 11.4. The Query Results Menu

The Query Results menu provides commands to manipulate the contents of the Query Results tab.

- Export As - exports the contents of the Query Results tab as a JRaceman export file to the specified file.



- Save SQL As - writes out the contents of the Query Results tab as a collection of SQL statements.
- Save HTML As - writes out the contents of the Query Results tab as an HTML table suitable for viewing in a browser.
- Save Tab Delimited As - writes out the contents of the Query Results tab as a tab-delimited file suitable for loading into a spreadsheet program.
- Save CSV As - writes out the contents of the Query Results tab as a comma-separated-values file suitable for loading into a spreadsheet program.
- Display HTML in Report Tab - converts the data displayed in the Query Results tab into HTML and displays it in the Report tab.
- Print HTML - converts the data displayed in the Query Results tab into HTML in the Report tab and opens the Print dialog to print that HTML report.

## 11.5. The Report Menu

The Report menu provides commands to manipulate the contents of the Report tab.

- Save HTML As - writes out the contents of the Report tab as an HTML file.
- Print - opens the Print dialog to allow printing the contents of the Report tab.

## 11.6. The Help Menu

The Help menu provides information to help you understand how to use JRaceman, displaying it in the Help tab.

- Tutorial Wizard - displays the Tutorial Wizard dialog.
- Welcome - displays the preface to the JRaceman Users Guide.
- User Guide - displays the start of the Users Guide.
- Help On This Tab - displays help about the currently displayed tab in the top half of the JRaceman window.
- Help On Help - displays help on how to use the Help menu.
- Open Page - allows specifying a URL to be opened in the Help tab.
- About - a submenu containing commands to display documentation about JRaceman.
- Back - go back to the previously displayed page in the Help tab.
- Forward - after going back, go forward again to the next displayed page in the Help tab.
- Show Location - print the URL of the current page in the status line.
- Print - print the page currently being displayed in the Help tab.



## Chapter 12. Batch Mode

This chapter describes command line options available for running JRaceman in batch mode.

Command line options are case-insensitive; the command **-initDB** can be entered as **-initdb**, **-INTDB**, or **-InitDb** with equal effect.

Some commands are action commands, which cause JRaceman to take the specified action. If no action command is specified, JRaceman will take its default action, which is to open the main JRaceman editing window. The following commands are action commands for which JRaceman will not open its main window: **-checkUpgrade**, **-createDB**, **-exportDB**, **-initDB**, **-repair**, **-runserver**, **-sql**, **-sqlFile**, **-sqli**, **-upgrade**.

The following command line options can be specified:

### Command Line Options

<b>-checkUpgrade</b>	Check to see if the database needs to be upgraded in order to use this version of JRaceman.
<b>-createDB</b>	Create an empty database at the specified location.
<b>-Dname=value</b>	Set the specified Java system property to the specified value.
<b>-database databasepath</b>	Specify the location of the database to use. The shorter command <b>-db</b> is equivalent to <b>-database</b> .
<b>-dbhost hostname</b>	Specify the hostname of the database server to use. Use this command line option instead of the <b>-database</b> option when you want to run in client/server mode rather than in standalone mode. See the <b>-runserver</b> option.
<b>-debug</b>	Turn on the debug flag.
<b>-exportDB exportpath</b>	Export the contents of the standard tables and columns to a file at the specified location.
<b>-help</b>	List these command line options.
<b>-initDB</b>	Create a database at the specified location and upgrade it to be compatible with this version of JRaceman. This command is equivalent to <b>-createDB -upgrade</b> .
<b>-password password</b>	Specify a database password.
<b>-repair</b>	Attempt to repair a corrupted database. Make sure no other JRaceman program is accessing the database when this option is used. This runs the Mckoi database repair tool. Depending on the amount of corruption, this may or may not be able to fix the problem. If the corruption can not be repaired, you will have to go back to a recent backup.
<b>-runserver</b>	Run a database server. The server listens on port 9157. After the server has started, run clients by using the <b>-dbhost</b> option. Shut down the server using the <b>-sql</b> option with the <b>SHUTDOWN</b> command.
<b>-setDefaultDB databasepath</b>	Set the default database path to use when JRaceman is run without the <b>-db</b> command line option. This option is only available on Windows.
<b>-sql sqlcode</b>	Execute the specified SQL command.
<b>-sqlFile filename</b>	Read and execute SQL commands from the specified file.



<code>-sqli</code>	Enter SQL-interactive line mode. In this mode, you can type in arbitrary SQL commands.
<code>-testDbRootDialog <i>path</i></code>	A debugging option to test the DB Root dialog.
<code>-textSnapShot <i>outdir</i></code>	Export a text snapshot of the database to a dated file in the specified directory.
<code>-trace</code>	Turn on trace mode for debugging.
<code>-tuneProgress</code>	A debugging option to tune the progress bar in the startup dialog.
<code>-upgrade</code>	Upgrade the database to be compatible with this version of JRaceman.
<code>-user <i>username</i></code>	Specify a database username.



## Chapter 13. Top Level Input Tabs

These are the top-level input tabs in the top half of the main window. They are listed here in alphabetical order.

### 13.1. By Event

The By Event tab allows you to select an Event on which to operate, and then to choose the task to perform on that Event. On race days you will probably be using this tab more than any other.

The By Event tab has no subtabs. Instead, it has two choice lists. On the left, you select the Event or Race on which to operate. On the right, you select the task to perform on that Event or Race. Selecting a task will change the contents of the By Event tab to allow you to perform that task.

#### 13.1.1. Selecting an Event

You can select an Event by Event number, Race number or Event Name.

To select an Event by Event number, choose the "By Event#" choice in the drop-down list at the top left of the tab, then choose the desired Event from the drop-down list just to its right.

To select an Event by Race number, choose the "By Race#" choice in the drop-down list at the top left of the tab, then choose the desired Race from the drop-down list just to its right.

To select an Event by Event name, choose the "By Event Name" choice in the drop-down list at the top left of the tab, then choose the desired Event from the drop-down list just to its right.

#### 13.1.2. Tasks

The ByEvent tab includes the following tasks:

- The Create Races task allows you to create the Races for the selected Event. This is similar to the Create Races box in the Schedule tab.
- The Entries/Progress task allows you to do manual and automatic progression, and to add, delete, and edit the Entries for the selected Event. The Add Entry and Delete Entry buttons are similar to the sheet in the Entries tab.
- The Results task allows you to enter results for the selected Event.
- The Reports task allows you to generate Reports for the selected Event. These reports are the same as the reports which are available in the Finish tab, plus the Progress report which is available in the Progress Auto tab.

When using the Entries task, the Group Selected Entries menu command in the By Event is enabled. You can select multiple Entries from the sheet (using shift-click or control-click to add to the existing selection), then select the Group Selected Entries command to group those Entries into a group by themselves.

When using the Results task, you can use the Import Race Results menu command in the By Event menu to import results data from a spreadsheet in the format used by the Omega finish-line system. To configure this, you must use the Context Options tab to define an option named "ImportRaceResults.filename" that points to the Excel spreadsheet file containing the results data. When you select the Import Race Results menu command the results for the selected Race are retrieved from the row in the spreadsheet matching the selected Race number. Race 1 comes from row 2, Race 2 from row 3, etc. If the selected Race number is not an integer, the Race number is truncated to an integer and the results for that Race number are retrieved from the spreadsheet. A dialog is then displayed, with the results and lanes sorted by time. Press the OK button to import the data and update the place and score data.



## 13.2. Database

The Database tab is used to do general tasks on the database.

The Database tab includes the following subtabs:

- The Checks tab allows you to check the consistency of the data in the database, so that you can find data entry problems and constraint violations.
- The Backup tab allows you to set up a directory into which backup snapshots can easily be generated with the push of a button.
- The Options tab allows you to edit the values of options that control the behavior of JRaceman.
- The Context Options tab allows you to edit the values of context-specific options that control the behavior of JRaceman.
- The Transfer tab allows you to transfer data to and from an automatic finish-line timing system.

## 13.3. Debug

The Debug tab allows you into the innards of the program. Most people probably won't have much use this tab.

The Debug tab includes the following subtabs:

- The SQL tab allows you to enter an arbitrary SQL command to the database. Pushing the Execute button will execute the SQL command and display the results in the Query Results output tab. A discussion of SQL is beyond the scope of this document.
- The Change ID tab allows you to change the ID of a row, along with all records that reference that row. You can specify the table, the old ID and the new ID.
- The Edit Field tab allows you to examine or modify the value of one field in the database. You can specify the table, the column and the WHERE clause to select the desired row. This is useful for making corrections to text fields containing a large amount of data.
- The Foreign Keys tab helps in examining and updated references to other records via foreign keys. You can specify a table and two key (ID) values, then display the list of foreign key columns for the table, display the two rows, find references to them, and update either set of references to point to the other row.
- The Properties tab allows you to examine or modify the values of Java System properties. Pushing the View Properties button displays the names and values of all system properties in the Help output tab. See the JDK documentation for information on Java System properties.
- The Generate tab allows you to generate large numbers of Team and People records for the purpose of testing and debugging.
- The Progress tab allows you to set up Races for an Event and progress those Events through their Stages, moving winning racers on to the next Stage and doing automatic lane assignments. This tab is obsolete, and is here only in case of bugs in the new code.
- The Finish tab allows you to easily enter the finishing times or marks for a Race. This tab is obsolete, and is here only in case of bugs in the new code.



- The Races tab allows you to examine the race records, making manual changes if necessary.
- The Lanes tab allows you to examine the individual entries to each race, making manual changes if necessary.

## 13.4. Meet Setup

The Meet Setup tab allows you to set up a meet and the events at that meet. This needs to be done once before each meet. You decide what events will be held at the meet and define them in these subtabs. Handling of registration of competitors is done in the Entries subtab.

The Meet Setup tab includes the following subtabs:

- The Meets tab allows you to define a new meet.
- The Registration Fees tab allows you to define how much to charge for each person entered into a Meet based on the number of entries for that person.
- The Registrations tab allows you to enter registration information for a meet.
- The Events tab allows you to define the events to run at a meet.
- The Entries tab allows you to enter people into different events. Once all entrants have been signed up for an Event, you tell JRaceman to generate the Races for you, which it does according to the Progression selected for that Event.
- The Schedule tab allows you to easily set up the start times and scheduled durations for all of the events in one area.

## 13.5. Plan Setup

The Plan Setup tab is used to define progression plans (Simplans and Complans).

The Plan Setup tab includes the following subtabs:

- The Simplans tab allows you to define the systems and plans used for simple progressions.
- The Simplan Stages tab allows you to define the number of sections of each Stage used in the simplans.
- The Simplan Rules tab allows you to define the progression rules for the simplans.
- The Complans tab allows you to define the systems and plans used for complex progressions.
- The Complan Stages tab allows you to define the number of sections of each Stage used in the complans.
- The Complan Rules tab allows you to define the progression rules for the complans.

## 13.6. Reports

The Reports tab allows you to generate reports.

The Reports tab includes the following subtabs:



- The General tab allows you to generate most reports. You can select the report format, which records you want to report on, and some formatting controls.
- The Score tab allows you to calculate team scores.
- The Award Labels tab allows you to print labels for awards.

## 13.7. Site Setup

The Site Setup tab allows you to define the locations where a meet is held. This information only needs to be entered the first time a meet is held at a site.

The Site Setup tab includes the following subtabs:

- The Sites tab allows you to define the Sites at which a meet can be held.
- The Areas tab allows you to define the areas within a site.
- The Lane Order tab allows you to define the order in which lanes are assigned for an Area during progression.

## 13.8. Sport Setup

The Sport Setup tab is used to set up information which is typically the same for all users of the program within the same sport. You should be able to go through all of the subtabs of the Sports Setup tab once when you first set up your database, after which you should almost never have to make any changes.

The Sport Setup tab includes the following subtabs:

- The Competitions tab allows you to set up the basic sports events for your sport. For example, Track and Field users might set up competitions such as *100 Meter Run* and *Shot Put*.
- The Levels tab allows you to define the different levels of competition in your sport. Usually this means different age groups.
- The Genders tab allows you to define the Gender terms used in your sport.
- The Progressions tab allows you to define the different kinds of Progressions used in your sport. A Progression defines how competitors advance from one round to the next, such as from Heat to Semifinal to Final.
- The Scoring tab allows you to define the scoring systems to use for your meets.
- The Stages tab allows you to specify the names of the different stages in a progression, such as *Heat* and *Final*.
- The Exceptions tab allows you to specify the different kinds of exceptions in your sport, which are the reasons why a competitor might be disqualified for an event.

## 13.9. Team Setup

The Team Setup tab allows you to set up information about the competitors. If you have multiple meets with some of the same competitors (such as the people on your own team), you only need to enter the information about those people once. Thus, after the initial work of entering information for the first meet, there will be less information to be entered for subsequent meets, since many of the teams and people will be the same.



The Team Setup tab includes the following subtabs:

- The Challenges tab allows you to define the Challenge names.
- The Teams tab allows you to define the Teams.
- The People tab allows you to define the People.
- The Seeding Plans tab allows you to define Seeding plans.
- The Seeding Lists tab allows you to define Seeding lists.



## Chapter 14. Input Tabs

These are the second-level input tabs which are contained in the top-level input tabs. They are listed here in alphabetical order.

### 14.1. Areas

This tab is in the Site Setup tab.

The Areas tab is used to define the areas within a site at which events can take place. One site can contain multiple areas at which different events can occur simultaneously.

Events for an Area are scheduled using the Schedule tab.

#### 14.1.1. Query and Edit Tabs

- The ID field is the unique ID for this Area. The default ID prefix is the site ID followed by a dot and the letter A.
- The Name field is the name of the area.
- The Site field indicates the Site containing this area.
- The Lanes field is the number of lanes or starting positions for this area. This is used by the Progression module to do lane assignments.
- The Extra Lanes field is the number of overflow lanes or starting position for this area, which are available for exception conditions. This is used by the Progression module to do lane assignments.

#### 14.1.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of the area within the site
siteId	String	The site at which this area is located
lanes	int	The number of lanes at this site
extraLanes	int	The number of overflow lanes at this site

### 14.2. Awards

This tab is in the Reports tab.

The Awards tab allows you to print labels for awards.

You can choose from the following label formats:

- The "1in by 2 5/8in (Avery 5160)" labels are formatted for 8-1/2 x 11 inch paper divided into 3 labels horizontally by 10 labels vertically, with half inch margins on top and bottom, and 3/8 inch margins on the sides, for a



total of 30 labels each 1 inch tall and 2-5/8 inches wide. Avery produces a number of different labels (white, clear, laser, ink-jet) in this format with numbers similar to 5160, all of which have the same layout as 5160.

- The "1in by 2.83in" labels are formatted for 8-1/2 x 11 inch paper divided into 3 labels horizontally by 11 labels vertically, for a total of 33 labels each 1 inch tall and about 2-3/4 inches wide. This is for an older style of labels which is no longer generally available.
- The "2.2in by 4.25in" labels are formatted for 8-1/2 x 11 inch paper divided into 2 labels horizontally by 5 labels vertically, for a total of 10 labels each about 2 inches tall and about 4 inches wide.
- The "1-2/3in round (Avery 5293)" labels are formatted for 8-1/2 x 11 in paper divided into 4 labels horizontally by 6 labels vertically, for a total of 24 labels each just under 2 inches in diameter. These are suitable for using on the back of round medals.

To print labels, select a value for the desired Meet, and for the desired Event number or numbers. To print labels for all events in a Meet, leave the Event number blank. On the right hand side of the form, select the desired label format from the list of available formats. The number of labels per page for that format is displayed immediately below the format name. Press the "Print Labels" button to generate the label pages and open the Print dialog. From the Print dialog you can select a printer, or print to a file.

To print labels for one event, enter the number of that event into the Event number field. To print labels for multiple events, enter the numbers of all desired events into the number field, separated by commas or spaces, and select the IN operator from the operator choices (between the "Event#" label and the Event number text entry field) for the Event number.

The "Labels For" choice allows you to choose to print labels only for Scoring finishers, only for Non-Scoring finishers, or for both. If both are printed, the Scoring finishers are printed first, matching the order of the Awards report in the Finish tab. The place printed for Scoring finishers is the scoring place; the place printed for Non-Scoring finishers is the overall place, followed by "(Non-Scoring)".

If you have printed a partial page of labels, you can later print on the rest of the labels by using the Blank Label Count field. Enter a number from in this field to skip that many labels on the printed page. (If you choose a number larger than the number of labels per page, the number will be reduced modulo the number of labels per page; it will not print out any blank pages.) For example, say you selected event number 11 (a K4 event) and printed out 8 labels (4 1st place and 4 2nd place) on a fresh page. The remaining 25 labels on the page are blank. Now you want to print labels for event 12 onto those unused labels. Insert that page into the printer, select event number 12 in the Awards tab, set the Blank Label Count to 8, and press the Print Labels button. The first 8 (already used) labels will be skipped, and the labels will start printing at label number 9. (Note, however, that the directions on some labels advise against running a sheet of labels through a printer more than once.)

For all labels other than the round labels, you can specify images to be used on the left and right ends of the labels. To specify an image to use on the left end of the label, use the Context Options tab to set the value of the option "Meets.Awards.labelImageLeft" to the URL of the image. If no matching Context Option is found, you can set that URL as the value of the Label Image Left field in the Meets tab. The right image can be specified similarly using the Context Option "Meets.Awards.labelImageRight" or the Label Image Right field of the Meets tab.

For round labels, the left image, when set as described above, is used as the background image for the label.

## 14.3. Backup

This tab is in the Database tab.

### 14.3.1. Creating Backups

The Backup tab allows you to set up one or more backup directory locations, then to easily create a backup of the



database to one of those locations.

To set up the first backup directory, enter the desired directory location into the "Snapshot Directory 0" field in the Backup tab. You may use the Browse button to help you find an appropriate directory.

Press the "Text Snapshot" button to produce a JRaceman export file in the backup directory. This creates a file with the same contents as would be created if you used the "Export Standard Data to JRaceman Data File" command in the Database menu, but it automatically generates a file name based on the date and time of the backup. Thus you can come back and press the button again, and it will create another backup file with a new name. The old backup files remain in the backup directory until you delete them (using your regular file system tools). You can have as many backup files as you have space on your disk.

The "Mckoi Snapshot" button works similarly to the "Text Snapshot" button, except that it creates a complete database directory, rather than a text file, in the backup directory. As with "Text Snapshot", the directory name is based on the date and time of the backup, so you can create multiple backups without overwriting the old backups. A Mckoi backup directory takes quite a bit more space than a text backup file, so you may want to be more conservative in how often you create them. The old Mckoi backup directories remain in the backup directory until you delete them (using your regular file system tools).

Note that the Mckoi backup is done by the database server, not by the client, so you will not be able to use the Mckoi Snapshot button from a client machine when running in client/server mode.

You can mix Text backups and Mckoi backups in the same backup directory. Text backups use the extension ".txt"; Mckoi backups use the extension ".db".

The second backup directory and set of buttons work just like the first. They are there to allow you to have two different backup directories. For example, you might put those backup directories onto two different machines.

### 14.3.2. The Quick Backup Button

The Backup button in the menu bar provides an easy and quick way to backup your database. Backup files are small, so you can (and should) backup often.

Set the directory into which quick backups are generated by using the Context Options tab to set the value of the option "Backup.directory" to the direction in which you want to save your backups. As a Context Option, you can set this on a per-host basis so that every client can back up to a local directory. If no Backup.directory option is found, the quick backup button uses Snapshot Directory 0.

### 14.3.3. Recovering A Backup

In the event that you have a crash that corrupts your database, and the repair procedure is unable to recover the data to your satisfaction, or if you have accidentally deleted a large amount of data, you will have to use a backup file.

In the case of a corrupted database, before using a backup file you can try running the repair tool by using the -repair command line options, as in this command:

```
java -jar jraceman.jar -repair
```

This runs the Mckoi repair tool, which attempts to repair corruption in the database. Make sure that you are not running a database server or any other program that accesses the database when you run the repair tool.

To recover using a Text backup file, rename the old database folder (the default name is jracedb, and the path is normally displayed in the splash screen as JRaceman starts) to another name, then rerun JRaceman to create a new database. Do not import a sport definition file; when it asks you about that, cancel. From the Database menu, select the "Import JRaceman Data File" command and specify the backup file from which to read. The backup filenames



encode the date and time they were made, with the year first and the seconds last. This means the oldest file will normally sort to the beginning, and the newest will normally sort to the end. You will probably want to use the newest file. When the import finishes, you are ready to go.

To recover using a Mckoi backup, rename the old database folder to another name, and copy the selected Mckoi backup folder (one with the name that starts with the year) to the database folder location that you just vacated. Restart JRaceman - it should read the new database and be ready to go. As with the Text backup files, the Mckoi backup directory names encode the date and time of the backup, and will normally sort with the oldest first and the newest last.

If you want to look at the data in a Mckoi backup, you can directly run JRaceman on that backup database using the `-db` command line option, such as in this command:

```
java -jar jraceman.jar -db/usr/jraceman-backups/2003-06-26-09-48-42.db
```

Any changes you make to this backup database will be stored in the backup database, but will not affect the main database or the other backup databases.

## 14.4. Challenges

This tab is in the Team Setup tab.

The Challenges tab is used to define the names of the Challenges which are available for use in the Teams tab.

### 14.4.1. Query and Edit Tabs

- The ID field is the unique ID for this Challenge. The default ID prefix is **CHL**
- The Name field is the name of the Challenge category.

### 14.4.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this Challenge category

## 14.5. Checks

This tab is in the Database tab.

The Checks tab allows you to check the database for data entry errors and constraint violations.

To check the database, select a value for the desired Meet and, where applicable, for the desired Event number:

- To check events in all meets in the database, leave the Meet blank. To check events in a specific Meet, select the Meet.
- To check all events in a Meet, leave the Event number blank.



- To check one event, enter the number of that event into the Event number field.
- To check multiple events, enter the numbers of all desired events into the number field, separated by commas or spaces, and select the IN operator from the operator choices (between the "Event#" label and the Event number text entry field) for the Event number.

On the right hand side of the form, push the button to run the desired check. Problems are displayed in the Query Results output tab. If no problems are found, only the header row will be displayed, but there will be no data rows displayed.

You can run the following checks:

- Registrations - checks to make sure that people are registered, paid up, have signed a waiver, and have an unexpired membership. Does not use the Events selection.
- Names - checks to make sure that names have no leading or trailing spaces and are not in all lower case; checks for duplicate names both within one team and across teams. This check can only be used when no Meet has been selected.
- Genders - checks to make sure that people are not entered into events for the opposite gender.
- Levels - checks to make sure that people are not entered into events for which they are too young or too old.
- Groups - checks for errors in entries to group events: too many or too few people in a group, too many alternates, group or alternate specified for a non-group event, no group specified for a group event, group name with leading or trailing spaces.
- Entries vs. Lanes - checks for consistency between Entries and Lanes, such as Lane assignments for a group which are not all in the same Lane.
- Concurrent Race Lanes - checks to make sure that the same Lane is not assigned in two Races which are in the same Area at the same Schedule Start Time.
- Max Entries Per Person - checks to make sure that each person is not entered in more than the specified number of events in one meet, not including alternate entries. Enter the maximum allowed number in the text field, then press the button.
- Max Alternates Per Person - checks to make sure that each person is not entered as an alternate in more than the specified number of events in one meet. Enter the maximum allowed number in the text field, then press the button.

This is a different check from the maximum number of alternates per event, which is defined in the Competitions tab and checked by the Groups check.

## 14.6. Competitions

This tab is in the Sport Setup tab.

The Competitions tab is used to define the basic types of events used in this sport. The name field indicates the type of event, but not including the level or gender, which are specified as part of Event Setup.

### 14.6.1. Query and Edit Tabs

- The ID field is the unique ID for this Competition. The default ID prefix is C.



- The Name field is the name of the competition. For example, "100 meter dash" or "K-1 500m".
- The Group Size field is used for events in which a group of people compete together against other groups. For example, in a 4x400m relay, the Group Size would be set to 4. For one-person events, this field should be left blank.
- The Max Alternates field specifies how many alternate Entries are allowed for a group entered into this Event. This is usually correlated with the Group Size.
- The Duration field should be set to the amount of time to be allocated to this event for scheduling purposes. For example, if you want events to start at 10 minute intervals, you would enter "10:00" into this field.

## 14.6.2. Export

The Export tab in the Competitions tab allows you to export your Sport Setup data to a file, to be easily imported by another JRaceman user.

There is currently only one type of export file in this tab: Sport Setup. Select an output file, then press the Export button.

## 14.6.3. Table

The Competitions table closely matches the fields in the tab.

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this event
groupSize	int	If nonzero, the number of people in each group competing in this event. If >1, then the group field in the Entries table for events of this competition must be set to the name of the group.
maxAlternates	int	The maximum number of alternate Entries allowed in one group for this Event.
scheduledDuration	int	The time (in seconds) that this Event should be scheduled to take per Race.

## 14.7. ComplanRules

This tab is in the Plan Setup tab.

The Complan Rules tab is used to define the progression rules for a plan in a complex progression plan.

For more information on Complans, see the Complans section.

### 14.7.1. Query and Edit Tabs



- The ID field is the unique ID for this ComplanRule. The default ID prefix is **SR**.
- The Complan field is the Complan to which this record applies.
- The From Round field is the round number from which progressions are made according to this rule.
- The From Section field is the section number from which progressions are made according to this rule.
- The From Place field is the finishing place number from which progressions are made according to this rule.
- The To Round field is the round number to which progressions are made according to this rule, or zero to indicate that this finisher is part of a best-time group.
- The To Section field is the section number to which progressions are made according to this rule, or the zero-based index of the best-time group to use when the To Round field is zero.
- The To Lane field is the lane number to which progressions are made according to this rule.

### 14.7.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
system	String	Name of the system for this plan
complanId	String	The Complan of which this record is a part [ref Complans(id)]
fromRound	int	The From Round number for this rule
fromSection	int	The From Section number for this rule
fromPlace	int	The From Place number for this rule
toRound	int	The To Round number for this rule
toSection	int	The To Section number for this rule
toLane	int	The To Lane number for this rule

## 14.8. Complans

This tab is in the Plan Setup tab.

The Complans tab is used to define the plans used in complex progressions.

Complans are defined in three tables: Complans, ComplanStages, and ComplanRules.

The Complans table defines the systems of plans. It contains one row for each plan within a system of plans. All plans in the same system share the same system name; each plan within a system specifies the number of entrants for which that plan applies. There can be multiple plans with the same number of entrants.

The ComplanStages table defines how many stages there are for a plan, the round number for each stage in the plan, and how many sections there are for each stage. Each row of the ComplanStages table represents one stage in one plan.

The ComplanRules table defines how finishers advance from one round to the next. Each row of the ComplanRules table specifies how one specific finisher progresses, from a specific round, section and finishing place to a specific



round, section and lane.

### 14.8.1. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **SP**
- The System field is the name of the system of plans.
- The Plan field is the name of a plan for a specified range of entries.
- The Min Entries field is the minimum number of entries for which this is a valid plan.
- The Max Entries field is the maximum number of entries for which this is a valid plan.
- The Plan Order field provides a mechanism by which a specific plan can be selected from among multiple plans with the same entry counts. Within a group of plans with the same entry counts, each plan should have a unique number such that there is a specific order among the plans. Because a module function is applied to the plan selection numbers, the planOrder values should start at 0 and count up from there rather than starting at 1 and counting up.

### 14.8.2. Export

The Export tab in the Complains tab allows you to export your Complan data to a file, to be easily imported by another JRaceman user.

You can select one of the following export types:

- Complan Setup includes all data about all Complains.

Select an output file, then press the Export button.

### 14.8.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
system	String	Name of the system for this plan
plan	String	Name of the plan for the specified range of entries
minEntries	int	Minimum number of entries for which this plan is valid
maxEntries	int	Maximum number of entries for which this plan is valid
planOrder	int	An ordering number among all plans in the system with the same entry counts.

## 14.9. ComplanStages



This tab is in the Plan Setup tab.

The Complan Stages tab is used to define the number of sections at each Stage of a plan in a complex progression plan.

For more information on Complans, see the Complans section.

### 14.9.1. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **SS**.
- The Complan field is the Complan to which this record applies.
- The Stage field is the Stage to which the section count applies.
- The Round field is the round number to which the section count applies.
- The Sections field is the number of sections of the specified Stage and round to be created for this plan.

### 14.9.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
system	String	Name of the system for this plan
complanId	String	The Complan of which this record is a part [ref Complans(id)]
stageId	String	The Stage for which the count is being defined [ref Stages(id)]
round	int	The round number of this stage in this plan.
sectionCount	int	The number of sections to create for the specified Stage and round for this plan.

## 14.10. ContextOptions

This tab is in the Database tab.

The Context Options tab is used to define option values that are optionally tied to specific hosts, Meets, or web contexts.

### 14.10.1. Query and Edit Tabs

- The ID field is the unique ID for this option. The default ID prefix is **CTX**
- The Name field is the name of the option. For example, "Backup.directory" or "ImportRaceResults.filename".



- The Value field is the value of the option, often a filename path.
- The Host field is the name of the host computer for which this option applies. If null, this option applies to all computers running JRaceman. If specified as a name, only the computer whose name matches this field will use this option. The host name of the current computer is displayed in the Database/Options tab as Localhost.
- The Web Context field is the name of the web context for which this option applies. If null, this option applies to all web contexts. If specified as a name, only the web context whose name matches this field will use this option. This column is only used for web-specific options.
- The Meet field is the id of the Meet for which this option applies. If null, this option applies to all Meets. If set, this option applies only to the matching Meet.

### 14.10.2. Available Options

This section lists the available context options that can be set.

Name	Description
Backup.directory	The directory (folder) into which backup files are written when using the Backup button. This is typically set with a Host value to allow different users (running on different hosts) to backup to different locations on each host.
ImportRaceResults.filename	The name of the finish line file used when importing finish line data from a spreadsheet.
Meets.Awards.labelImageLeft	The name of the file to use as the left-hand image when printing labels using a format that includes a left-hand image, or the name of the file to use as the background image for label formats that include a background image.
Meets.Awards.labelImageRight	The name of the file to use as the right-hand image when printing labels using a format that includes a right-hand image.
Meets.WebReports.directory	The directory to use as the target for web reports. This overrides the Web Reports Dir field of the Meets tab.
Reports.ShowProgressionDescriptions	If set to "true" (without quotes), the description of the progression used for an Event will appear at the bottom of reports for that Event.

### 14.10.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this option
value	String	Value of this option
host	String	Name of the host computer to which this option applies
webContext	String	Name of the web context to which this option applies
meetId	String	The Meet to which this option applies [references Meets(id)]



## 14.11. Debug Change ID

This tab is in the Debug tab.

The Change ID tab allows you to change the ID of a record in a table, along with changing all references to that ID.

To change an ID, fill in the fields that specify what row to change:

- Table: the name of the database table containing the primary data.
- Old ID: the ID value identifying the row to be changed.
- New ID: the new ID value to set into that row.

After filling in the above fields, press the Change ID button. The total number of rows changed will be shown in the status line.

## 14.12. Debug EditField

This tab is in the Debug tab.

The Edit Field tab allows you to edit the contents of a single database field. This is useful for text field that contain large amounts of text when you want to change only a few characters.

To edit a field, fill in the fields that specify what field to edit:

- Table: the name of the database table containing the data
- Column: the name of the column in that table you want to change
- Where: an SQL query expression that selects a single row from the specified table

After filling in the above fields, press the Edit Field button. If the WHERE clause does not select exactly one row, an error dialog will be presented. If exactly one row is selected, a dialog will appear with the contents of the specified field of that row. After changing the contents as desired, press the OK to save your changes to the database.

## 14.13. Debug ForeignKeys

This tab is in the Debug tab.

The Foreign Keys tab assists in finding and changing data that is the target of foreign key references.

To find the records that reference a specific row, enter the table containing that row into the Table Name field, enter the ID of the row into the ID A field, and press the Show References to ID A button.

If you have part of a table displayed in the Query Results tab, you can pick out the ID from that row and put it into the ID A field by pressing the Take Selected ID to A button.

The ID B field works just like the ID A field, but with its own set of buttons.



## 14.14. Debug Generate

This tab is in the Debug tab.

The Generate tab allows you to generate pseudo-random Team and People records for testing and debugging.

Enter the number of Teams and People to create and press the Generate button. Each time you press the button, more records are generated.

The Random Seed field allows you to specify a seed value to the random number generator in order to cause it to generate the same records on two different runs. This assumes that the database on each run starts with identical Team and People entries.

## 14.15. Debug Properties

This tab is in the Debug tab.

The Properties tab allows you to view and set Java System properties.

To view all system properties, press the View Properties button. The values of all system properties will be displayed in the Report output tab.

To change the value of a system property, fill in the name and value text fields, then press the Set Property button.

## 14.16. Debug SQL

This tab is in the Debug tab.

The SQL tab allows you to enter arbitrary SQL commands. You can use this tab to make ad-hoc database queries, or to make corrections to the data to work around program bugs. Be careful when changing the database this way, as related updates which are normally handled automatically will not be made. For example, if you change the result value of a Lanes entry with an SQL command, the place values are not updated, as they are if you update it through the Finish tab.

Enter the command into the text area, then press the Execute button.

The results of SELECT commands are displayed in the Query Results tab in the lower half of the window.

### 14.16.1. Buttons

The Execute button executes the selected SQL from the SQL text area. If a part of the text has been selected using the mouse, and is thus highlighted, that text will be executed. Otherwise, the executed text will include all lines in the paragraph around the current insert point, where a paragraph extends to the nearest blank lines in either direction. In other words, you can enter multiple SQL statements into the text area as paragraphs, separated by blank lines; when you click in one of those paragraphs and push the Execute button, only that paragraph will be executed.

The Clear button clears all text from the SQL text area.

The Append Last SQL button appends the SQL text of the most recently executed SQL command to the SQL text buffer, preceded by a blank line if there is already other text in the text area. This is useful for debugging after getting an error on an SQL statement executed by the program.



## 14.16.2. Commands

In addition to entering any standard SQL statement, you can enter special commands, indicated by a leading exclamation mark. The following special commands are available:

- `!meta catalogs` - list the available catalogs
- `!meta columns [ tablePattern ] [ columnPattern ]` - list the column matching the specified column pattern (or all columns if no column pattern specified) in the tables matching the specified table pattern (or all tables if no table pattern specified)
- `!meta crossreference [ primaryTablePattern ] [ foreignTablePattern ]` - list the foreign key definitions for columns in tables matching the foreign table pattern to columns in tables matching the primary table pattern.
- `!meta exportedKeys [ tablePattern ]` - list the foreign keys in any table which refer to columns in the tables matching the table pattern
- `!meta importedKeys [ tablePattern ]` - list the foreign keys in any tables matching the table pattern which refer to columns in any table
- `!meta primaryKeys [ tablePattern ]` - list the primary keys for any tables matching the table pattern
- `!meta schemas` - list the available schemas
- `!meta tables [ tablePattern ]` - list all of the tables matching the table pattern, or all tables if no pattern specified
- `!meta tableTypes` - list the table types
- `!meta typeInfo` - list the type info
- `!set` - show all available values
- `!set catalog` - show the current catalog value
- `!set catalog name` - show the current catalog value
- `!set schema` - show the current schema value
- `!set schema name` - show the current schema value

## 14.17. Entries

This tab is in the Meet Setup tab.

The Entries tab is used to register people to participate in an Event.

### 14.17.1. Query and Edit Tabs

When creating an entry, you must select the Meet and enough information to identify the Event. You can specify the Event in one of the following ways:

- Specify the Event number in the Event# field.



- Specify the Event name in the Event Name field.

- Specify the Competition, Level, and Gender.

If there is no Event defined for the specified selection, you will not be able to create an entry.

When entering people for a group event, such as a relay race, there must be a separate Entry for each person, all with the same group name. Some of the entries for a group may be marked as alternates. If the alternate Entry is used, turn off the alternate flag in that record and turn it on in the record of the exiting group member.

- The ID field is the unique ID for this Entry. The default ID prefix is **EN**.
- The Person field identifies the person entering the event. This is a reference to the People table. In order to reference a person in an entry, you must first enter that person into the TeamSetup.People tab, then come back to this tab and select the person from the list.
- The Group field is used when this entry is for a group event. If the event is an individual event, leave this field blank.
- The Alternate field indicates that this entry is an alternate in a group. For individual events, leave this field off.
- The Scratched field indicates that this person has been scratched from the Event. Entries scratched in this way will show up in reports in strikeout text (with a line through the person's name).
- The Meet field is the Meet for this event.
- The Event# field is the Event number for this event.
- The Event name field is the Event name for this event.
- The Competition field is the Competition type of this event.
- The Level field is the Level for this event.
- The Gender field is the Gender for this event.

### 14.17.2. Sheet

The Sheet tab allows you to edit the information about multiple Entries on one page, to simplify entering or editing a group of Entries.

To edit a set of Entries, press the "Search" button. A list of all Entries will appear just below the button.

For more details on how to use the Sheet, see the section on Sheets.

### 14.17.3. Person Entries

The Person Entries tab allows you to quickly add or edit all of the Entries in a meet for one person. After selecting a Meet and a Person, push the Edit Entries button. This will open a dialog with one toggle button for each event in that Meet. Scratched Events do not appear in the list. Push on each Event button to select or deselect the Events into which the person is entered.

To see a more detailed list of events, selected the Full tab. This switches to a view with one line per events. In this view, you can sort by the various event attributes, such as level or gender, which could make it easier to find the desired event. To switch back to the check-box-only view, select the Compressed tab.

When entering a group event, you should specify a group name for that entry.



When a group event is selected, a row for that event appears in the table on the bottom half of the dialog, allowing you to enter the Group name and Alternate flag for that person in that event. Note that you must press Enter or Tab after entering a value into one of the fields of the Group table; if you don't, the value will not be stored.

If you are viewing the Full tab rather than the Compressed tab, you can enter the Group name and Alternate flag in either the top half (Full tab) or bottom half (selected Events) of the dialog.

When you are finished, press the OK button to apply the changes to the database. The number of records changed is noted in the status line.

#### 14.17.4. Table

Column	Type	Description
id	String	Unique ID [primary key]
personId	String	The person entering the event [ref People(id)]
group	String	The group including this person for this event
alternate	boolean	True if this person is an alternate for the group
scratched	boolean	True if this Entry has been scratched from the Event
eventId	String	The event entered by the person [ref Events(id)]

### 14.18. Events

This tab is in the Meet Setup tab.

The Events tab is used to define the events at a Meet. Each meet may include a different set of events.

You must select an Event when you add Entries.

#### 14.18.1. Query and Edit Tabs

- The ID field is the unique ID for this Event. The default ID is the meet ID followed by a dot, the letter **E**, and the event number. If there is no event number, then the dot is followed by the letters **EV** and an automatically generated number.
- The Meet field is the Meet for this event.
- The Event Number field is an identifying number for this event.
- The Event Name field is the name of this event. If this field is left blank, the event name is synthesized from the other fields.
- The Competition field is the Competition type of this event.
- The Level field is the Level for this event.



- The Gender field is the Gender for this event.
- The Area field is the Area where this event will be held.
- The Seeding Plan field specifies the Seeding Plan to use when drawing lanes for this Event.
- The Progression field is the Progression to use for this event when there are multiple rounds.
- The Progression State field is used by the selected progression to store information about the current state of this event. You can ignore this field during normal use of the program.
- The Scoring field is used to specify the Scoring System to use for this Event. Normally this field is left blank, in which case the Scoring System specified in the Meet for this Event is used.
- The Scratched field is used to indicate that this Event has been scratched. A scratched Event shows up in Schedule reports in strikeout text, with a line through it.
- The Event Comment field can be used to add a comment that is printed out in Schedule reports. This is typically used to explain why an Event has been scratched, but you can include any comment you wish here.

### 14.18.2. Sheet

The Sheet tab allows you to edit the information about multiple events on one page, to simplify entering or editing the events for a meet.

To edit a set of Events, select a value for the Meet of the Events whose records you want to edit, then press the "Search" button. A list of matching Events will appear just below the Meets choice.

All columns except for the first column (ID) can be edited.

To add a new Event, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. If you have selected a Meet in the meet search field, that meet will be set in the Meet column of the new row. All other fields in the new row will be blank. After you have filled in values for all required fields (competition, level, gender, and progression) an ID will be generated and the values in that row will be stored in the database.

For more details on how to use the Sheet, see the section on Sheets.

### 14.18.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
meetId	String	The meet of which this event is a part [ref Meets(id)]
number	int	The number of this event
name	String	The name of this event
competitionId	String	The type of this event [ref Competitions(id)]
levelId	String	The level of this event [ref Levels(id)]
genderId	String	The gender of participants for this event [ref Genders(id)]
areaId	String	The area to use for this event [ref Ar-



Column	Type	Description
		eas(id)]
seedingPlanId	String	The Seeding Plan to use for this event [ref SeedingPlans(id)]
progressionId	String	The progression to use for this event [ref Progressions(id)]
progressionState	String	State information for the progression on this event
scoringSystemId	String	The scoring system to use for this event [ref ScoringSystems(id)]

## 14.19. Exceptions

This tab is in the Sport Setup tab.

The Exceptions tab is used to define the names used for the different exceptions used in events. These are reasons why an entrant did not finish an event. These entrants are not assigned finishing rankings in the event, and are typically disqualified from participating in further rounds.

### 14.19.1. Query and Edit Tabs

- The ID field is the unique ID for this Exception. The default ID prefix is **X**
- The Name field is the name of the exception. For example "Did not finish" or "Did not start".
- The Short Name field is a short name for the exception. This is typically the initials of the long name, or an abbreviation. It should be small enough to fit into a results printout in place of the finishing time (or other measure). For example, "DNF" for "Did not finish"; "DNS" for "Did not start".
- The Result Allowed/Required field specifies whether a result is allowed or required in a Lane containing this Exception, as follows:
  - 0 = result not allowed
  - 1 = result is allowed but not required
  - 2 = result is required

### 14.19.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Full name for this reason
shortName	String	Short name for this reason, typically initials
resultAllowedRequired	Integer	0, 1 or 2 as result is not allowed, allowed, or required in a Lane with this Exception.



## 14.20. Finish

### Note

This tab is obsolete and has been moved into the Debug tab. Use the Results task in the By Event tab to enter the results for an Event. Use the Reports tab to generate reports for one or more Events.

The Finish tab allows you to edit the finish information for Races and to produce Lane and Result Reports. Much of the functionality of this tab is also included in the Results task of the By Event tab.

### 14.20.1. Editing Results

To edit a Race, select a Race from the Query Results tab. You can get a set of Races into the Query Results tab by doing any of the following:

- Use a Query in the the Races tab.
- Select the desired criteria for Meet, Event number, Race Number, Race Date and Round (or any subset of those fields) in the Finish tab, then press Query Races.
- Select the desired criteria for Meet and Event in the Finish tab, then press Query Events. The Race Date and Round fields are ignored. This will produce a list of Events in the Query Results tab (alternatively, you can use the Events tab to get a list of Events into the Query Results tab). Select an Event from the Query Results tab; the Get Races button will become enabled. Press the Get Races button, which will display the Races for the selected Event in the Query Results tab.

You can sort the table by any column by clicking on the header label for that column. Click again to sort in the opposite direction.

You can directly edit the value of the fields in the **Lane**, **Result** and **Exception** columns. These column are indicated by a header label in **bold font**. Depending on the state of the editing options (see Data Entry Options below), you may be able to enter values into the **Place**, **Score Place** and **Score** columns; if so, the column header will be labeled in bold font.

To edit a Lane or Result value, click on the field, type in the new value, and press Enter. The new value is stored in the database when you press Enter.

To edit an Exception value, click on the field. A choice list of Exceptions will be displayed. Click on any item, or use the arrow keys to move up or down in the list to the desired item and then press Enter.

When you enter a Result or an Exception in one member of a group (all of which are in the same lane), the same value is stored into the corresponding field of all members of that group.

You can re-sort the table after making changes by clicking on the column header again.

### 14.20.2. Data Entry Options

There are two options you can select to change the behavior of the table as you enter data.

The Auto-Update Places choice controls what happens to the Place field when data is entered into the table:

- If the Auto-Update Places choice is set to **Results By Place** or **No Auto Places**, you can edit the value of the



fields in the **Place** column. The column header becomes bold to indicate this.

- If the Auto-Update Places choice is set to **Results By Place**, then when you enter a Result in any row, the new set of Results is checked to make sure that the values match the Place values which have already been entered.
- If the Auto-Update Places choice is set to **Auto-Update Places**, then when you enter a Result or an Exception in any row, the Place values for all rows are recalculated and displayed.

The Auto-Update Scores toggle controls what happens to the Score Place and Score fields when data is entered into the table:

- If the Auto-Update Scores toggle is unchecked, you can edit the value of the fields in the **Score Place** and **Score** columns. The column header becomes bold to indicate this.
- If the Auto-Update Scores toggle is set, then when you enter a Result or an Exception in any row, the Score Place and Score values for all rows are recalculated and displayed.

### 14.20.3. Enter Places By Lane

Pressing the **Enter Places By Lane** button opens the Enter Places dialog to allow you to easily specify which lane came in first place, second place, etc. Lanes for which an Exception has already been entered will not appear in the dialog.

In the Enter Places dialog, press the lane number buttons in the same sequence as the order in which the lanes placed. For example, if lane 3 came in first place, lane 5 came in second place, and lane 4 came in third place, you would press the buttons 3, 5, and 4 in that order. Once all the lane numbers have been entered, press the Done button to save that data. The lane table will be updated to display the newly entered places sorted by place, and the Auto-Update Places choice will be set to Results By Place.

If you make a mistake while entering the lanes, press the Undo button to undo the most recent place. Press the Cancel button to dismiss the dialog without taking any action.

If you use the Enter Place By Lane button on a race for which places have already been entered, the Enter Places dialog opens with the existing places preset in the dialog. You can use the Undo button to undo the previous places just as if they have been entered in the Enter Places dialog. When you press the Done button, the new place order is saved.

### 14.20.4. Reports

To produce a report, use the row of components labeled "Report:". Select a report format, select the races on which to report, and press the Generate button.

- The Lanes report includes one row per lane, with columns for Lane, Person, Team, Place, and Result, and is sorted by Lane.
- The Finish Line report is like the Lanes report, plus it includes a finish-line results table to allow the finish line people to directly enter place-by-results, with a place to attach any printed output from timers.
- The Results report includes one row per lane, with columns for Lane, Person, Team, Place, and Result, and is sorted by Place.
- The Score report is like the Results report with the addition of a Score column.
- The Awards report is similar to the Results report, except that it only shows the first three finishers, and it does



not show the Lane. If there are non-scoring finishers in the top three, then a second section is added with those finishers, and the first section shows the first three scoring finishers, including both the scoring place and the overall place.

- The Web reports are like the other reports of similar name, except that output is written directly to the Web Reports Directory along with additional index files.

The "on" choice lets you select the set of races on which to report:

- **Selected Race** will generate a report on one Race, the Race which is currently selected, which is noted on the left side of the editing tab.
- **Matching Races** will generate a report for all Races which match the query criteria on the left side of the editing tab.
- **Query Results** will generate a report for all Races which are listed in the Query Results tab in the lower half of the screen, or all Races for all Events listed in the Query Results tab.

The "Preliminary" toggle, if set, causes the word PRELIMINARY to appear as a watermark in large dim gray letters at an angle in the background of the report.

The "Paging" choice lets you select how the resulting report is paged:

- **Single** prints one Race per page, with a page break between each race, so that each race report comes out on a separate page.
- **Double** prints two Races per page, with a page break between each pair of races.
- **Paged** prints as many reports it estimates will completely fit on a page, then outputs a page break to prevent a race from being split across two pages.
- **Packed** prints reports packed together, with no explicit page breaks. Races are likely to be split across pages. This format will print using the fewest number of pages.

## 14.20.5. Table

The data displayed in this tab are kept in the Lanes table. For information about this table, see the documentation on the Lanes tab.

## 14.21. Genders

This tab is in the Sport Setup tab.

The Genders tab is used to define the names used for the different gender groups used in events. Your events may use "Men" and "Women", or you may use "Boys" and "Girls". Some events are open to participants of both sexes. You might call these events "Mixed", or you might call them "Open".

### 14.21.1. Query and Edit Tabs

- The ID field is the unique ID for this Gender. The default ID prefix is **G**.
- The Name field is the name of the gender group.



### 14.21.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this gender

## 14.22. LaneOrder

This tab is in the Site Setup tab.

The LaneOrder tab is used to specify an order in which lanes are to be assigned during first-round draw and for Simplan and Complan progressions.

For progressions other than Simplan and Complan, the progression can supply its own preferred lane order. If no LaneOrder is specified, and an Event uses the USACK progression, then the lane order preference is 5,4,6,3,7,2,8,1,9,10,0. For other progressions, the lane order preference when no LaneOrder is specified is 1,2,3 etc.

### 14.22.1. Query and Edit Tabs

- The ID field is the unique ID for this LaneOrder. The default ID prefix is the area ID followed by a dot and the letter L.
- The Area field is the Area to which this lane ordering applies.
- The Lane field is the lane number to be ordered.
- The Order field is the order in which to assign this lane.

### 14.22.2. Sheet

The Sheet tab allows you to edit the information about multiple LaneOrder records on one page, to simplify entering or editing the lane ordering.

To edit a set of LaneOrders, press the Search button. All entries will appear in the sheet.

All columns except for the first column (ID) can be edited.

To add a new LaneOrder, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. All fields in the new row will be blank. After you have filled in values for all required fields (area, lane, order) an ID will be generated and the values in that row will be stored in the database.

For more details on how to use the Sheet, see the section on Sheets.

### 14.22.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
areaId	String	The Area to which this lane ordering



Column	Type	Description
lane	int	applies. [ref Areas(id)] The lane number to be ordered.
order	int	The ordering for this lane.

## 14.23. Lanes

This tab is in the Debug tab.

The Lanes tab is used to place and manage Entries into Races.

Lanes are usually set up automatically by a Progression. You can use this screen to view the lanes that have been set up, or to make manual changes if there is a problem with the program.

For individual races, one Lane record corresponds to the use of one lane in one race. For group races, each member of the group has a Lane record with the same lane number as the other members of the group in that race; all of these Lane records together correspond to one lane in one race.

### 14.23.1. Query and Edit Tabs

- The ID field is the unique ID for this Lane. The default ID prefix is **LN**.
- The Entry field specifies the Entry to which this Lane is attached. The Entry record refers to the entrants name via the People table.
- The Event field specifies the Event to which this Lane is attached, via the Entry and Race records.
- The Race field specifies the Race to which this Lane is attached.
- The Lane field specifies the lane number for this Lane.
- The Result field records the time, distance, or weight for this Lane.
- The Exception field records the Exception for this Lane, the reason why this competitor was disqualified.
- The Place field records the finishing place for this Lane; 1 for the first place finisher, 2 for second place, etc.
- The Score Place field records the finishing place for this Lane among competitors eligible to score; 1 for the first place scoring finisher, 2 for second place, etc. Competitors are non-scoring if they are on a Team for which the non-scoring flag has been set.
- The Score field shows the team score for this Lane.

### 14.23.2. Sheet

The Sheet tab allows you to edit the information about multiple Lanes on one page, to simplify entering or editing a group of Lanes.

To edit a set of Lanes, press the "Search" button. A list of all Lanes will appear just below the button.

For more details on how to use the Sheet, see the section on Sheets.



### 14.23.3. Table

The Person and Event are represented by the entryId field, which references an entry in the Entries table that includes these items.

The remainder of the items are stored in the Lanes table.

Column	Type	Description
id	String	Unique ID [primary key]
entryId	String	The entry included in this race [ref Entries(id)]
raceId	String	The race of which this lane is a part [ref Races(id)]
lane	int	The lane or starting position for this lane, or a negative number as a BYE or OUT indicator
result	float	The result time, distance, weight, etc. for this entrant in this race.
exceptionId	string	The reason this entrant did not finish this race [ref Exceptions(id)]
place	int	The finishing order of this entrant within the race.
scorePlace	int	The finishing order of this entrant within the race among competitors eligible to score.
score	float	The team score for this entrant in this race.

### 14.24. Levels

This tab is in the Sport Setup tab.

The Levels tab is used to define the competition levels (typically age groups) for the different events.

The age of each participant (the "qualifying age") is calculated based on the birthday field of the person's record in the People table as of the date specified as the Age Date in the Meets table. This age is then compared against the records in the Levels table to determine for which level the person qualifies.

The Minimum Age and Maximum Age values define the range of ages for the Level. People within this range are competitors at that Level. Typically the Min/Max Ages for the different Levels do not overlap, and together cover the entire age range for the sport. Sometimes there may be additional Levels based on additional criteria other than age.

The Min/Max Entry Age values are used to determine if a competitor is allowed to compete in an Event based on age. If the Levels are strictly done based on age, with no exceptions, then the Min/Max Entry Ages should be set to the same values as the Min/Max Ages. If people are allowed to compete in Levels outside of their own, the Min/Max Entry Ages should reflect that.

The Use Group Average field can be set to specify that, for group Events that use this Level, the qualifying age is the average (mean) of the qualifying age of all group members rather than the individual qualifying ages of group



members.

### 14.24.1. Query and Edit Tabs

- The ID field is the unique ID for this Level. The default ID prefix is **L**
- The Name field is the name of the level. For example, "Bantam" or "Junior".
- The Min Entry Age field is the youngest age allowed to compete in this level. If a competitor is younger than the Minimum Age but older or the same as the Min Entry Age, he is competing "up" in this Level. It is common to allow the best younger competitors to compete up in this way.
- The Minimum Age field is the youngest age included in the level.
- The Maximum Age field is the oldest age included in the level. If the level includes only one year, the minimum and maximum ages will be the same number.
- The Max Entry Age field is the oldest age allowed to compete in this level. If a competitor is older than the Maximum Age but older or the same as the Max Entry Age, he is competing "down" in this Level. Usually the Max Entry Age will be the same as the Maximum Age.
- The Use Group Average field should be set to True on Levels for which group events should use the group average age. If not set, false is assumed.

### 14.24.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this level
minEntryAge	int	Minimum age of competitors allowed at this level
minAge	int	Minimum age to belong to this level
maxAge	int	Maximum age to belong to this level
maxEntryAge	int	Maximum age of competitors allowed at this level
useAverageAge	boolean	True if group Events using this Level should use group average age

## 14.25. Meets

This tab is in the Meet Setup tab.

The Meets tab is used to define a meet. You do not need to use a new database for each meet, but can keep the data for previous meets in your database. Because all results are tied to a meet, you can easily deal only with the current meet, or you can query for information across multiple meets.

### 14.25.1. Query and Edit Tabs



- The ID field is the unique ID for this Meet. The default ID prefix is **M**.
- The Name field is the full name of the meet.
- The Short Name field is a short name for the meet.
- The Site field is the Site at which the meet takes place.
- The Start Date field is the first day of the meet.
- The End Date field is the last day of the meet.
- The Age Date field is the day on which ages of People should be calculated for the purpose of determining their Level. This is often January 1 of the current year, but can be any day. If not specified, January 1 of the same year as the Start Date is used.
- The Web Reports Directory field is the directory into which Web Reports are generated. When you select a web report, output files are created into this directory and subdirectory. Old files in this directory are silently overwritten. Web reports are available in the Schedule tab, the Finish tab, and the Score tab.

The context option `Meets.WebReports.directory` can be set, which will override the setting of the Web Reports Dir field in the Meets tab. This is useful to set up host-specific target directories for generating web reports.

- The Transfer Directory field is the directory used for transferring data to and from the automatic finish-line system using the Transfer tab.
- The Label Image Left field is used to specify the file path or URL to the image file (GIF or JPEG) to place on the left end of awards labels generated in the Award Labels tab. Some formats of award labels may use this image as a background image. The ContextOption `Meets.Awards.labelImageLeft`, if set, takes precedence over this value.
- The Label Image Right field is used to specify the file path or URL to the image file (GIF or JPEG) to place on the right end of awards labels generated in the Award Labels tab. The ContextOption `Meets.Awards.labelImageRight`, if set, takes precedence over this value.
- The Scoring field specifies the default Scoring System to use for Events in this Meet. You can override the Scoring System to use on a per-Event basis, but normally you specify the Scoring System once here for the Meet and leave it unspecified in the Events.

## 14.25.2. Export

The Export tab in the Meets tab allows you to export your Meet data to a file, to be easily imported by another JRaceman user.

You can output information about a single Meet, selected by the "Meet to Export" choice; or you can output information about all Meets by setting that choice to the first entry, which is blank.

You can choose from the following Export Types:

- Meet Setup includes the selected Meet, all Events for the selected Meet, the Areas for the Events, and the Sites for the Meet Areas and Event Areas.
- Meet Entries includes all of the Teams, People and Entries for the selected Meet. This data includes references to the data in the Meet Setup export, so you will not be able to import this data into another database unless that database includes the data from that export as well.
- Meet Results includes all of the results (Races and Lanes records) for the selected Meet. This data includes refer-



ences to the data in the Meet Setup and Meet Entries exports, so you will not be able to import this data into another database unless that database includes the data from those exports as well.

Select an output file, then press the Export button.

### 14.25.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
fullName	String	The full name of the meet
shortName	String	A short name of the meet
siteId	String	The location of the meet [ref Sites(id)]
startDate	Date	The date of the first day of the meet
endDate	Date	The date of the last day of the meet
ageDate	Date	The date for which the level of each person is to be calculated; often January 1, current year.
webReportsDirectory	String	The location of the directory into which to write web reports.
transferDirectory	String	The location of the directory used for transferring data to and from the automatic finish-line system.
labelImageLeft	String	The file path or URL for the image to use on the left side of award labels.
labelImageRight	String	The file path or URL for the image to use on the right side of award labels.
scoringSystemId	String	The Scoring System to use for Events in this Meet [ref ScoringSystems(id)].

## 14.26. Options

This tab is in the Database tab.

### 14.26.1. Editing Options

The Options tab allows you to change the values of options that control the behavior of JRaceman. To change the value of an option, find the field for the option you want to change, then click in that text field and enter the new value. When you press enter or move your mouse out of the field, the new value is stored in the database.

### 14.26.2. Options

You can change the following options in the Options tab:

- Snapshot Directory 0: the directory in which Snapshots from the Database.Backup tab are stored (for the first



line). This value can also be changed in the Database.Backup tab.

- Snapshot Directory 1: the directory in which Snapshots from the Database.Backup tab are stored (for the second line). This value can also be changed in the Database.Backup tab.
- URL to web registration page: the web location for the JRaceman servlet being used to do web registration for this database, typically of the form `http://www.yourdomain.com/servletname`.
- Email Address for web registration: the email address to use on the web registration pages when telling the user who to contact for help or special requests.
- ID of Meet for web registration: the ID of the Meet to be used by the web registration servlet.
- MailHost for web registration outbound email: The name of the host to which we send output email for web registration (such as the account creation confirmation email containing the password).
- URL to payment info for web registration: The URL to put into the web summary page that points to a page with instructions for payment.
- URL to waiver info for web registration: The URL to put into the web summary page that points to a page with instructions for waivers.
- Is web registration open?: If set to "false", closes web registration. When registration is closed, users can still log on but they can not make any changes. If blank or "true", web registration is not closed.

### 14.26.3. Table

Column	Type	Description
name	String	Name of this option [primary key]
value	String	Value of this option

## 14.27. People

This tab is in the Team Setup tab.

The People tab is used to record information about all of the people participating in a meet. Not all of the people in the table must be participating in a meet. If you use the program for multiple meets which are attended by mostly the same people, your database will already have information about those people so that you will not have to reenter it for every meet.

You will need to select a person when you set up Entries.

### 14.27.1. Query and Edit Tabs

- The ID field is the unique ID for this person. The default ID is created by taking the ID value of the person's team (which is usually the team's initials) followed by a dot plus the initials of the person's first and last name.
- The Last Name field is the person's last name.
- The First Name field is the person's first name.



- The Title field is the person's title, such as "Coach".
- The Team field indicates the Team to which this person belongs. If your sport allows independent participants, you should define a special "Team" called "Independent" or "Unattached" (or something similar), and assign the independents to that team. This will allow them to be indicated as independents in printouts, using whatever term you select.
- The Birthday field is the person's date of birth. This is used, along with the Age Date field of the Meets table, to calculate the person's age for the purposes of comparing to the Level of the Event in which that person is entered.
- The Gender field indicates the person's sex, male or female.
- The Membership field is the person's membership number in the organization sponsoring the meets.
- The Membership Expiration field holds the expiration date of the membership specified by the Membership field.
- The Team Edit field, when set to true, indicates that this person is allowed to edit the Entries for all members of his team when using the web registration interface. This is typically set for team coaches.
- The Street field is the first line of the street address. For example, "123 Main Street".
- The Street2 field is the second line of the street address. This field is typically blank.
- The City field is the name of the city.
- The State field is the name of the state or province.
- The Country field is the name of the country.
- The ZIP field is for the ZIP code or other postal code.
- The Phone field is for the primary voice telephone number associated with this address.
- The FAX field is for the FAX telephone number associated with this address.
- The Email field is the person's email address. This is used as the person's login id for web registration.
- The Web Password field is the person's password for use with the web registration system.

### 14.27.2. Sheet

The Sheet tab allows you to edit the information about multiple people on one page, to simplify entering or editing a team roster or group of people.

To edit a set of People, select a value for the Team of the People whose records you want to edit, then press the "Search" button. A list of matching People will appear just below the Team choice.

All columns except for the first column (ID) can be edited.

To add a new person, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. If you have selected a Team in the team search field, that team will be set in the Team column of the new row. All other fields in the new row will be blank. After you have filled in values for all required fields (team, last name, first name, and gender) an ID will be generated and the values in that row will be stored in the database.

To make the Sheet include all of the address fields, check the "Show Addresses" box. If this box is not checked, the



address fields will not be displayed in the Sheet, which leaves more room to see the other fields.

For more details on how to use the Sheet, see the section on Sheets.

### 14.27.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
lastName	String	Participant's last name
firstName	String	First name
title	String	This person's title (e.g. Coach, Assistant Coach)
teamId	String	[references Teams(id)]
birthday	Date	Used to determine level (age group)
genderId	String	The person's sex [references Genders(id)]
membership	int	Membership number in sponsoring organization
membershipExpiration	Date	Expiration date of the Membership
teamEdit	boolean	True indicates this person is allowed to edit Entries for all people on the same team
street	String	Street address
street2	String	Second line of street address
city	String	
state	String	
country	String	
zip	String	or other postal code
phone	String	
fax	String	
email	String	
webPassword	String	

## 14.28. Progressions

This tab is in the Sport Setup tab.

The Progressions tab is used to define the progressions used in Events.

### 14.28.1. Query and Edit Tabs

- The ID field is the unique ID for this Progression. The default ID prefix is **PR**.
- The Name field is the name of the progression.



- The Class field is the class of the progression. If no package prefix (the part before the last dot) is specified, JRaceman assumes a package of "net.jimmc.racer.progression", which is the package containing the standard progressions. See the list of Progressions.
- The Parameters field allows a string of parameters to be passed to the Class for this progression. This allows one Class to be used for multiple progressions with minor differences. Parameters are typically passed as a set of name=value pairs separated by semicolons.

## 14.28.2. Available Progressions

- ProgressionComplan - Complex Progression Plans. See the section on Complains. Allowed parameters:

Parameter	Type	Default	Description
multipleFinals	boolean	false	If true, multiple Final rounds are scheduled (as defined in the US-ACK or ICF progressions). If false, only Final A is scheduled.
planChoice	string	(none)	<p>The selection of which plan to use when multiple plans are available for the same entry counts. If only one plan is available, this parameter is ignored. This can be one of the following values:</p> <ul style="list-style-type: none"> <li>• A number, such as 0 or 1. This number is taken modulo the number of available plans and used as an index into the ordered list of plans. 0 will choose the first plan, 1 will choose the second plan, etc.</li> <li>• "entryCount" - The number of entries is taken modulo the number of available plans and used as an index into the ordered list of plans.</li> <li>• "eventId" - The Event ID is taken modulo the number of available plans and used as an index into the ordered list</li> </ul>



Parameter	Type	Default	Description
			<p>of plans.</p> <ul style="list-style-type: none"> <li>"ee0" - The Event ID and the entry count are added together and the result is taken modulo the number of available plans and used as an index into the ordered list of plans.</li> <li>"ee1" - The Event ID and the entry count are multiplied together and the result is taken modulo the number of available plans and used as an index into the ordered list of plans.</li> <li>"random" - A random number is generated between 0 and one less than the number of plans (inclusive) and used as an index into the ordered list of plans. Note that this may select a different plan if the races are redrawn, even if there are no changes in the data.</li> </ul>
system	string	(none)	<p>The name of the Complan system to use for Events which use this Progression. The Complan name can also be overridden in the Event by setting the system name in the progressionState field of the Event.</p>
useExtraLanes	boolean	false	<p>If true, races with 10 contestants are run using the extra (10th) lane. If false, races with 10 contestants will be run in two sections.</p>



- ProgressionSimplan - Simple Progression Plans. See the section on Simplans. Allowed parameters:

Parameter	Type	Default	Description
system	string	(none)	The name of the Simplan system to use for Events which use this Progression. The Simplan name can also be overridden in the Event by setting the system name in the progressionState field of the Event.

- ProgressionUsack - The USACK and ICF progressions. Allowed parameters:

Parameter	Type	Default	Description
multipleFinals	boolean	false	If true, multiple Final rounds are scheduled (as defined in the USACK or ICF progressions). If false, only Final A is scheduled.
useExtraLanes	boolean	false	If true, races with 10 contestants are run using the extra (10th) lane. If false, races with 10 contestants will be run in two sections.
useIcfProgressions	boolean	false	If true, the ICF progressions are used. If false, the USACK progressions are used.

### 14.28.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this progression
class	String	Class of this progression
parameters	String	Parameters to the Class for this progression

## 14.29. Progress

### Note



This tab is obsolete and has been moved into the Debug tab. Use the Entries/Progress task in the By Event tab to manage the progression of an Event.

### 14.29.1. Automatic or Manual Progression

There are two ways to manage progressions: automatically or manually. The Auto tab within the Progress tab allows you to do automatic progression based on the progression plan for the Event as set in the Events Edit or Events Sheet tab. Manual progression using the Manual tab within the Progress tab must be used if no progression has been set. It can also be used to override the automatic progression in case that does not behave as desired.

### 14.29.2. Automatic Progressions

The Progress Auto tab allows you to manage the progress of an Event. In this tab you control these steps:

- Create Races for the Event based on the number of Entries for that Event.
- Draw random lane assignments for the first Round of the Event.
- Progress from a completed Round to the next Round, where the advancement of competitors and their lane assignments are determined by the Progressions selected for the Event.

To manage an Event, select the desired values from the choices on the left side of the tab, then push the "Search Events" button to see the set of matching Events in the Results tab. Alternatively, you can use the Events tab in Query mode to display a set of Events in the Results tab. Select an Event from the Results tab to control the Progress of that event.

When you select an event, information about that event is displayed on the right side of the tab. The first line tells you what step needs to be done next for that Event. After this summary line are the action buttons that allow you to take the various required steps in the progression. Below the buttons is the detailed information about the state of the event.

You can get a detailed report on the progression of an event by pressing the Progress Report button. This will produce a report for the event with one row for each entrant, and one column for each round, showing the section, lane, result, and place of that entrant in that round. This format makes it easy to see who progresses from one round to the next, who is out or gets a bye, and how lanes are assigned.

### 14.29.3. The Auto Progress Action Buttons

On the right side of the Auto tab are the action buttons to control the progress of the event.

- **Create Races** creates the Races records for the Event. It counts the number of lanes required by the Entries for this event, then creates the number of Races required for that many lanes according to the Progression selected for the Event. You must ensure that all competitors are entered in the Entries table before creating Races.
- **Draw First Round** randomly assigns the competitors to lanes in the first Round of Races. You will normally select this action immediately after creating the Races. However, you may wish to manually assign some lanes (using the Lanes tab), followed by selecting the Draw First Round action to randomly assign the rest of the lanes.
- **Undraw First Round** unassigned the first-round lanes which were assigned by the Draw First Round button. This is useful when you have added more entries to an Event after you have already done the first-round draw, and you need to increase or decrease the number of Races. You can undraw the first-round lanes, change the number of Races, and re-draw the first round lanes.



- **Progress** uses the finishing places in the just-completed round to advance competitors to the next round and to do lane assignments based on the Progression selected for this Event.

## 14.29.4. Manual Progressions

To manually manage the progression for an Event, select the desired values from the choices on the left side of the tab, then push the "Query Events" button to see the set of matching Events in the Query Results tab. Alternatively, you can use the Events tab in Query mode to display a set of Events in the Query Results tab. Select an Event from the Query Results tab to manage the Progress of that event.

When you select an Event from the Query Results, the progress Sheet for that Event will appear. Work on the progression from one round of the Event by selecting a value from the "From Round" choices in the upper right corner of the tab. The sheet shows you one line for each Entry in the selected Event. For each Entry, you can select the round, section number and lane number to which each Entry will advance from the selected round. Changes are saved to the database as soon as you either type Enter or Tab to the next field. As with any other Sheet, you can sort on any column by clicking on the header for that column. For more details on how to use the Sheet, see the section on Sheets.

The "From Round" choices include the "Draw" choice, which allows you to manually assign (or "draw") the Lanes for the first round. This choice can also be used to manually edit the lane assignments done by the Draw First Round button in the Progress Auto tab.

## 14.29.5. Table

The data displayed in this tab are kept in the Races and Lanes tables. For information about these tables, see the documentation on the Races tab and the Lanes tab.

## 14.30. Races

This tab is in the Debug tab.

Races are created automatically by ao Progressions. You can use this tab to view the races that have been set up for an Event, or to make manual changes to a record if the Progression did not set up the race properly.

### 14.30.1. Query and Edit Tabs

- The ID field is the unique ID for this Race. The default ID prefix is **R**.
- The Event field is the Event for this event.
- The Stage field is the Stage for this race.
- The Round field is the round number of this race within the Event. For example, if you have three Heats, two Semifinals, and one Final, the Round for all three of the Heats would be 1, the Round of both Semifinals would be 2, and the Round of the Final would be 3. Note that the Stage and the Round will not always pair up the same way for different Events. For example, if you have only one round which is a Final, the Round for that Final would be 1, since it is the first round, which is different from the previous example.
- The Section field is the number of this race within its stage. For example, if you have three Heats, two Semifinals, and one Final, the Section for the three Heats would be 1, 2, and 3, the Semifinals would be numbered 1 and 2, and the Final would be numbered 1.



- The Area field tells what Area will be used for this race.
- The Race Number field is the number of this race. Races are typically numbered in starting-time sequence during Scheduling.
- The Scheduled Start field tells when this race is scheduled to start. You should use the Schedule tab to fill in a value for this field.
- The Scheduled Duration field tells how long to allot for this race in the schedule. You should use the Schedule tab to fill in a value for this field.
- The Actual Start field records the time that the race actually started. You should use the Schedule tab to fill in a value for this field.

### 14.30.2. Sheet

The Sheet tab allows you to edit the information about multiple Races on one page, to simplify entering or editing a group of Races.

To edit a set of Races, press the "Search" button. A list of all Races will appear just below the button.

For more details on how to use the Sheet, see the section on Sheets.

### 14.30.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
eventId	String	The event of which this race is a part [ref Events(id)]
stageId	String	The stage for this race [ref Stages(id)]
round	int	The round number of this race
section	int	The number of this race within its round
areaId	String	The area at which this race will be held [ref Areas(id)]
number	float	The number of this Race.
scheduledStart	Date	The scheduled start time for this race
scheduledDuration	int	The amount of time (in seconds) that this race is allotted in the schedule.
actualStart	Date	The actual start time for this race

## 14.31. Registration Fees

This tab is in the Meet Setup tab.

The Registration Fees tab is used to define the registration charges for the people entering a Meet.

If you are not using the Registrations. tab, you don't need to use the Registration Fees tab.



### 14.31.1. Query and Edit Tabs

When creating a Registration Fee, you must select the Meet, so this must be specified first. In other words, before you can use this tab to define registration fees, you must use the Meets tab in the Meet Setup to create a record for the meet for which you want to define charges.

- The ID field is the unique ID for this Registration Fee. The default ID is created from the Meet ID and event count used for this registration fee.
- The Meet field is the Meet in which the person is registering.
- The Event Count field specifies how many events a person must be registered for in order for this charge amount to apply. The calculation for amount charged finds the row in this table with the largest Event Count value which is the same as or less than the number of events in which the person is entered.
- The Amount Charged field specifies the total amount to charge a person who has entered the specified number of events.

### 14.31.2. Sheet

The Sheet tab allows you to edit the information about multiple Registration Fees on one page, to simplify entering or editing a group of Registration Fees.

To edit a set of Registration Fees, press the "Search" button. A list of all Registration Fees will appear just below the button.

For more details on how to use the Sheet, see the section on Sheets.

### 14.31.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
meetId	String	The meet for which the person is registering [ref Meet(id)]
eventCount	int	If a person is entered for this many events, the amountCharged from this row will be applied.
amountCharged	int	The total registration fees for a person registered for the number of events specified in the eventCount field.

## 14.32. Registrations

This tab is in the Meet Setup tab.

The Registrations tab is used to register entrants participating in a Meet.

You are not required to use the Registrations tab. JRaceman will work fine with no Registration data entered. By en-



tering Registration data into JRaceman, you can use the Registrations check, which will tell you who has not yet paid, has not yet had their membership status confirmed, or has not yet turned in a signed waiver. This same information is printed out in the Registration Summary report.

### 14.32.1. Query and Edit Tabs

When creating a Registration, you must select the Meet and the Person, so both of these must be specified first. In other words, before you can use this tab to register someone, you must use the People tab in the Meet Setup to create a record for that person with name and other details.

- The ID field is the unique ID for this Registration. The default ID is created from the IDs of the Meet and Person used for this Registration.
- The Meet field is the Meet in which the person is registering.
- The Person field identifies the person registering to participate in the meet. This is a reference to the People table. In order to reference a person in an entry, you must first enter that person into the TeamSetup.People tab, then come back to this tab and select the person from the list.
- The Amount Charged field holds the total entrance fees that must be paid by this person. Units are cents, so one dollar is entered as 100.
- The Surcharge field holds the amount of additional charges to this person. This can be used for late fees or other manual corrections to the Amount Charged.
- The Discount field holds the amount of discount credited to this person. This can be used for early discounts or other manual adjustments to the Amount Charged.
- The Amount Paid field holds the total amount that this person has paid. If the Amount Charged plus the Surcharge is less than the Amount Paid plus the Discount, the person still owes the difference.
- The Waiver Signed field tells whether the person has signed the waiver (also called a release) form.
- The Due field can be used for searching based on amount due, but is not stored in the database. In the sheet, the value displayed in this column is (amountCharged+surcharge)-(discount+amountPaid).
- The Payment notes field can be used to store a note about this person's payment, such as the amount paid and a check number or the last four digits of the credit card used.

### 14.32.2. Sheet

The Sheet tab allows you to edit the information about multiple Registrations on one page, to simplify entering or editing a group of Registrations.

To edit a set of Registrations, press the "Search" button. A list of all Registrations will appear just below the button.

For more details on how to use the Sheet, see the section on Sheets.

### 14.32.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
meetId	String	The meet for which the person is registering [ref Meet(id)]



Column	Type	Description
personId	String	The person registering [ref People(id)]
amountCharged	int	The total entrance fee for this person. Units are cents.
surcharge	int	Additional charges for this person. Units are cents.
discount	int	Credits or discounts for this person. Units are cents.
amountPaid	int	The amount this person has paid. Units are cents.
waiverSigned	boolean	True to indicate this person has turned in a signed waiver.
paymentNote	String	An optional note about this person's payment.

## 14.33. Reports.General

This tab is in the Reports tab. The Reports.General tab allows you to generate a number of different reports. Depending on which report format has been selected, you can choose a set of values to select the data on which to report.

### 14.33.1. Format Choices

On the left side of the Reports tab are a set of choice that affect the format of the report.

- The Report Format choice allows you to select the type of report you want. When you select a Report Format, a brief description of that format appears below the choices in the right half of the Reports tab.
- The Annotation choice allows you to specify a PRELIMINARY or REVISED annotation to be used as a watermark in the background of the report.
- The StyleSheet choice allows you to specify a style sheet for use by the report. Not all reports take full advantage of this feature. You can select from the available built-in styles, or you can create your own. This is discussed in more detail below.
- The Paging choice allows you to specify whether or not to generate page breaks at selected points in the report. In Single, Double or Paged mode, page breaks are inserted after one, two, or as many pieces as fit on a page; in Packed mode, no page breaks are inserted.
- The Order By choice allows you to specify the order in which data is printed in the report. The available values change based on the selected Report Format, and may be blank if there are no sorting options for that format.
- The Generate button generates the report. The output is placed into the Report tab in the bottom half of the screen, from which it can be saved or printed using commands in the Report menu.

### 14.33.2. Selection Choices

On the right side of the Reports tab are a set of choices for different types of data. When you select a Report Format, some of these choices are available and some are not. By default, unavailable selection fields are removed from the



form. If you want to see all of the fields, including those that are currently unavailable, set the "Show Disabled Selection Fields" toggle. Fields which are not editable are not used in the selected report.

You can enter values in the editable fields to select the set of records on which to report. Each field you specify further restricts the selected records on which to report. For example, if you specify both an Event Number and a Race Date, then only Races which are for the selected Event Numbers AND which are on the selected Race Date are used.

### 14.33.3. Report Styles

The general report mechanism in JRaceman is built on top of HTML, and takes advantage of HTML's CSS (Cascaded Style Sheet) ability. For the reports in the Reports.General tab and in the Reports task of the ByEvents tab you can select the style sheet to use. Alternatively, you can provide your own custom style sheet.

The standard styles available are:

- Classic - This is the style used in all versions of JRaceman through 0.9.3, and is equivalent to no CSS styling. All style properties are set to their default HTML values.
- Zebra - This style uses alternating white and light gray backgrounds for each row of the report, and reduces the report spacing to pack more data onto each page of output. Using this format allows up to 50% more data to be printed onto each page of paper. If you save the report to an external file and use Firefox (or possibly another standard web browser) to view and then print the file (with print-background-colors enabled), it will squeeze a few percent more per page.

You can define custom styles by adding the appropriate data to the ContextOptions table. You can do this manually using the Database/ContextOptions tab, or you can create a JRaceman text file and import it using the "Import JRaceman Data File" item in the Database menu.

For an example of loading a custom style from a file, see the file `data/sample-custom-style.txt`, which defines two sample custom styles:

- `custom1` - a custom style that uses an external style sheet located in `/tmp/custom.css`.
- `custom2` - a custom style that uses an internal style, which gets included into the header of the report when it is generated.

You can either load this file then edit the data using the Database/ContextOption tab, or you can copy the file to your own location, edit the file, and then load the edited file.

To create a custom style by hand, you need to create a Context Option called `Report.Style.YourCustomName.css`, where *YourCustomName* is replaced by whatever name you want to appear in the Style choice list. The name must not include any spaces or punctuation characters, except that it may include period, dash, and underscore. The value for that Context Option should be a valid piece of HTML, which will be inserted into the report header. That HTML snippet should be either a `<style>` element for an internal style, or a `<link>` element for an external style. Your custom style names will appear in the Style choice list after all of the standard choices.

When you add or change custom style names, the Style choice list in the Reports.General tab will automatically update to reflect that change the next time the tab is selected. If you are viewing that tab when you import a file that makes changes to the custom style names, you must select another tab and reselect your tab in order to make the Style choice list update to reflect your changes.

When creating a custom style sheet, you can create styles for the following class names used in the reports:

- `bottom` - The part of the header or footer that follows the `jracmanDateLine`. This area contains three subareas with class names `left`, `center`, and `right`.
- `center` - The center element within a top or bottom element.



- footer - The footer area of the report, after the main table. This area contains subareas with the same three class names as the header class: top, jracmanDateLine, and bottom.
- header - The header area of the report, before the main table. This area contains subareas with the same three class names as the footer class: top, jracmanDateLine, and bottom.
- jracmanAnnotatedDateLine - The line containing the date the report was printed (actually generated) and the jracman.org notice when it includes an annotation.
- jracmanDateLine - The line containing the date the report was printed (actually generated) and the jracman.org notice.
- jracmanFooter - A line containing JRaceman info.
- left - The leftmost element within a top or bottom element.
- main - The main data table or area.
- preFooter - An area in a few reports that comes after the main area and before the footer area.
- preHeader - An area in a few reports that comes before the header area.
- right - The rightmost element within a top or bottom element.
- rowFinishAttachment - A row in the Race Finish report that includes a place to attach a results printout.
- rowFinishAttachmentSpace - A row in the Race Finish report that provides a spacer above a rowFinishAttachment row.
- rowHeader - A header row in the main table.
- rowParity0, rowParity1 - these two classes are used on alternating rows of the table. They can be used to implement alternating background colors as us used in the Zebra style sheet.
- rowProgressionInfo - a row under the main table in a few reports that includes the progression info for the Race.
- rowSep - A row in one of the Race reports, such as Race Lanes, that separates the rows containing the entrant data.
- rowSepUnderHeader - The separator row that appears just under the table header in some reports.
- rowTotal - A total row, typically the last row of the main table.
- titleArea - The area containing the title information. This preceeds the main area but follows the header.
- top - The part of the header or footer that preceeds the jracmanDateLine. This area contains three subareas with class names left, center, and right.

## 14.34. Schedule

This tab is in the Meet Setup tab.

### 14.34.1. Creating Races

Use the Schedule tab to create Race records for your Events. Select a set of Events by entering values for the Meet and optionally Event# fields, then press the Query Events button. The set of matching Events will appear in the



Query Results tab at the bottom of the window.

To create Races for one Event, select that Event from the Query Results tab. The Create Races area (with the Search Event Races and Create Races button) will be filled in with some information about the selected Event. The entries/groups text box in that area will be filled with the number of entries or the number of groups for the selected Event. This is the number of Lanes that are needed for the first round of the Event. If you want to use a different number (such as if you expect to receive more Entries), enter a different number into that box. Press the Create Races button to create the Races. If there are already Races for the Event, a dialog will appear describing the existing Races and what needs to be done. If there are already results entered for the Event, you will not be allowed to recreate the Races; if you must do so, you first must manually delete the Lanes records.

Before creating Races for multiple Events (see below), you may want to scratch the Events for which there are no Entries. To do so, select the Meet (and optionally a set or range of Events) and select the Query Matching Events With No Entries menu item from the Schedule menu. The results of this query are displayed in the Query Results tab. You can then select the Scratch All Query Results Events menu item from the Schedule menu. To see the changes, select the Query Matching Events With No Entries menu item again.

To create Races for multiple Events, enter the query values for the desired Events into the Meet and Event# boxes (or the other query boxes near them). You can preview the list of matching Events by pressing the Query Events button, which lists the matching Events in the Query Results tab. If this list is correct, select the Create Races for Matching Events command from the Schedule menu (which is only visible when the Schedule tab is selected). This will create Races for all of the matching Events. For each Event, it will use the number of entries or groups for that Event (the same number as is placed in the entries/groups text box when you select that Event from the Query Results list).

After you have created the Races for some or all Events, you can draw the Lane assignments for those Races. Using the same Event selection values as for the Create Races for Matching Events menu command, select the Draw First Round for Matching Events command from the Schedule menu. This will do the lane assignments for the first-round Races for all of the matching Events. For Events in which the first-round Lane assignments have already been done, there is no change. For Events in which some first-round Lanes have been assigned, but new Entries have been added since then, redrawing the first round will leave the existing Lane assignments alone, but will add Lane assignments for the new Entries.

Race creation and Lane assignment are not done for scratched Events.

### 14.34.2. Adding Special Items

You can include non-race items in the schedule. These are represented in the database as records in the Races table associated with special Events whose number is less than zero. After adding special items to your schedule, you may see these negative event numbers, along with your special item names, when you do queries which return a set of Events.

To create a special item, select the Add Special to Schedule command from the Schedule menu. If there are not yet any special event types, this command will open the Add Event Type dialog (see below). If there are already some special event types, this command will open a dialog allowing you to select the kind of special entry you want to appear in the Schedule, as well as the start time and race number for that special item, either of which can be left blank. The dialog allows you to choose from any of the existing special event names that you have previously created, or to create new special event names (see below). After selecting one of the event types and optionally choosing a value for the start time and race number, push the Add Special button. This will create a special record of that type. Press the Search Races button to see the newly added item in the Races Sheet. You can then edit the Start Time of that special item the same as for a regular Race. You do not need to set a race number on special items, but you may find it useful to do so if you are creating the special items before you have set times in the schedule. You can create multiple special schedule items of the same type.

To create a new type of special item, press the Add Event Type button in the dialog. In the next dialog, enter the special item name you want to use. Typical entries are "Break" or "Lunch". Press the OK button to create this new event type and return to the Add Special dialog.



If you have selected a Race in the Races editing sheet, and that Race has a Scheduled Start, then the Add Special dialog will include a default value for its Date and Time field. That default value will be the value of the selected Race plus the number of minutes specified in the "F6 auto-increment minutes" field (or plus one minute if that field is blank), as long as that value is still before the next item in the schedule. If not, then the default time will be half way between the time of the selected Race and the next item in the schedule. In addition, there will be a button next to Race Number field that will allow you to quickly fill in a value which is greater than the selected Race but less than the next higher assigned number.

### 14.34.3. Sheet

The Schedule tab allows you to edit the scheduling information for Races and produce a Schedule Report.

To edit a set of Races, select values for the Meet and the Area of the Races you want to schedule, then press the Search Races button. A list of matching races will appear in an editing sheet just below the Meet and Area choices.

You can directly edit the value of the fields in these columns:

- Scheduled Start
- Scheduled Duration
- Actual Start
- Race Number
- Race Scratched

For more details on how to use the Sheet, see the section on Sheets.

To set the comment for a scratched Race, select the Race from the sheet, then select the Set Comment for Race command from the Schedule menu (this command is only available when a Race is selected). Fill in the dialog with your comment and press OK. The comment will appear after the Race name in the Schedule reports.

### 14.34.4. Using Auto-Increment

The F6 key can be used to simplify entering times and renumbering races. Start by entering race numbers for all of your races. If you have to squeeze a race in between two others, just use a decimal number between the two other numbers. For example, to move a race to be in between races 4 and 5, you could give it the number 4.1; to fit another between 4 and 4.1, you could number it 4.05.

After the races are numbered, click on the Race Number column header to sort the races, so that the lowest numbered races appear first. Click on the Scheduled Start field of the first race and enter the date and time for that race. At the top right of the form, enter the number of minutes between races in the text box labeled "F6 auto-increment minutes". Click on the Scheduled Start field of the second race, but don't double-click it (if you do, press Escape to get out of field-edit mode). Press the F6 key; this will add the auto-increment minutes to the value above the selected box, set this value into the selected box, and move down one row. You can quickly press F6 multiple times to increment and fill in the values for the races. If there is a break in the schedule or some other reason why the next race is not the same number of minutes later than the last one, simply enter in the value for that different race, then continue with the F6 procedure below that race.

You can easily renumber the races by using a similar technique. Sort the races by number by clicking on the Race Number column so that the lowest number race appears at the top of the column. Click in the Race Number field of the first row and enter the new race number for that Race. Click in the Race Number field of the second race, but don't double-click to edit it. Press the F6 key; this will set the race number of the selected race to one greater than the race number of the row above it, and move down one row. You can quickly press F6 multiple times to set each



race to the next number after the race above it. Note that this procedure is just like setting the race times, except that you can't select the increment values for race numbers, they always increment by one.

### 14.34.5. Creating a Report

To create a Schedule Report, select the desired report format and press the Generate button to the right of the format choice.

The With Area and Without Area reports include one row for each race in an event, sorted by scheduled start time. In addition to the event and stage information which identifies a race, each row includes the scheduled start time, the actual start time, the scheduled duration, and the race number. The With Area report includes the area for that race. These formats are suitable for printing as worksheets, or for posting to the web.

The Two Column report prints the race number, event number and name, and scheduled start time for each event, in two columns down the page. The date is printed only when it changes between races. This format is suitable for printing and distributing.

The Web Per-Day report writes a set of web pages to the "schedule" subdirectory of the Web Reports Directory (see Meets ). It writes one page for each day in the start/end range of the selected meet, plus a schedule index page. Note that races with a date outside of the start/end date of the meet will not be included in any schedule with this report.

### 14.34.6. Table

The data displayed in this tab are kept in the Races table. For information about this table, see the documentation on the Races tab.

## 14.35. Score

This tab is in the Reports tab.

The Score tab allows you to view team scores.

To see scores, select a value for the desired Meet, select a report format, then press the Generate button. If you have selected a non-Web report format, a Score Report will appear in the Report output tab. Web report formats are the same as the non-web version, but are generated directly into the "scores" subdirectory of the web reports directory (see the Meets tab).

### 14.35.1. Reports

- Teams - a list of teams and their scores, one line per team, sorted in score order with the highest team score first.
- People - a list of people and their scores, one line per person, sorted in score order with the highest personal score first.
- Details - a report giving the detailed scores, one line for each person and event with a score, sorted by team and person, with additional total lines for each team or person with more than one score item. You can use this to see the source for all points.
- Teams and Levels - a table with teams listed on the left side and levels listed across the top, showing the points earned at each level by each team. The teams are sorted by total score, with the highest score at the top. The levels are sorted by minimum age, with the lowest age level on the left. The right hand column gives the total across all levels for each team. The last row gives the total across all teams for each level. The lower right corner give the total number of points across all teams and all levels.



- The Web reports are like the other reports of similar name, except that output is written directly to the Web Reports Directory along with additional index files.

## 14.35.2. Table

For information about the Lanes table, see the documentation on the Lanes tab.

## 14.36. Scoring

This tab is in the Sport Setup tab.

The Scoring tab is used to define the scoring systems available for use in a Meet.

Scoring is defined in two tables: ScoringSystems and ScoringRules.

The ScoringSystems table defines the different available scoring systems. It contains one row for each available scoring system.

The ScoringRules tables defines the set of rules that make up the scoring system. Each rule specified how many points are received for specified values of important attributes, such as finishing place.

### 14.36.1. Systems Sheet

The Systems Sheet tab allows you to edit the information about multiple ScoringSystems on one page.

To edit a set of ScoringSystems, press the "Search" button. A list of ScoringSystems will appear just below the buttons.

All columns except for the first column (ID) can be edited.

To add a new Scoring Rule, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. All fields in the new row will be blank. After you have filled in values for all required fields (name) an ID will be generated and the values in that row will be stored in the database.

For more details on how to use the Systems Sheet, see the section on Sheets.

### 14.36.2. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **ScR**.
- The System field is the Scoring System to which this rule applies.
- The Rule field is the name of the rule type for this record. It must be one of the following keywords: place, group.
- The Value field is the value of the rule property for which the specified number of points are given. For example, a value of 1 for a "place" rule means first place, 2 means second place, etc. For a "group" rule, the value is the size of the group.
- The Points field is the number of points that are awarded to a Lane that satisfies this rule.



### 14.36.3. Sheet

The Sheet tab allows you to edit the information about multiple ScoringRules on one page, to simplify entering or editing the rules for a scoring system.

To edit a set of ScoringRules, press the "Search" button. A list of ScoringRules will appear just below the buttons.

All columns except for the first column (ID) can be edited.

To add a new Scoring Rule, push the Add button. This will add a new row to the end of the editing sheet and scroll the sheet to the end. All fields in the new row will be blank. After you have filled in values for all required fields (system, rule, value, and points) an ID will be generated and the values in that row will be stored in the database.

For more details on how to use the Sheet, see the section on Sheets.

### 14.36.4. Table

Column	Type	Description
id	String	Unique ID [primary key]
scoringSystemId	String	The scoring system of which this rule is a part [ref ScoringSystems(id)]
rule	String	The rule name for this entry.
value	int	The parameter for the specified rule type.
points	float	The number of points to assign for this rule.

## 14.37. Seeding Lists

This tab is in the Team Setup tab.

The Seeding Lists tab is used to set up seeding lists associated with a seeding plan. Each list is associated with one seeding plan.

See the section on Seeding Plans for more information about seeding plans.

### 14.37.1. Query and Edit Tabs

- The ID field is the unique ID for this Seeding. The default ID is the seeding plan id, a dot, and the seeding rank.
- The Seeding Plan field is the name of the seeding plan. This is the value specified in the Event record in order to use this seeding list when drawing lanes for that Event.
- The Rank field controls the seeding order. Lower numbers are seeding first when drawing lanes, so they are assigned better lanes. If two seedings within the same list are specified with the same rank, they are order by the ID value of the seeding.
- The Person field specifies the person to be seeded. Any people in the seeding list who are not entered into a seeded Event are ignored for that Event. If you want, you can set up one seeding list that ranks all of the entrants



to your Meet and use that one seeding list for all Events.

### 14.37.2. Sheet

The Sheet tab allows you to edit the information about multiple seedings on one page, to simplify entering or editing a seeding list.

To edit a seeding list, select a value from the List Name choice, then press the "Search" button. A list of all Seeding Lists will appear just below the button.

All columns except for the first column (ID) can be edited.

For more details on how to use the Sheet, see the section on Sheets.

### 14.37.3. Table

The SeedingLists table closely matches the fields in the tab.

Column	Type	Description
id	String	Unique ID [primary key]
seedingPlanId	String	[references SeedingPlans(id)]
rank	int	The seeding rank of this person within this list.
personId	String	The person for this seeding [ref People(id)]

## 14.38. Seeding Plans

This tab is in the Team Setup tab.

The Seeding Plans tab is used to set up plans for seeding to seed lane assignments for Events. Multiple seeding plans can be defined. The seeding plan is independently specified for each Event to be seeded. One seeding plan can be used for multiple Events, or each Event can reference a separate seeding plan.

You can specify that an Event is to be seeded by setting the Seeding Plan in the record for that Event. When drawing lanes for that Event, the specified Seeding Plan will be used to determine the order in which entrants are assigned to lanes. Any entrants found in the seeding list associated with the specified plan will be assigned lanes first, in rank order starting with the lowest rank number into the preferred lanes. After all seeded entries have been assigned lanes, all other entries are assigned lanes in random order.

For group events, the rank for the group is calculated by taking the average (mean) of the rank for all members of the group, including alternates. If there are no seeded entries in the group, the group is an unseeded group and is assigned a lane in random order after all seeded groups have been assigned preferred lanes. If there are some seeded and some unseeded entries in a group, all unseeded entries in that group are treated as having been given a seeding rank of one larger than the largest seeding rank defined in the list being used (so are thus seeded at the end of the list), after which the group average rank is calculated.

### 14.38.1. Query and Edit Tabs



- The ID field is the unique ID for this Seeding. The default ID prefix is "SeedP".
- The Name field is the name of the seeding plan.

### 14.38.2. Sheet

The Sheet tab allows you to edit the information about multiple seeding plans on one page.

To edit a seeding plan, press the "Search" button. A list of all Seeding Plans will appear just below the button.

All columns except for the first column (ID) can be edited.

For more details on how to use the Sheet, see the section on Sheets.

### 14.38.3. Table

The SeedingPlans table closely matches the fields in the tab.

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of the seeding plan

## 14.39. SimplanRules

This tab is in the Plan Setup tab.

The Simplan Rules tab is used to define the progression rules for a plan in a simple progression plan.

For more information on Simplans, see the Simplans section.

### 14.39.1. Query and Edit Tabs

- The ID field is the unique ID for this SimplanRule. The default ID prefix is **SR**.
- The Simplan field is the Simplan to which this record applies.
- The From Stage field is the Stage from which progressions are made according to this rule.
- The To Stage field is the Stage to which progressions are made according to this rule.
- The Thru Place field tells which places advance according to this rule.
- The Next Best Times field tells how many additional entrants advance from the place after the Thru Place.

### 14.39.2. Table

Column	Type	Description
id	String	Unique ID [primary key]



Column	Type	Description
system	String	Name of the system for this plan
simplanId	String	The Simplan of which this record is a part [ref Simplans(id)]
fromStageId	String	The From Stage for this rule [ref Rules(id)]
toStageId	String	The To Stage for this rule [ref Rules(id)]
thruPlace	int	The places which advance according to this rule
nextBestTimes	int	The number of additional entrants that advance according to this rule

## 14.40. Simplans

This tab is in the Plan Setup tab.

The Simplans tab is used to define the plans used in simple progressions.

Simplans are defined in three tables: Simplans, SimplanStages, and SimplanRules.

The Simplans table defines the systems of plans. It contains one row for each plan within a system of plans. All plans in the same system share the same system name; each plan within a system is for a different number of entrants.

The SimplanStages table defines how many stages there are for a plan, and how many sections there are for each stage. Each row of the SimplanStages table represents one stage in one plan.

The SimplanRules table defines how finishers advance from one round to the next. Each row of the SimplanRules table relates a from-round and a to-round within one plan.

### 14.40.1. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **SP**
- The System field is the name of the system of plans.
- The Plan field is the name of a plan for a specified range of entries.
- The Min Entries field is the minimum number of entries for which this is a valid plan.
- The Max Entries field is the maximum number of entries for which this is a valid plan.

### 14.40.2. Export

The Export tab in the Simplans tab allows you to export your Simplan data to a file, to be easily imported by another JRaceman user.

You can select one of the following export types:



- Simplan Setup includes all data about all Simplans.

Select an output file, then press the Export button.

### 14.40.3. Table

Column	Type	Description
id	String	Unique ID [primary key]
system	String	Name of the system for this plan
plan	String	Name of the plan for the specified range of entries
minEntries	int	Minimum number of entries for which this plan is valid
maxEntries	int	Maximum number of entries for which this plan is valid

## 14.41. SimplanStages

This tab is in the Plan Setup tab.

The Simplan Stages tab is used to define the number of sections at each Stage of a plan in a simple progression plan.

For more information on Simplans, see the Simplans section.

### 14.41.1. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **SS**.
- The Simplan field is the Simplan to which this record applies.
- The Stage field is the Stage to which the section count applies.
- The Sections field is the number of sections of the specified Stage to be created for this plan.
- The Fill Order field tells the order in which the sections in this stage are filled for a draw or progression. The choices are list here, each with an example showing the section ordering for that Fill Order when there are four sections. order of sections used w
  - SEQ = Sequential: fill up the section before moving on to the next one. 1,1,1,1,..., 2,2,2,2,..., 3,3,3,3,..., ...
  - SEQ\_SEQ = Sequential/Sequential: same as SEQ, but fill the lanes within each section in sequential order as well (the lanes specified in the Lane Order are sorted).
  - RR = Round Robin: 1,2,3,4, 1,2,3,4, 1,2,3,4, ...
  - SNAKE = Snake (or serpentine; boustrophedonic) 1,2,3,4, 4,3,2,1, 1,2,3,4, 4,3,2,1, ...
  - ZZ = Zig-Zag (like Round Robin but offset by one on each pass): 1,2,3,4, 2,3,4,1, 3,4,1,2, 4,1,2,3, 1,2,3,4, ...



## 14.41.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
system	String	Name of the system for this plan
simplanId	String	The Simplan of which this record is a part [ref Simplans(id)]
stageId	String	The Stage for which the count is being defined [ref Stages(id)]
sectionCount	int	The number of sections to create for the specified Stage for this plan.
fillOrder	String	The order in which sections are filled for draws or progressions. One of the short upper-case strings defined above.

## 14.42. Sites

This tab is in the Site Setup tab.

The Sites tab is used to define the sites at which a meet can take place. One site can contain multiple Areas at which different events can occur simultaneously.

You will need to select a Site when you set up records in the Areas table.

### 14.42.1. Query and Edit Tabs

- The ID field is the unique ID for this Site. The default ID prefix is **SI**.
- The Name field is the full name of the site.
- The Street field is the first line of the street address. For example, "123 Main Street".
- The Street2 field is the second line of the street address. This field is typically blank.
- The City field is the name of the city.
- The State field is the name of the state or province.
- The Country field is the name of country.
- The ZIP field is for the ZIP code or other postal code.
- The Phone field is for the primary voice telephone number associated with this address.
- The FAX field is for the FAX telephone number associated with this address.

### 14.42.2. Table



Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of the site
street	String	Street address
street2	String	Second line of street address
city	String	
state	String	
country	String	
zip	String	or other postal code
phone	String	
fax	String	

## 14.43. Stages

This tab is in the Sport Setup tab.

The Stages tab is used to define the names used for the different stages of progression in events. Typical names are "Heat" and "Final". You may also have "Semifinal", "Quarterfinal", or any other name.

### 14.43.1. Query and Edit Tabs

- The ID field is the unique ID for this Stage. The default ID prefix is **S**.
- The Name field is the name of the stage.
- The Number field is the number of the stage.
- The Is Final field is true for the final stage.

### 14.43.2. Table

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Name of this stage
number	String	Number of this stage
isFinal	Boolean	True if this is the final stage

## 14.44. Teams

This tab is in the Team Setup tab.

The Teams tab is used to define all of the teams that are participating in a meet. Not all of the teams in the table must be participating in a meet.



You will need to select a Team when you set up records in the People table.

### 14.44.1. Query and Edit Tabs

- The ID field is the unique ID for this Team. The default ID prefix is the team's short name.
- The Short Name field is a short name for the team, typically the initials or an abbreviation. This is typically used in event printouts to succinctly identify the team for each entrant in an event. If this field is blank when a new event is saved, it is filled in with the team's initials as extracted from the Name field.
- The Number field allows you to specify a number to be associated with each team. This field is optional.
- The Name field is the full name of the team.
- The Challenge field allows you to specify a Challenge category name for this team. This field is optional.
- The NonScoring field is used when a team is being allowed to participate in a meet, but is not to be given any team points; for example, when an international team is allowed to compete in a national competition as a courtesy, but will not be given any awards.
- The Street field is the first line of the street address. For example, "123 Main Street".
- The Street2 field is the second line of the street address. This field is typically blank.
- The City field is the name of the city.
- The State field is the name of the state or province.
- The Country field is the name of the country.
- The ZIP field is for the ZIP code or other postal code.
- The Phone field is for the primary voice telephone number associated with this address.
- The FAX field is for the FAX telephone number associated with this address.

### 14.44.2. Sheet

The Sheet tab allows you to edit the information about multiple teams on one page, to simplify entering or editing a group of teams.

To edit a set of Teams, press the "Search" button. A list of all Teams will appear just below the button.

All columns except for the first column (ID) can be edited.

If the Short Name field is blank when you enter a value into the Name field, it is automatically filled in using the initials of the Name, the ID is generated from the short name, and the row is stored. If you want to specify the Short Name, fill in that field before filling in the Name field.

To make the Sheet include all of the address fields, check the "Show Addresses" box. If this box is not checked, the address fields will not be displayed in the Sheet, which leaves more room to see the other fields.

For more details on how to use the Sheet, see the section on Sheets.



### 14.44.3. Export

The Export tab in the Teams tab allows you to export your Team data to a file, to be easily imported by another JRaceman user.

You can output information about a single Team, selected by the "Team to Export" choice; or you can output information about all Teams by setting that choice to the first entry, which is blank.

You can select one of the following export types:

- Team Roster includes the selected Team, all People for the selected Team, and all Addresses referenced by the selected Team or People.
- Team Entries includes the information in Team Roster plus all of the Entries for the People on the selected Team.

Select an output file, then press the Export button.

### 14.44.4. Table

The Teams table closely matches the fields in the tab.

Column	Type	Description
id	String	Unique ID [primary key]
name	String	Full name of the team
shortName	String	Short name for the team, typically initials
number	int	Number for the team
challengeId	String	[references Challenges(id)]
street	String	Street address
street2	String	Second line of street address
city	String	
state	String	
country	String	
zip	String	or other postal code
phone	String	
fax	String	

### 14.45. Transfer

This tab is in the Database tab. The Transfer tab allows you to interoperate with an automated finish-line system. The only system currently supported by this tab is the FinishLynx [<http://www.finishlynx.com>] system. (The Omega system is supported with a different mechanism, see the By Event tab.)

In order to use the Transfer tab, you must fill in a value for the Transfer Directory field of the Meets tab. JRaceman uses a single directory for both input and output. Configure your FinishLynx system to use that same directory for both input and output.



### 14.45.1. Sending Data

To send data to the finish-line system, select a Meet (and optionally other values) on the left hand side of the tab, fill in the "to" filename field with the desired base name (you can usually leave it set to the default value) and press the Send button. This will create two files in the Transfer directory of the selected Meet with the base name as specified in the "to" field and the extensions `.sch` and `.evt`. These files contain the schedule and event information, respectively, for all selected races.

You can only send races with a race number. When you press the Send button you will be notified about any races in your selection which do not have race numbers.

### 14.45.2. Receiving Data

To receive data from the finish-line system, select a Meet on the left hand of the tab, fill in the "from" filename field with the desired base name pattern (you can usually leave it set to the default value) and press the Receive button. This will read all files from the Transfer directory which match the specified base name pattern, enter their data into the database, and rename the file to add a `.done` extension to indicate that the file has been read.



## Chapter 15. Output Tabs

These are the output tabs in the bottom half of the main window. They are listed here in alphabetical order.

### 15.1. Help

#### 15.1.1. The Help Tab

The Help tab displays help information from various sources:

- When you first run JRaceman, the Welcome page is display in the Help tab.
- The first time you visit a new editing tab after starting JRaceman, the Help tab is raised and the help information for that tab is displayed. You can see the help information for a tab at any time by selecting the "Help On This Tab" item from the Help menu.
- Selecting any item from the Help menu (exception Show Location) raises the Help tab and displays the selected information in that tab.

#### 15.1.2. The Help Menu

The Help menu contains the following items:

- **Tutorial Wizard** opens a Wizard dialog to step you through the basics of JRaceman. The Tutorial Wizard opens automatically the first time you run JRaceman.
- **Welcome** displays the Welcome page, the same page as is displayed when the program is first started.
- **User Guide** displays the table-of-contents page of the JRaceman Users Guide.
- **Help On This Tab** displays help about the currently displayed editing tab, the same page as is displayed the first time that tab is visited after starting the program.
- **Help On Help** displays this help page.
- **Open Page...** allows you to enter the URL of any page to display in the Help tab.
- **About** is a pull-right (nested) menu that contains commands to display version, date, licensing and copyright information about JRaceman and its components.
- **Back** displays the previous page in the Help history list. You can use this to back up after following links.
- **Forward** displays the next page in the Help history list. You can use this after using the Back command.
- **Show Location** shows the location of the currently displayed help page in the Status Line. This is useful if you want to use another browser to view the page.
- **Print...** opens the Print dialog to allow printing the contents of the Help tab.

#### 15.1.3. Help File Navigation

The Help tab functions much like a little web browser. Links are displayed as underlined and colored text. You can



follow a link to another help page by moving the pointer over the underlined text and clicking the left mouse button.

The Help tab maintains a history of help pages. You can move backwards through this history to revisit pages, then forward again, by using the Back and Forward items in the Help menu.

The Help pages are HTML, so you can use your favorite web browser to look at them. To see the location of the page being displayed in the Help tab, select the Show Location item from the Help menu. This will display the location in the Status Line, which you can then type (or copy and paste) into your favorite browser. If you have problems selecting the text from the Status Line, you can select the Messages tab and copy it from there.

## 15.2. Messages

The Messages tab displays status and information messages. You can scroll back through this tab to see earlier messages. When a message appears, the Messages tab is automatically raised.

Each message in the Messages tab includes the time and the source tab of the message.

## 15.3. Query Results

The Query Results tab displays the results of searches from any of the Edit tabs. There is only one Query Results tab, so each time a search is performed from any Edit tab, the previous search results are discarded.

When the Query Results tab contains the results of a query from an Edit tab, the name of that tab is added to the name of the Query Results tab. The number of items currently displayed in the Query Results tab is included in square brackets after the Edit tab name.

When there is data displayed in the Query Results tab, you can operate on the data with the commands in the Query Results menu.

## 15.4. Report

The Report tab displays reports generated by the program.

When there is data displayed in the Report tab, you can operate on the data with the commands in the Report menu.



## **Part IV. Appendices**



## Appendix A. Software License

JRaceman is licensed under the GNU General Public License (GPL), Version 2 of June 1991, the full text of which appears below. The GPL is also available on-line [<http://www.gnu.org/copyleft/gpl.html>].

### A.1. GNU General Public License

GNU                      GENERAL                      PUBLIC                      LICENSE  
Version                      2,                      June                      1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU                      GENERAL                      PUBLIC                      LICENSE  
TERMS      AND      CONDITIONS      FOR      COPYING,      DISTRIBUTION      AND      MODIFICATION



0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.



3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent



license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO

## WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER



PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name> of <author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.



## Appendix B. Debugging and SQL

JRaceman is designed such that each operation reads data from the database, modifies that data according to the operation, and writes the modified data back to the database. This means the data used for all calculations are visible in the database, and the results of all calculations can be manually modified by editing the contents of the database after the calculation is complete. While editing the database directly is not the preferred way to operate, the ability to do so in emergencies means that at least there is something that can be done when the software is not behaving quite as expected or desired.

When manual editing of the database is required, using a standard editing form in one of the Edit tabs is the preferred technique. See the tabs chapter for a list of tabs in JRaceman, or use the **Edit** command in the **Tabs** menu to see a list of the Edit tabs.

For more powerful direct editing of the database, you can enter SQL commands directly into the Debug->Sql tab.

Since JRaceman is built on top of Mckoi Database, you can enter any valid Mckoi SQL command into the SQL window. A complete description of SQL is beyond the scope of this manual; you can find the documentation for Mckoi SQL on-line [<http://www.mckoi.com/database/docindex.html>] .

See also the Debug/Sql documentation for commands that can be entered in addition to the Mckoi SQL commands.

### B.1. SQL Primer

SQL has a large number of data types, but JRaceman only uses four types of data: string, integer, floating, and boolean.

String values are quoted using single-quote character ('). Integers are entered using decimal digits (0 to 9) with an optional leading negative sign (-). Floating numbers are entered as integers optionally immediately following by a decimal point (.) and decimal digits (0 to 9). Boolean values are entered as the word `true` or the word `false`.

There are four basic SQL commands: SELECT, INSERT, UPDATE, and DELETE. The SELECT command retrieves data from the database; the other three commands modify the database.

In the descriptions below, the use of square brackets ([]) indicates that the contents of the square bracket are options. The use of an asterisk (\*) after the square brackets indicates that the contents of the square brackets can be repeated zero or more times.

The format of the basic SELECT command is

```

SELECT
FROM
[
WHERE
]
[
ORDER
BY
]
columns
table
condition
sort-columns

```

For example, to see minAge and maxAge values in all rows of the Levels table, you would enter the SQL command

```
SELECT      minAge,      maxAge      FROM      Levels
```

To select all columns from all rows of the Competitions table, and to order by the name column, you would enter the SQL command



```
SELECT      *      from      Competitions      ORDER      BY      name
```

The format of the basic INSERT command is

```
INSERT      INTO      table
              (
                column-names
              )
              VALUES
              (
                values
              )
```

For example, to insert a new Level, you could enter the SQL command

```
INSERT      INTO      Levels(id,name,minAge,maxAge)
              VALUES( 'L9', 'Midget',10,12)
```

The format of the basic UPDATE command is

```
UPDATE      table
              SET
              column-name
              =
              value
              [
                ,
                column-name
                =
                value
              ]*
              [
                WHERE
                condition
              ]
```

For example, to change the teamId of all people on the team with id T1 to T2, you would enter the SQL command

```
UPDATE      People      SET      teamId= 'T2'      WHERE      teamId= 'T1'
```

## Note

If you omit the WHERE clause, *all rows* of the table will be updated. Be careful always to include a WHERE clause unless you really want to update every row of the table.

The format of the basic DELETE command is

```
DELETE      FROM      table
              [WHERE
                condition
              ]
```

For example, to delete all Lane records for the race with id R12, you would enter the command

```
DELETE      FROM      Lanes      where      raceId= 'R12'
```

## Note



If you omit the `WHERE` clause, *all rows* of the table will be deleted. Be careful always to include a `WHERE` clause unless you really want to delete every row of the table.

There are many more powerful things you can do with SQL, such as using expressions rather than column names in a `SELECT`, or using a `SELECT` statement to get a list of values which are used with the `IN` operator in the `WHERE` clause of a containing statement. For more information on these more advanced kinds of SQL statements, consult an SQL reference.



## Appendix C. Mckoi Database

JRaceman uses the Mckoi Database [<http://www.mckoi.com/database/>] to store all of its data.

Mckoi is a relational database licensed under GPL and written 100% in Java. It is fast, powerful, and easy to use. In fact, it is so easy to use, you probably won't even notice it: JRaceman uses Mckoi internally to manage its database so that you don't need to worry about it.

Mckoi supports both embedded mode and client/server mode, which allows JRaceman to run in single-user mode or in client/server mode, where multiple people can access the same database at once over a network.

You might notice a file called `mckoidb.jar` in the JRaceman distribution. This is the Mckoi database software; this file comes from the Mckoi distribution.

For more information about the Mckoi Database, including source and binary downloads and documentation, see the Mckoi Database home page [<http://www.mckoi.com/database/>].



## Appendix D. Index

### C

Competitions  
definition, 13

### E

Events  
definition, 13

### G

Groups  
definition, 13

### L

Lanes  
definition, 13

### M

Meets  
definition, 13

### P

People  
definition, 13  
primary key, 14  
Progression Plan  
definition, 16  
Progressions  
definition, 16

### R

Races  
definition, 13  
Rounds  
definition, 13

### S

Sections  
definition, 13  
Stages  
definition, 13

### T

Teams  
definition, 13