

## 1 Introduction

Race Coordinator (RC) is a free alternative slot car lap counting application that offers user customizable User Interfaces (UI). It currently supports the Trakmate interface over a serial connection as well as generic Parallel Port, Game Port and Web Camera track communication.

For the latest RC information, visit us at [www.racecoordinator.net](http://www.racecoordinator.net)

## 2 System Requirements

Race Coordinator requires .NET 4, and as such it has the minimum system requirements of .NET 4. The Microsoft website indicates that .NET 4 requires Windows XP or higher and the installer enforces this. The installer should handle checking if .NET 4 is installed already and install it for you if it is not.

For more information on .NET check out:

<http://msdn.microsoft.com/en-us/netframework/aa569263.aspx>  
**.NET 4 has many requirements on the OS. If it does not install, it is most likely because you need to perform one or more Windows updates before you can install it. It is highly recommended that your OS be fully updated before installing Race Coordinator**

## 3 Race Coordinator Goals

### 3.1 Skinnable Race Screens

First and foremost is that no single race screen fits every racers needs. Sadly none ever will. Everybody wants something different and there are too many possibilities to accommodate everybody. With that in mind, Race Coordinator goes for flexibility by allowing for user customizable screens. If no single screen can meet the needs of every person out there, make software that has as many screens as people want.

### 3.2 Usability

Second, it is incredibly important that racers are able to use the software with as little hassle as possible. Although RC provides numerous options to customize the race experience, it also does the best job it can at making these options accessible.

First, it provides a default database that shows off the simplest (and possibly most common) form of racing. A simple Round Robin race is setup and a race event with a Practice time and the Round Robin race is also provided. For the casual racer this may be all the racing formats needed.

RC also provides much more complicated race formats including things like group races, step-up

races and advanced scoring systems. To help ease the setup of these more complicated formats, RC provides two forms of configuration screens. The first are the setup Wizards. The setup Wizards walk you through each and every option RC has, explaining in detail what each option does. Going through each Wizard should give you a good idea of just how much RC supports as well how to setup the race you're looking for. The second form of configuration screens are the management screens. These are a cruder version of the Wizards. They allow for quicker setup if you already have an idea of how things work. They also allow you to modify configurations you've already created. The intent is for the user to use the Wizards as long as it takes to understand RC's vast amount of settings, and then use the Management screens for easier access to these settings, however both the Wizards and management screens can be used to create new configurations at any time.

### 3.3 Statistics

RC is setup to store every lap ever run. It currently supports exporting individual race data to an xls file for viewing giving you an overall view of the race as well as individual heat views. Excel does not need to be installed to export the data, however excel or an Excel Viewer needs to be installed to view the output. A free Excel Viewer that should work with the RC export can be downloaded here:

<http://www.microsoft.com/downloads/en/details.aspx?familyid=1cd6acf9-ce06-4e1c-8dcf-f33f669dbc3a&displaylang=en>

### 3.4 Support

Support is a big deal for any lap counting software. Making sure it works as needed and has the features the community requires. With that said, in an effort to give a little back to the community, I will continue to support RC as long as there is need. There are no guarantees a feature request gets put into RC, but if it makes sense and works within the RC framework it will probably happen.

The team working on RC is small and as such testing every aspect of RC thoroughly is a challenge. We've made every effort to make RC bug free, however if issues do arise, once they are brought to our attention every effort will be made to fix them quickly.

## 4 Using Race Coordinator

The main window has menu items that allow you to manage the data base and controls to configure the particular race you want to run. Before you can race, you must create driver(s), track(s), and race profile(s). Creating cars is optional but will provide more statistical data and potentially a nicer looking UI if cars are added into the mix. Creating events is also optional. Events are a series of races chained together resulting in one final winner at the completion of the entire event. It is a more advanced form of racing that the casual user may not need to use.

## 4.1 Race Day Setup

The main dialog puts all the data entered together to configure the particular race you want to run. Drivers are added/removed as needed. Drivers that are added to the race can be assigned a car. Race selection is made and the heats are assigned.

All data entered in this dialog is saved from run to run so as long as the data entered exists, you can run the same race over and over again just by clicking 'race'.

## 5 Skinning The Race Screen

The main race screen is loaded from a xaml file and is therefore 100% customizable within the limits of what is currently supported. To accomplish this named elements in the xaml file represent specific data values within a race. All a user needs to know to customize the UI is xmal basics (there are plenty of online resources for this) and how to name the elements so that they update real time during a race.

There are three types of bindings that can be accessed during Race Day.

1. Generic data not associated with a specific racer. For example, track information, heat status information, or race status information
2. Data associated with a particular racer. This includes, heat data and overall data and includes things like name, nickname, lap times, etc.
3. Action commands to control the heat. These commands are only available at certain times throughout the heat, and the action is only performed when the element is selected.

Each of the 'Racer Data' names requires one of the three postfixes below:

- '\_Lane<#>': The data requested is for a racer in a specific lane on the track.
- '\_Position<#>': The data requested is for a racer in a specific position in the heat (useful to display the heat standings).
- '\_RaceLeader<#>': The data requested is for a racer in a specific position in the entire race (useful for a leader board).
- '\_GroupLeader<#>': The data requested is for a race in the currently heats group in a specific position in that group (useful for a group leaderboard).
- '\_TeamLeader<#>': The data requested is for a team in a specific position in the entire race (useful for a team race leaderboard).

Both the '\_Lane' and '\_Position' postfixes request data for a racer in the current heat. The racer

is therefore in a specific lane on the track and the background color of the label will automatically be changed to be the lane color they are in. ‘\_RaceLeader’ prefixes do not necessarily have a lane assignment and the background color is left alone.

The following is a list of elements supported and the way to name them to get real time updates.

XAML Element Type	Data Type	Name	Description
Label   TextBlock	Generic Data	TrackName	Displays the track name of the race being run.
Label   TextBlock	Generic Data	RaceTime	Display how much time is left in the heat. It will count down for timed heats and it will count up for lap based races.
Label   TextBlock	Generic Data	RaceLaps	Displays how many laps are left in the heat. It will count down for lap based heats and will count up displaying the heat leaders lap count for time based races
Label   TextBlock	Generic Data	EndHeatValue	Displays the number of laps required to end the heat in a lap based race, or the time in seconds that end the heat in a time based race. This value does not change throughout the race.
Label   TextBlock	Generic Data	RaceName	Displays the name of the race being run
Label   TextBlock	Generic Data	RaceFilter	Displays the filter of the race being run. If set to ‘Unlimited’ any car is available for this race. If set to ‘Practice’ there are no heats and any driver can race any car. If set to a specific car type, only cars of that type are allowed to race.
Label   TextBlock	Generic Data	HeatNumber	Displays the current heat number
Label   TextBlock	Generic Data	NumHeats	Displays the total number of heats in the race
Label   TextBlock	Racer Data	Name	Driver’s real name
Label   TextBlock	Racer Data	Nickname	Driver’s nickname
Label   TextBlock	Racer Data	Lap	Lap count for the driver
Label   TextBlock	Racer Data	LapTime	Driver’s last lap time.
Label   TextBlock	Racer Data	ReactionTime	Driver’s reaction time off the line at the start of the heat
Label	Racer Data	AverageTime	Driver’s average lap time

TextBlock			
Label   TextBlock	Racer Data	BestLapTime	Driver's best lap time
Label   TextBlock	Racer Data	GapLeader	Driver's time gap between him/herself and the race leader. This value will be negative for driver's not in the lead, and positive for the race leader
Label   TextBlock	Racer Data	GapPosition	Driver's time game between him/herself and the driver one position ahead. This value will be negative for everybody but the leader who will have a positive gap time.
Label   TextBlock	Racer Data	TotalLaps	Driver's total number of laps completed across all heats
Label   TextBlock	Racer Data	TotalTime	Driver's total lap time accumulated across all heats
Label   TextBlock	Racer Data	TotalExtendedTime	Total time a driver has raced beyond the actual heat time across all heats. This time is only accumulated in Allow Finish races that end the heat based on time.
Image	Generic Data	TrackImage	Display the image for the track being raced on
Image	Generic Data	RaceStateImage	Displays one of the state images setup by the '<color> Flag Image Path' in the Manage/Track Dialog.
Image	Generic Data	StartLamp<#>	Displays the '<Red Yellow Green> Flag Start Image Path' image setup in the Manage/Track Dialog based on how much time is left in the start countdown
Image	Racer Data	Avatar	Displays the driver's image
Image	Racer Data	CarImage	Display's the driver's car image
Button   MenuItem	Action	Start	When selected, start the race
Button   MenuItem	Action	Pause	When selected, pause the race
Button   MenuItem	Action	AddLaps	When selected, the lap adjustment dialog will come up. Laps are adjustable at the end of every heat and only the racers in the heat can have their laps adjusted
Button   MenuItem	Action	NextHeat	Advance to the next heat
Button   MenuItem	Action	RestartHeat	Stop the current heat and set it up to be started over. All lap data will be lost.
Button   MenuItem	Action	Window	Opens up a new window. The 'CommandParameter' attribute specifies the xaml file to load as the Window content
Button   MenuItem	Action	Dialog	Opens up a new dialog. The 'CommandParameter' attribute specifies the

			xaml file to load as the Window content
Button   MenuItem	Action	ClearLane	Clears the lane specified in the 'CommandParamter' attribute of all its data. A 0 clears all lanes.

To support displaying the same race data or action more than once, the Race Coordinator automatically looks for elements names with an '\_#' appended to them. Starting with 1, it looks for elements of each name until it can't find the element. As an example, if you wanted to display the Track Name twice you would create two labels. The first would be named 'TrackName\_1' and the second would be called 'TrackName\_2'. If you called them 'TrackName\_1' and 'TrackName\_3' only the first one would actually update to the proper track name because 'TrackName\_2' wouldn't be found and the system would never even look for 'TrackName\_3'.

Any field that represents time such as lap time, best lap, etc. support the 'DataContext' attribute. This attribute is a modified JSON object which supports the following fields. For examples see Practice.xaml

index: Where appropriate (for example lap time) specifies which lap to display. 0 will display the last lap, 1 will display the second last lap, 2 the third last lap, etc

decimals: How many decimal points to display for precision purposes.

Here are some 'Racer Data' example names:

- 'Avatar\_Position2\_1': Displays the image of the driver in second place for the heat
- 'Avatar\_Lane2\_1': Displays the image of the driver in lane two for the heat
- 'Avatar\_RaceLeader2\_1': Displays the image of the driver in second overall place

There are several xaml files provided with the RC install. These can be used as a How To guide in accessing many of the UI features RC supports. They are located in the <install\_dir>\data\xaml directory.

## 6 Webservice

### 6.1 File Serving

RC will serve any local file you create. Any request that does not start with "api" will be treated as a file request. RC will map the path to a directory on the local file system and serve up the file specified. For example <http://localhost:1234/demo.html> will cause RC to serve up the file "demo.html" as long as it exists in the root directory the webservice is configured to use. Sending <http://localhost:1234> will cause RC to serve up the default html file (index.html by default).

## 6.2 Query Protocol Version 0.0.1

RC will treat any local path that starts with “/api” as a query for race data. All queries consist of the initial /api an optional version number such as “/v1”, “/v1.1”, etc, and the query itself. If the version is left off the latest and greatest version of the query is run. If the version is specified, a response code of 302 (Permanently Moved) will be returned if the older version is no longer supported, otherwise the older version of the query will be run and returned. As newer versions of the protocols come out the old versions should be considered deprecated. The old versions will be maintained for an unspecified amount of time and as such client code should be changed over to use the newer versions as soon as possible.

Every query can pass in a transaction id (tid) that will be returned in the response. This tid can be use by clients to pair their queries with responses. This is very important when a client makes more than one query. Although the response order will typically be in the same order as the request, any number of things can cause the response to come out of order. The client needs to be coded to handle this.

The query(s) to run are specified as the “q” string. The value is a bit field, each bit representing which query to run. Parameters to the query(s) are then specified as additional query strings. Each parameters value is a bit field for the query the parameter is used for followed by the value, separated by a colon. In some cases queries can be run more than once with different parameters. In this case two parameters would be passed both with the same bit field, but with different parameters.

Finally, the response is returned as a json object with two fields. The first is the “tid” which was passed in the query. This value is returned exactly as it was received. The second object is the “r” field which is a json array of response objects. Each object in the array is itself a json object with two fields. The first field is “q” with a value equal to the bit the response is for. The second field “r” contains the response for the query. This response is query specific but will usually be a json object with the requested fields returned.

Here’s an example query and response. The rest of this section gets into the details of the queries and their responses.

Query:

```
http://localhost:8080/api?tid=m0&q=\[{"trackData":0}, {"raceData":0}, {"raceDriverData":0}, {"heatData":0}, {"heatDriverData":{"f":\["\*"\]}}, {"heatDriverLapData":{"did":3}}, {"heatDriverLapData":{"did":4}}\]
```

Response:

```
{"tid":"m0","r":[{"q":{"trackData":0},"r":{"n":"Bright Plume Raceway","l":[{"c":"Red","l":70}, {"c":"White","l":70}, {"c":"Blue","l":70}, {"c":"Yellow","l":70}]}]}, {"q":{"raceData":0},"r":{"n":"Fuel Round Robin","d":[{"n":"Gene","nn":"Swamper Gene","s":1,"did":6}, {"n":"Andrea","nn":"The Pants","s":2,"did":2}, {"n":"Christine","nn":"Peo Fuente","s":3,"did":3}, {"n":"Dave","nn":"Olden"}]}]}]}
```

```

McGroin", "s":4, "did":1}, {"n":"Meyer", "nn":"Bull
Dog", "s":5, "did":4}, {"n":"Noah Jack", "nn":"Boy
Wonder", "s":6, "did":5}}]}, {"raceDriverData":{"d":[{"n":"Dave", "nn":"Old
en McGroin", "s":4, "did":1, "v":9}, {"n":"Christine", "nn":"Peo
Fuente", "s":3, "did":3, "v":8}, {"n":"Gene", "nn":"Swamper
Gene", "s":1, "did":6, "v":8}, {"n":"Andrea", "nn":"The
Pants", "s":2, "did":2, "v":7}, {"n":"Meyer", "nn":"Bull
Dog", "s":5, "did":4, "v":0}, {"n":"Noah Jack", "nn":"Boy
Wonder", "s":6, "did":5, "v":0}}]}, {"heatData":{"s":4, "sname":"Racing", "hn
":0, "t":93.903}}, {"heatDriverData":{"drivers":[{"n":"Gene", "nn":"Swampe
r
Gene", "rt":0.05, "led":0, "l":8, "lt":10.808, "blt":8.649, "a":10.278, "g":9.
279, "gp":2.055, "p":2, "f":200, "fm":200, "did":6}, {"n":"Andrea", "nn":"The
Pants", "rt":0.361, "led":0, "l":7, "lt":10.949, "blt":9.524, "a":10.836, "g":
13.742, "gp":4.435, "p":3, "f":200, "fm":200, "did":2}, {"n":"Christine", "nn"
:"Peo
Fuente", "rt":0.39, "led":0, "l":8, "lt":9.631, "blt":7.283, "a":10.021, "g":7
.224, "gp":7.224, "p":1, "f":200, "fm":200, "did":3}, {"n":"Dave", "nn":"Olden
McGroin", "rt":0.397, "led":9, "l":9, "lt":9.964, "blt":6.692, "a":9.212, "g":
0, "gp":0, "p":0, "f":200, "fm":200, "did":1}}]}, {"heatDriverLapData":{"did"
:3, "l":[{"abs":11.697, "lt":11.697}, {"abs":23.07, "lt":11.372}, {"abs":33.
846, "lt":10.776}, {"abs":42.452, "lt":8.606}, {"abs":49.736, "lt":7.283}]}
, {"heatDriverLapData":{}}}

```

## 6.2.1 Track Data

Returns information relative to the track the race is being run on.

### 6.2.1.1 Bit Value

0x0001

### 6.2.1.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.

### 6.2.1.3 Response

Key	Type	Value
<b>n</b>	string	Name of the track
<b>l</b>	array	Array of Lane Object

#### 6.2.1.3.1 Lane Object

Key	Type	Value
<b>c</b>	string	Color of the lane
<b>l</b>	float	Length of the lane (in feet)



#### 6.2.1.4 Example

<http://localhost:8080/api?tid=m0&q=1>

returns

```
{ "tid": "m0", "r": { "q": 1, "n": "Bright Plume Raceway", "l": { "c": "Red", "l": 70, "c": "White", "l": 70, "c": "Blue", "l": 70, "c": "Yellow", "l": 70 } } }
```

#### 6.2.2 Race Data

Returns information specific to the race queried. This includes race name and configuration parameters, as well as which drivers are participating/participated in the race.

##### 6.2.2.1 Bit Value

0x0002

##### 6.2.2.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.

##### 6.2.2.3 Response

Key	Type	Value
<b>n</b>	string	Name of the race
<b>rlt</b>	object	Record Object identifying the driver with the all-time record lap time for this race configuration. If there is no record, an empty object is returned
<b>rs</b>	object	Record Object identifying the driver with the all-time record score for this race configuration. If there is no record, an empty object is returned
<b>blt</b>	object	Record Object identifying the driver with the best lap time for the specific race specified by the rid parameter. If there is no record, an empty object is returned
<b>d</b>	array	Array of Race Driver Object. This list can change throughout the race if late joiners are added on the fly. However, this is probably a fairly rare event.

###### 6.2.2.3.1 Race Driver Object

Key	Type	Value
<b>n</b>	string	Name of the driver
<b>nn</b>	string	Nick name of the driver
<b>s</b>	int	Seed of the driver for this race
<b>did</b>	int	Driver ID used to uniquely reference this driver in other queries
<b>h</b>	array	Array of integers. Each integer representing which heat the driver is in. The heat numbers are 0 based so a 0 is the first heat in the race. If this is

		for a live race, the heat may not have been run yet or may be in progress. In a live race, any heat not already run could change at any time. Heats can be added/removed and drivers can be changed. Once a heat is run however, it cannot be changed.
--	--	--

#### 6.2.2.3.2 Record Object

Key	Type	Value
<b>n</b>	String	Name of the driver holding this record
<b>nn</b>	string	Nick name of the driver holding this record
<b>v</b>	Double	Value of the record. This value could be a lap count, points scored, or some form or time all dependent on the race configuration
<b>d</b>	string	Date the record was obtained. If this is an empty string it indicates a new record during the current live race

#### 6.2.2.4 Example

<http://localhost:8080/api?tid=m0&q=2>

returns

```
{
  "tid": "m0",
  "r": {
    "q": 2,
    "n": "Round Robin",
    "rlt": {
      "n": "Gene",
      "nn": "Swamper Gene",
      "v": 7.36212143252148,
      "d": ""
    },
    "rs": {
      "n": "Gene",
      "nn": "Swamper Gene",
      "v": 1,
      "d": ""
    },
    "blt": {
      "n": "Gene",
      "nn": "Swamper Gene",
      "v": 7.36212143252148,
      "d": ""
    },
    "d": {
      "n": "Andrea",
      "nn": "The Pants",
      "s": 1,
      "did": 2,
      "h": [0, 1, 2, 3]
    },
    "n": "Christine",
    "nn": "Peo Fuente",
    "s": 2,
    "did": 3,
    "h": [0, 1, 2, 5]
  },
  "n": "Dave",
  "nn": "Olden McGroin",
  "s": 3,
  "did": 1,
  "h": [0, 1, 4, 5]
},
  "n": "Gene",
  "nn": "Swamper Gene",
  "s": 4,
  "did": 6,
  "h": [0, 3, 4, 5]
},
  "n": "Meyer",
  "nn": "Bull Dog",
  "s": 5,
  "did": 4,
  "h": [2, 3, 4, 5]
},
  "n": "Noah Jack",
  "nn": "Boy Wonder",
  "s": 6,
  "did": 5,
  "h": [1, 2, 3, 4]
}
}
```

### 6.2.3 Heat Data

Returns data related to the heat including its state, time and other useful information

#### 6.2.3.1 Bit Value

0x0004

#### 6.2.3.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.

### 6.2.3.3 Query Response

Key	Type	Value
<b>s</b>	int	State represented as an integer: 0. Heat has not started yet 1. Heat was started, but aborted, and has not been started since the abort. 2. Heat start count down is in progress 3. Heat re-start count down is in progress 4. Heat is currently in progress. Laps will be counted 5. Heat is paused (yellow flag). Laps will not be counted 6. The heat has ended 7. The heat has ended and this is the end of the race
<b>sname</b>	string	English translation of the heat state
<b>cnt</b>	int	Number of heats in the race
<b>hn</b>	int	Heat number. This value is 0 based so 0 is the first heat in the race. The range of values will be [0, cnt)
<b>t</b>	float	This value depends on the state of the race. 0 (Not started) Either the seconds left in the auto start countdown or a static value representing the first time seen if the race was started. 1 (Not Restarted after an abort) Static value representing the first time seen once the race is started again 2, 3 (Start and Restart Countdown) How many seconds left in the countdown 4, 5 (Racing/Paused) Time in seconds representing either the time left in the heat or the amount of time the heat has been in progress for. This value depends on if the race is lap or time based. 6 (Heat ended) Either the second left in the auto advance countdown or a static value as the last value seen before the heat ended. 7 (Race Ended) Last value seen when the last heat of the race was ended.

### 6.2.3.4 Example

<http://localhost:8080/api?tid=m0&q=4>

returns

```
{"tid":"m0", "r":[{"q":4,"s":5,"sname":"Paused","hn":0,"cnt":6,"t":13.071}]}
```

### 6.2.4 Race Driver Data

Retrieve data for each driver in the race. These value are an accumulation of every heat the driver has been in at the time of the query.

### 6.2.4.1 Query Value

0x0008

### 6.2.4.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.
<b>f</b>	yes	["*"]	JSON array of strings specifying what fields you want back in the response. Passing in "*" as any of the elements in the array will cause every field to be returned. All other accepted values are the names of the response fields. See Race Driver Object for a list of field names that can be specified

### 6.2.4.3 Query Response

Key	Type	Value
<b>d</b>	array	Array of Race Driver Object. This array is in the order of the current overall race standings with the first place driver appearing first in the list. Each field returned represents data for the drivers entire race (all heats they've raced in combined)

#### 6.2.4.3.1 Race Driver Object

Key	type	Value
<b>n</b>	string	Name of the driver
<b>nn</b>	string	Nick name of the driver
<b>did</b>	int	Unique id for the driver. Can be used as a parameter in other queries.
<b>v</b>	double	Drivers overall race score as defined by the race configuration. This could be their total lap count, total time, fastest lap, or points earned thus far in the race.
<b>led</b>	int	How many laps the driver has led in this race.
<b>l</b>	double	Lap count
<b>blt</b>	double	Best lap time
<b>a</b>	double	Average lap time
<b>g</b>	double	Gap between driver and leader
<b>gp</b>	double	Gap between driver and driver one position ahead
<b>*</b>	N/A	Return all fields listed in this table. If * is specified, there is no need for any other fields to be specified. Use this with caution as in large races it could return a lot of data.

#### 6.2.4.4 Example

<http://localhost:8080/api?tid=m0&q=8>

returns

```
{
  "tid": "m0",
  "r": {
    "q": 8,
    "d": {
      "n": "Gene",
      "nn": "Swamper Gene",
      "did": 6,
      "v": 1,
      "led": 1,
      "l": 1,
      "blt": 7.362,
      "a": 7.362,
      "g": -2.047,
      "gp": -2.047
    },
    "n": "Christine",
    "nn": "Peo Fuente",
    "did": 3,
    "v": 1,
    "led": 0,
    "l": 1,
    "blt": 9.409,
    "a": 9.409,
    "g": 2.047,
    "gp": 2.047
  },
  "n": "Andrea",
  "nn": "The Pants",
  "did": 2,
  "v": 1,
  "led": 0,
  "l": 1,
  "blt": 10.531,
  "a": 10.531,
  "g": 3.169,
  "gp": 1.122
},
  "n": "Dave",
  "nn": "Olden McGroin",
  "did": 1,
  "v": 1,
  "led": 0,
  "l": 1,
  "blt": 11.757,
  "a": 11.757,
  "g": 4.395,
  "gp": 1.226
},
  "n": "Meyer",
  "nn": "Bull Dog",
  "did": 4,
  "v": 0,
  "led": 0,
  "l": 0,
  "blt": -1,
  "a": 0,
  "g": 7.362,
  "gp": 11.757
},
  "n": "Noah Jack",
  "nn": "Boy Wonder",
  "did": 5,
  "v": 0,
  "led": 0,
  "l": 0,
  "blt": -1,
  "a": 0,
  "g": 7.362,
  "gp": 0
}
}
}
```

### 6.2.5 Heat Driver Data

Retrieve data for each driver in the heat

#### 6.2.5.1 Bit Value

0x0010

#### 6.2.5.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.
<b>hn</b>	Yes	-1	Heat number the data is being requested for. This value is 0 based so the first heat in a race would be specified as 0. If -1 is specified, the last heat in the race run will be used so -1 can be used to retrieve the currently running heats data.
<b>f</b>	yes	["*"]	JSON array of strings specifying what fields you want back in the response. Passing in "*" as any of the elements in the array will cause every field to be returned. All other accepted values are the names of the response fields. See Heat Driver Object for a list of field names that can be specified

#### 6.2.5.3 Query Response

The tid field is always returned, otherwise only the fields

Key	Type	Value
<b>d</b>	array	Array of Heat Driver Object. One entry for each driver racing in the heat. If a lane is empty the Heat Driver Object will be an empty object.

### 6.2.5.3.1 Heat Driver Object

Key	Type	Value
<b>n</b>	string	Name of the driver
<b>nn</b>	string	Nick name of the driver
<b>rt</b>	double	Reaction time for the driver. Only valid on certain race formats in which the driver starts behind the S/F line
<b>led</b>	int	How many laps the driver has led in this heat.
<b>l</b>	double	Lap count
<b>lt</b>	double	Last lap time
<b>blt</b>	double	Best lap time
<b>a</b>	double	Average lap time
<b>g</b>	double	Gap between driver and leader
<b>gp</b>	double	Gap between driver and driver one position ahead
<b>p</b>	int	Position of the driver. This value is 0 based so 0 indicates the leader
<b>did</b>	int	Unique id for the driver. Can be used as a parameter in other queries.
<b>f</b>	int	How many units of fuel the driver has left. This value is only meaningful in fuel races and can actually be larger than the "fm" value depending on race configuration.
<b>fm</b>	int	Max fuel a car can obtain when refueling.
<b>bf</b>	int	Number of laps the driver has left to serve a Black Flag Penalty before being disqualified. 0 if driver has no black flag penalty.
<b>bft</b>	int	Amount of time driver has left to serve a time stopped penalty. Only valid if bf is greater than 0 and the race configuration is setup for timed stops.
<b>*</b>	N/A	Return all fields listed in this table. If * is specified, there is no need for any other fields to be specified. Use this with caution as in large races it could return a lot of data.

### 6.2.5.4 Example

[http://localhost:8080/api?tid=m0&q=16&f\[\]=16:n:lt:blt:l](http://localhost:8080/api?tid=m0&q=16&f[]=16:n:lt:blt:l)

returns

```
{"tid":"m0",  
"r":[{"q":16,"d":{"n":"Andrea","l":1,"lt":10.531,"blt":10.531},"n":"Christine","l":1,"lt":9.409,"blt":9.409},"n":"Dave","l":1,"lt":11.757,"blt":11.757},"n":"Gene","l":1,"lt":7.362,"blt":7.362}]}
```

### 6.2.6 Heat Driver Lap Data

Returns lap timing data for a specific driver. More than one driver may be queried at a time through the required "did" parameter

#### 6.2.6.1 Bit Value

0x0020

### 6.2.6.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.
<b>hn</b>	Yes	-1	Heat number the data is being requested for. This value is 0 based so the first heat in a race would be specified as 0. If -1 is specified, the last heat in the race run will be used so -1 can be used to retrieve the currently running heats data.
<b>did</b>	No	N/A	Colon separated list of Driver IDs to return data for. This value should have been returned by one of the other queries that return did's such as Race Data or Heat Driver Data
<b>ln</b>	Yes	0	Lap number that we start returning laps at. 0 indicates the first lap, 1 the second, 2 the third, etc.
<b>l</b>	Yes	5	Limit the number of lap times returned to this value. If 0 or less than 0, all lap times will be returned. ***WARNING, requesting all lap times could put a large burden on the server depending on how many laps there are to return. It's best to request the lap times in "chunks" using this parameter and the "ln" parameter. The server may also impose its own limit in which case no more than that limit will ever be returned regardless of this parameter.

### 6.2.6.3 Query Response

Key	Type	Value
<b>did</b>	int	Driver id the laps pertain to. This is simply passed back so the client can easily tell which driver this response is for which is especially useful if the client makes this request for more than one driver at a time using different "did" values.
<b>l</b>	array	Array of Lap Times Object

#### 6.2.6.3.1 Lap Times Object

Key	Type	Value
<b>abs</b>	double	Absolute time in seconds this lap occurred
<b>lt</b>	double	Time in seconds this lap took to complete
<b>p</b>	int	Position in the heat the driver was in when this lap was completed. This value is 0 based so the first place driver will be 0

### 6.2.6.4 Example

[http://localhost:8080/api?tid=m0&q=32&did\[\]=32:1:2:3:6](http://localhost:8080/api?tid=m0&q=32&did[]=32:1:2:3:6)

returns

```
{
  "tid": "m0",
  "r": [
    {
      "q": 32, "did": 1, "l": [
        {
          "abs": 11.757, "lt": 11.757, "p": 3
        }
      ]
    },
    {
      "q": 32, "did": 2, "l": [
        {
          "abs": 10.531, "lt": 10.531, "p": 2
        }
      ]
    },
    {
      "q": 32, "did": 3, "l": [
        {
          "abs": 9.409, "lt": 9.409, "p": 1
        }
      ]
    },
    {
      "q": 32, "did": 6, "l": [
        {
          "abs": 7.362, "lt": 7.362, "p": 0
        }
      ]
    }
  ]
}
```

## 6.2.7 Races

Returns a list of races available to run other queries on. This list includes any race previously run to completion. If a race is currently being run, it will only appear in this list if it has been completed.

### 6.2.7.1 Bit Value

0x0040

### 6.2.7.2 Query Parameters

Query String	Optional	Default	Description

### 6.2.7.3 Query Response

Key	Type	Value
<b>r</b>	array	Array of Race Object

#### 6.2.7.3.1 Race Object

Key	Type	Value
<b>rid</b>	integer	Race ID other queries can use to reference this race
<b>n</b>	string	Name of the race data is available for
<b>en</b>		Event name this race was run with. Empty string if it was not part of an event.
<b>d</b>	string	Date the race was completed
<b>ed</b>	string	Date the event this race is a part of was started. Empty string if it was not part of an event.
<b>w</b>	String	Name of the winning driver
<b>s</b>	double	Score the winning driver had

### 6.2.7.4 Example

\*\*\*Note, currently not implemented

Returns

## 6.2.8 Race Config

Returns a summarized view of the race configuration. This will not contain all the configuration options, just enough for the client to make smart decisions on what/how to display things. This is one of the few queries that cannot change it's return value once the race is in progress so a client only needs to make this query once.

### 6.2.8.1 Bit Value

0x0080



### 6.2.8.2 Query Parameters

Query String	Optional	Default	Description
<b>rid</b>	Yes	-1	ID specifying which race the track data is being requested for. -1 indicates the currently running race. Any other value must come from first running the Races query which returns a list of available races.

### 6.2.8.3 Query Response

Key	Type	Value
<b>f</b>	bool	True if this is a fuel race, false otherwise
<b>d</b>	bool	True if this is a digital race, false if this is an analog race
<b>srms</b>	bool	True if this is a digital scorpius wireless race, false if its either analog or a non-scorpius race
<b>hs</b>	integer	Heat scoring method 1. Lap based scoring 2. Fastest lap time based scoring 3. Total lap time based scoring
<b>he</b>	Integer	Heat end method 1. Infinite heat 2. Time based heat 3. Lap based heat
<b>hev</b>	Integer	Value heat ends at. Either 0 for infinite heats, time in seconds for time based heats, or lap count for lap based heats.
<b>su</b>	bool	True if this is a step up race, false otherwise
<b>dn</b>	integer	How many of the lowest heats to drop for a driver (drop N), 0 to count all heats
<b>bf</b>	Integer	Black flag penalty type 1. Drive Through 2. Stop and Go 3. Timed Stop 4. None
<b>bft</b>	Integer	Black flag time in seconds a driver must stop in the pits to serve a Timed Stop penalty

### 6.2.8.4 Example

<http://localhost:8080/api?tid=m0&q=128>

Returns

```
{"tid":"m0", "r":{"q":128,"f":false,"d":false,"hs":1,"su":false,"dn":0}}
```

### 6.2.9 Race Director

Performs the requested race director action(s) if possible.

### 6.2.9.1 Bit Value

0x0100

### 6.2.9.2 Query Parameters

Query String	Optional	Default	Description
<b>c</b>	No	N/A	Command bit values the client wants to run. To run more than one command at a time simply add these values together just like the bit values for queries. See the Command Parameter Table for details on each command that can be run. ***Note that commands will be run in the order listed in the table. Therefore if the client wish to restart a running heat it could pass a 0x000A (10) as the command. This will pause the running heat and then restart it. In this same situation the client could pass a 0x004A command which would pause the running heat, restart it, and then send a call button which will start the restarted heat back up.
<b>p</b>	No	N/A	Password associated with the command. If the RC Server is configured with a password, this value must match that password exactly for the command to be accepted from the client. If the Server does not have passwords enabled, any password will be accepted.

#### 6.2.9.2.1 Command Parameter Table

Command	Bit Value	Description
<b>Pause Heat</b>	0x0001	Pause the current heat. Can fail if the heat is not currently started.
<b>Start/Resume Heat</b>	0x0002	Start or resume the current heat. Can fail if the heat is already starting, is already started, or is over.
<b>Next Heat</b>	0x0004	Advance to the next heat. Can fail if the current heat is not over.
<b>Restart Heat</b>	0x0008	Restart the current heat. Can fail if the heat has not been started and the heat is not currently paused. Will also fail if the heat has ended.
<b>Skip Heat</b>	0x0010	Skip the remainder of the current heat and advance to the next heat. Can fail if the current heat is not paused or has already ended.
<b>Skip Race</b>	0x0020	Skip the entire race. Can fail if the current heat is not paused or the current heat or race is already over.
<b>Call Button</b>	0x0040	Simulate a track call button press. This has many different behaviors based on the current race state. While a heat is in

		progress it can be used to pause or start/resume the heat. When in between heats it can be used to advance to the next heat and start the new heat. It can also be used to abort the race start countdown. This call will not fail.
--	--	---

### 6.2.9.3 Query Response

Key	Type	Value
<b>r</b>	integer	Integer value representing all the bit value commands that succeeded. If this value is -1 it indicates that the password did not match the RC server password so the commands were ignored.

### 6.2.9.4 Example Query

<<[http://localhost:8080/api?tid=m0&q=256&c\[\]=256:64&p\[\]=256:Race Coordinator](http://localhost:8080/api?tid=m0&q=256&c[]=256:64&p[]=256:Race Coordinator)>>

Returns

```
{"tid":"m0", "r":{"q":256,"r":64}}
```

### 6.2.10 Multi Query Example

With this protocol you can run more than one query at a time. Simply add up all the bit values for the queries you want run and make sure to specify parameters as appropriate. The following query will request all the currently support queries, including the Heat Driver Lap Data for 4 drivers. To request every possible query simply add up the bit values documented here. If a query bit value is request that is not currently supported, the entire query will fail. The following query will work and run several queries at once including the HeatDriverLapData query 4 times to get 4 different drivers lap information.

[http://localhost:8080/api?tid=m0&q=63&did\[\]=32:1:2:3:6](http://localhost:8080/api?tid=m0&q=63&did[]=32:1:2:3:6)

This queries response will look like this:

```
{"tid":"m0", "r":{"q":1, "n":"Bright Plume Raceway", "l":{"c":"Red", "l":70}, {"c":"White", "l":70}, {"c":"Blue", "l":70}, {"c":"Yellow", "l":70}}, {"q":2, "n":"Round Robin", "rlt":{"n":"Gene", "nn":"Swamper Gene", "v":7.36212143252148, "d":""}, "rs":{"n":"Gene", "nn":"Swamper Gene", "v":1, "d":""}, "blt":{"n":"Gene", "nn":"Swamper Gene", "v":7.36212143252148, "d":""}, "d":{"n":"Andrea", "nn":"The Pants", "s":1, "did":2, "h":[0,1,2,3]}, {"n":"Christine", "nn":"Peo Fuente", "s":2, "did":3, "h":[0,1,2,5]}, {"n":"Dave", "nn":"Olden McGroin", "s":3, "did":1, "h":[0,1,4,5]}, {"n":"Gene", "nn":"Swamper Gene", "s":4, "did":6, "h":[0,3,4,5]}, {"n":"Meyer", "nn":"Bull Dog", "s":5, "did":4, "h":[2,3,4,5]}, {"n":"Noah Jack", "nn":"Boy Wonder", "s":6, "did":5, "h":[1,2,3,4]}}, {"q":4, "s":5, "sname":"Paused", "hn":0, "cnt":6, "t":13.071}, {"q":8, "d":{"n":"Gene", "nn":"Swamper Gene", "did":6, "v":1, "led":1, "l":1, "blt":7.36212143252148, "a":7.36212143252148, "g":-2.047, "gp":-2.047}, {"n":"Christine", "nn":"Peo Fuente", "did":3, "v":1, "led":0, "l":1, "blt":9.40923039119689, "a":9.40923039119689, "g":2.047, "gp":2.047}, {"n":"Andrea", "nn":"The Pants", "did":2, "v":1, "led":0, "l":1, "blt":10.5312418540841, "a":10.5312418540841, "g":3.169, "gp":1.122}, {"n":"Dave", "nn":"Olden McGroin", "did":1, "v":1, "led":0, "l":1, "blt":11.7572971910786, "a":11.7572971910786, "g":4.395,
```

```
gp":1.226},{ "n":"Meyer", "nn":"Bull Dog", "did":4, "v":0, "led":0, "l":0, "blt":-
1, "a":0, "g":7.362, "gp":11.757}, {"n":"Noah Jack", "nn":"Boy
Wonder", "did":5, "v":0, "led":0, "l":0, "blt":-
1, "a":0, "g":7.362, "gp":0}}, {"q":16, "d":{"n":"Andrea", "nn":"The
Pants", "rt":0.026, "led":0, "l":1, "lt":10.531, "blt":10.531, "a":10.531, "g":3.169, "gp":1.122, "p":2, "f":
0, "fm":200, "did":2}, {"n":"Christine", "nn":"Peo
Fuente", "rt":0.307, "led":0, "l":1, "lt":9.409, "blt":9.409, "a":9.409, "g":2.047, "gp":2.047, "p":1, "f":0,
"fm":200, "did":3}, {"n":"Dave", "nn":"Olden
McGroin", "rt":0.255, "led":0, "l":1, "lt":11.757, "blt":11.757, "a":11.757, "g":4.395, "gp":1.226, "p":3,
"f":0, "fm":200, "did":1}, {"n":"Gene", "nn":"Swamper
Gene", "rt":0.165, "led":1, "l":1, "lt":7.362, "blt":7.362, "a":7.362, "g":0, "gp":0, "p":0, "f":0, "fm":200,
"did":6}}, {"q":32, "did":1, "l":{"abs":11.757, "lt":11.757, "p":3}}, {"q":32, "did":2, "l":{"abs":10.531,
"lt":10.531, "p":2}}, {"q":32, "did":3, "l":{"abs":9.409, "lt":9.409, "p":1}}, {"q":32, "did":6, "l":{"abs":
7.362, "lt":7.362, "p":0}}}]}
```

### 6.2.11 Quick Reference

The following is a list of queries and their bit values for quick reference:

Query Name	Bit Value ("q")
Track Data	0x0001
Race Data	0x0002
Heat Data	0x0004
Race Driver Data	0x0008
Heat Driver Data	0x0010
Heat Driver Lap Data	0x0020
Races	0x0040
Race Config	0x0080

## 6.3 Displaying Race Information On a Mobile Device

Race Coordinator (RC) has the ability to provide race data to any connected mobile device that has a web browser via a Local Area Network (LAN). There are a few steps that must be followed to enable Remote Race Information. First a WiFi network must be setup and configured, RC must then be configured to provide the Race Data, and finally the Windows Firewall (or any firewall you're using) must be configured to allow remote devices to make requests to RC.

### 6.3.1 Setup and configure a WiFi network.

Due to the number and diversity of hardware it is beyond the scope of this document to be able to offer much help in setting up a WiFi network; many homes will already have a network that can be used. Clubs and halls may not have a suitable network so there are two possibilities, the first and probably simplest is to add a Wireless Router to the PC running RC, and use this to provide a Wireless Network. This can be an old Wireless broadband router that is no longer used, since for most cases it can be used as a Wireless Network without a broadband connection to the internet. Another option would be to configure a WiFi enabled laptop to act as a virtual WiFi router, this may be easier on a Windows 7 PC.

### 6.3.2 Configure Race Coordinator

Race Coordinator must be configured to allow live race updates from remote clients. By default this access is disabled. To enable it, simply go to the “Options/Web Server” menu option on the Race Day Setup screen. Here’s a breakdown of the options and their usage:

- **Enabled** : If checked, remote clients can connect to the live race and obtain information about the race
- **Listener Port**: The port clients should connect on. This port must not be in use by any other services on your PC. The default of 8080 should work on most setups, but if you have connection issues try different ports.
- **Server Base Directory**: This is the root directory where your client HTML code will be served from. You do not have to provide client code, however a default set of HTML files is available [here](#). It’s a good idea to provide default client code so the clients don’t have to trouble themselves with it. This directory should point to the location where the index.html file ends up.
- **Default Page**: This is the default page for a client to use if they request your HTML client. Typically this will be index.html which will act as a landing page for users.
- **Server Cache Size**: Size in Megabytes of the server cache. The larger this is, the potentially more memory RC will use, however client request will be processed faster because RC won’t have to load as much from disk. This size will depend on the size of your HTML client files. Setting this to 0 will disable caching, but is only recommended if you’re developing client code.
- **Race Director Password**: Some queries to the server require a password from the client. It is recommended that you change this password and only tell users you which to act as race directors what it is. If this password is obtained by public users, they can start/stop your races remotely from their mobile devices.

### 6.3.3 Configure Windows Firewall.

Windows Firewall must either be disabled (not recommended) or configured to allow the port used by RC to be accessed from remote devices, if not then any remote device may hang making a request to the RC web server. Depending on what operating system RC runs on, check out the following links for details on how to open the port needed by RC. When asked, you want to specify the same port number set by the “Listener Port” value and you want to open the TCP port. You do not need to open the UDP port.

- Windows XP: <http://support.microsoft.com/kb/308127>
- Windows Vista: <http://windows.microsoft.com/en-us/windows-vista/open-a-port-in-windows-firewall>
- Windows 7: <http://windows.microsoft.com/en-us/windows7/open-a-port-in-windows-firewall>

If you use other anti-virus software they may also provide a firewall. You will have to similarly either disable that firewall (not recommended) or open the port in it as well.

### 6.3.4 Connecting Mobile Devices.

Once configured then any mobile device's web browser may access the Race Information whenever RC is running, to do this you must direct your web browser to the one of the following URL's (depending on if name resolution is supported).

<http://localhost:8080>

<http://xxx.xxx.xxx.xxx:8080>

<http://pcname:8080>

The xxx.xxx.xxx.xxx number is the IP address of the PC running Race Coordinator. To determine your ip address, use the Windows Start Menu and run the program "cmd". In the command prompt window type "ipconfig". A typical IP Address would be 192.168.xx.yy

This can be found by running ipconfig from the command line on the PC, so a typical IP address would be 192.168.1.20. Find your IPv4 Address and give it a go.

The easiest case to get working is the first one <http://localhost:8080>, however that particular address will only work on the computer running RC. It is strongly recommend you try this first, because getting that working will test your RC configuration and HTML client install. Once that's working, try things out on a remote machine which will involve your firewall being configured correctly and you using the correct address in your browser.

## 7 Contact

For issues, questions, comments, or suggestions, feel free to PM me on [www.slotcarillustrated.com](http://www.slotcarillustrated.com) (user name daufferh) or email me us at [contact@racecoordinator.net](mailto:contact@racecoordinator.net)

For bugs, please describe the bug as clearly as possible. Screen shots are very helpful when possible, and your system setup may be needed to trouble shoot issues. RC has been tested on both Windows7 and Windows XP, but there can never be too much testing.

## 8 Changelist

- 1.11.4.0
  - Removed software sensor debounce for arduino when the hardware debounce is being used
- 1.11.3.0
  - Fixed crash in arduino interface when you configure more lap pins then lanes on the track and then un-pause a heat
  - Fixed arduino refueling bug when coming out of a yellow flag
- 1.11.2.0
  - Updated xls export library to support faster exports, and better cell formatting
  - Added graphing to the xls export now that its supported by the library being used
  - Added CarName, CarNumber, and CarType to the available xls cell values

- Added save/load functionality for Practice races. This must be manually done, but will work
- Added export functionality to Practice races.
- Changed when arduino sends a segment time that is triggered off the lap sensor so that the segment is counted as part of the lap just being triggered.
- Enhanced segment timing display and exporting. \*\*\*Changes are not backwards compatible so races with saved segment data will not export correctly and existing custom xaml files that display segment timing need to be updated.
- Fixed text in guided race setup for practice lap sounds
- Fixed step-up scoring. \*\*\*NOTE: Rankings were correct but the reported scoring both in race and exported were not correct
- 1.11.1.0
  - Fixed exporting stats from the options/stats menu on the Race Day Setup screen
- 1.11.0.0
  - Added tiebreaker configuration
  - Added text to speech support
  - Added warmup time during auto advance and auto start periods
  - Added support for step up races defined by custom rotation files
  - Added power status LED support for arduino track interface. No more power regulators running off your track power.
  - Added segment time support to arduino interface (currently not able to export the segment times)
  - Added fuel begin/fuel end support to arduino interface
  - Fixed intermittent issue with the arduino having the wrong power state for a master relay at the start of the first heat in a race
  - Fixed issues with incorrect lap times in start at current races after the drivers second heat
  - Fixed issue with start at current races not counting lap times correctly if the last lap in the previous heat was faster than the min lap time.
  - Fixed issues with TM counting double lap times for start at current races
  - Fixed issues surrounding start at current 1<sup>st</sup> lap times being inaccurately reported
  - Fixed 'Adjust Drift Laps' which were totally broken
  - Hack fixed counting large lap times (greater than 6000 seconds reported by DSxx boxes)
  - Fixed issue with heat leader LEDs staying on between heats. They now start turned off until an 'official' heat leader is determined by drivers crossing the S/F line for the first time.
  - Fixed Add Lap Sections dialog so that it takes into account the lane font color when displaying information
  - Fixed Ondeck and NextHeat windows to display the lane colors

- Changed callbutton and space bar (the “toggle” xaml keyword) from trying to close the raceday window at the end of a race.
- 1.10.0.0
  - Fixed issue in which the wrong lap time is marked as a drift lap if the lap occurred during the actual drifting
  - Changed heat tiebreakers such that if “Start at Current” is not configured, there is no heat tie breaker, and thus the driver crossing the S/F line first will always be given the heat lead as expected.
  - Fixed rare crash related to adding font color configuration
  - Fixed rare crash on shutdown when using more than one arduino for the track interface
  - Fixed lap based races so that they correctly take into account a drivers penalty laps and/or lap sections as well as their actual lap count when determining if the heat should end
  - Fixed Gameport connection so that it properly supports refueling
  - Fixed heat leader LEDs to prevent the LED from incorrectly changing after drivers complete their first lap in a heat.
  - Fixed Step-Up races to work with the RC point system.
  - Added heat reversal option
  - Additional localization support
  - French translation
- 1.9.0.0
  - Fixed buffer overflow with certain arduino configurations. The fix requires the sketch to be re-uploaded to your arduino.
  - Fixed crash related to the “Lane Font Color” not being set to a valid color
  - Fixed issues with exporting xls fastest lap times in the middle of a race
  - Fixed issues with exporting xls fastest lap times causing the fastest time to always be reported as the fastest time the driver raced in a given lane
  - Fixed race state flags for lap based/allow finish races in which the white flag wasn’t coming out when the first driver hits the lap limit.
  - Added Guided Race and Driver setups to make configuration even easier than before and in doing so removed the wizards for driver and race creation
  - Added debounce option to trakmate interface
  - Added debounce to the arduino sketch (requires new sketch upload)
  - Changed race stats screen to allow removal of a single race. This makes the dialog a bit clearer regarding what’s actually going to happen.
  - Fixed localization support. Foreign languages like German that use ‘,‘ as a decimal point will no longer have the decimal values truncated in the various configuration options. \*\*\*Translators wanted\*\*\*
- 1.8.1.0



- Fixed a rare crash when using “allow finish (auto segments)” option. If a driver did not run any laps and the allow finish portion of the heat kicks in, then the driver runs laps, RC would crash out.
- Trakmate hardware timing now being used.
- Fixed arduino config screen so that it sends pin modes when pin settings change
- Fixed arduino interface from adding pause time to lap times when a lane is NOT assigned a pin on the arduino
- Added support for #argb color values as lane and font colors
- Clicking ‘Accept’ in the add laps window (opened by clicking a drivers lap count on the race day window) now closes the dialog when adding laps for a single driver
- Stopped preference window from resetting the RCLive server cache size upon entering the screen.
- Removed Team View Window from Race Day screens. Team support was removed a long time ago.
- Added Defer Heat option into the race director screen. This is a convenient way to move a heat to the end of a race if for some reason the drivers in the heat are not ready to race. \*\*\*If this is done in a step-up race, the deferred heat essentially becomes the finals heat. It’s probably not wise to use it in a step up race.
- Added median lap time calculation display and export options
- Changed space bar to behave exactly as the track call button does
- Added image display for when a driver completes a lap during the drift time. The default is a red start in the lap count column on the race screen, but is configurable through the Expert Race Setup.
- Removed arduino patch version number check
- Changed practice race to default to auto start.
- Race formats now support a breakout time. If set, any lap faster than the breakout time will not be counted and the time for the lap will be added to the next counted lap the driver completes. This is not the same as the min lap time. It allows you to run one fast car with one slow car and still give the slow car a fighting chance by limiting the fast cars lap times.
- v1.8.0.0
  - Changed web server response headers to allow better access to the server data.
  - Fixed color scheme for Expert Driver and Expert Race config screens.
  - Fixed problems with using phidget boards for lap counting and/or refueling
  - Fixed combo box display issues on the Expert Race Setup screen
  - Fixed auto segment calculations.
  - Fixed issue with average lap time taking into account lap segments in the calculation
  - Added scaling to the leader board

- Changed leader board -1 values to “—“ for a better display
- Fixed rare crash condition in which the race director closes the race window, but never makes a choice on the confirmation dialog. If any form of auto advance was set, RC would crash when the auto advance timer expired. Now, when the Race Director tries to close the window, it works as if they also pressed the call button. So the heat will pause, auto advance timers will stop, start windows will close, etc...
- Changed heat tie break for point based heats to average lap time
- Enabled multi track interface support and debug logging from the new options/preferences menu rather than requiring a custom release
- Fixed false starts when using per lane relays and time penalties to count the false start as the reaction time so the next time over the sensor the lap is counted as a lap
- Changed it so that lap and time penalties for false starts are reset if the heat is reset, but only if the “Reset on false start” option is NOT set.
- Enabled live web update data to be sent by RC during practice races. This is untested and more than likely would ultimately require new HTML screens to better support practice races. In theory RC at least supports this effort now.
- RC Live client files are now included with the installer. **\*\*\*Note: the files are installed to <install directory>/data/html and the uninstaller blows the entire directory away.** In other words, RC owns that directory. If you choose to create custom html/js files you must put them in your own directory so that the RC installer/uninstaller won't overwrite/delete them.
- RC Live client fixes. If you use an unmodified version of the client you'll get these fixes automatically and you'll want them.
- Added port auto detection for the Trakmate interface
- Added support for Bepfe/Titus track interface
- Added support for USB slotmaster track interface
- Added arduino support including the sketch needed and documentation including parts lists for somebody to build a low cost, very solid track interface. Special thanks to Slingshot for once again doing the bulk of the work on this one. This is a full features interface, it supports absolutely everything RC has to offer and is extendable by the community.

- v1.7.1.0

- Added ability to manually enter port address for parallel port track interfaces
- Added ability to generate debug logs and to use multi-track interface mode all from the Race Day Setup screen. These options should only be used by expert users that know what they're doing

- Fixed issue with the min lap time audio playing on legitimate laps. This only happens with certain body types and certain track interfaces.
  
- v1.7.0.0
  - Added support for live web updates via a web server built into RC. HTML/JS sample client made available through an external download.
  - Added power cuts to lanes in which drivers false start. Users must have per lane relays installed and configured and they must set the false start time penalty in the race configuration. If you use hot starts with false starts, this is by far the best way to penalize a driver for the false start
  - Added permanent window position storage and made it so that the reset window positions button should now work under all situations. No more window position reset when you update versions <YAY>!!!
  - Added laps led for both heat and over all race. These can be displayed in the UI with the "Led\_" prefix and I've added it to the race results and kitchen sink but no other screen. Led laps are also exported in the default xls template
  - Added ability to display heat "total lap time" on the Race Day screen. Use the "TotalHeatTime\_" prefix to access it.
  - Added right double clicking to the driver names to add them to the race on the "Modify Heats" screen. This makes adding late joiners more consistent with the main Race Day Setup screen.
  - Reworked setup interfaces to make them more user friendly and unify them with my other RMS software.
  - Added new allow finish type which will allow drivers to complete one final lap, and rather than counting that lap towards their lap total it uses the time it took to finish the lap to estimate sections
  - Added sound effect for when a driver records a lap faster than the min lap time. This is very helpful in knowing when something is wrong on the track and/or your race is configured incorrectly.
  - Fixed season "race carry over percentage" which was off by a factor of 100.
  - Fixed a fuel usage issue that allowed laps that came after a lap that was under the min lap time to use far too much fuel.

- Fixed crash caused by the restart on false start if you cancel the restart dialog. This dialog no longer comes up as it was set by configuration to restart.
  - Fixed an issue where lap based races prevented the event auto advance feature from working.
  - Fixed track setup so that lane color and length boxes update when a new lane is selected.
  - Fixed issue with phidget 0/0/4 phidget interface causes “drift” laps to have incorrect lap times. This will only affect you if you downloaded a multi-interface build which is not common. **\*\*\* For this fix to be applied you must remove your phidget 0/0/4 interface, update your race so it no longer has the configuration, then re-add the phidget 0/0/4 to your interface setup.**
  - Attempted to make drop down menus in the setup screens more readable
  - Changed reaction time to only affect lap times for total time, and lap based heats. Best lap time heats will not have the reaction time put into the lap time
  - Changed when dropN races start dropping heats for a driver. Whereas before it dropped heats as soon as there were more heats than the N value which made for a very confusing leader board, I now drop heats only when a driver has N or few heats left in their race. So if N is 1 all heats are counting until they race in their last heat. In the last heat the leader board will only change when the last heat out scores any other heat the driver has raced in.
  - Fixed minor issues with the custom rotation file editor
  - Added more robust error messages for custom rotation files
- v1.6.3.0
    - **Critical fix to v1.6.2.0 for Trakmate users.** Fixed a bug with the trakmate interface in which if a car tripped the sensor after a yellow flag restart within 0.5 seconds the lap would be missed.
    - Fixed parallel port control register support so RC now properly supports 8 lane tracks through the parallel port
    - Fixed rare crash when editing laps for a race after the number of lanes on the track has been changed to be lower than the saved race.
    - Fixed issue with removing custom heat files from an already created race.
    - Changed gap calculations to support fastest lap and total time races. Also

changed lap calculation to better reflect gap of the second place car.

- Added Hours to the race timer so that for enduro races the user does not have to convert a display of 180 minutes to 3 hours. All properties remain unchanged, so the race time is still expressed in seconds, but the display is now easier to read.

- v1.6.2.0

- Fixed bug related to Trakmate interface and having 2 or more cars refueling at the same exact time
- Fixed bug with resetting a heat not resetting a drivers "total lap time". This would cause "total time" races and any lap count ties to be handled incorrectly in the final standings
- Fixed issue with "Allow all to Complete" lap based races having the power cut to individual lanes incorrectly when using per lane relays.
- Fixed issue with pausing a heat and then closing the restart window before the heat resumes.
- Fixed DS200 lap counting. It was counting on lanes 7 and 8 and is now counting on lanes 1 and 2 appropriately.
- Fixed issue with using multiple 0/0/4 phidget interfaces to control lane power on 8 lanes. This will only affect you if you downloaded a multi-interface build which is not common.
- Fixed issue that caused the modify heats screen to remove a driver from a lane by simply left clicking.
- Fixed issue with invalid custom rotation files crashing the app.
- Added Power control to the race director. This allows you to override the power state of the system. RC does not know you did this though and will continue the race 100% as if you didn't change the power. This feature allows you to run un-timed practice laps between heats, and cut power to a driver in an emergency during the race.
- Added ability to use space bar to advance to the next heat.
- Added right mouse button double clicking the 'Drivers Available' or the 'Drivers Racing' to add/remove drivers from a race on the race day setup screen.

- v1.6.1.0

- Fixed crash bug with restarting a heat.
- v1.6.0.0
  - Added Phidget track interface support
  - Added DSxx track interface support
  - Added per lane relay support for the parallel port interface. Lane power will now be disabled when a car finishes its heat (in an 'allow finish' race) or when it runs out a fuel in a 'fuel end heat' race. More lane control to come for crash and burn races, and to assign time penalties to false starts, and to disable track power on empty lanes.
  - Added "#Heat" support to the default export template which outputs the heat number for the sheet.
  - Added Practice 4L and 6L xaml files so the auto file picker will work properly and updated the default database to use them.
  - Added "NextHeat" as a window label. This window works similar to the On Deck window but shows all the drivers in the upcoming heat rather than just sit outs coming in.
  - Added edit stats screen in both the out of race stats screen and during a race. This can be used to 'ignore' laps that for whatever reason are too fast and set race records. Why would you do this? First, so that you can have proper record tracking, but second because in some race formats, the records affect season scoring and/or the race leader board so getting the fastest times correct can be very important.
  - Added fuel support to web cam races. You must add two new zones in your web cam software. The first zone should be the exact same zone as your Start/Finish line zone. When triggered it should run 'webcam.exe -pitenter [1-8]' where 1-8 is the lane number. The second trigger is one right after the Start/Finish and Pit Enter trigger. When a car enters this trigger hot spot you must run 'webcam.exe -pitexit [1-8]' to end the pit stop. A car moving through this area at high speed should not get any fuel back, however a car that stops at the pitenter trigger and before the pitexit trigger will refuel. \*\*\*NOTE: because of how analog fuel works, currently the pit enter MUST be the same location as the start/finish line.
  - Added a new allow finish type which allows each driver to race to the lap limit in lap based races. This is a very nice feature for Rally type races.

- Added 'TotalTime' to the xls export so that you can export the drivers total time for the entire race, including adjusted heat times and penalty times
- Added 'LaneTotalTimes' to the xls export so that you can export each drivers lane total lap time. This works just like 'LaneScores' but for total time instead of the lane scoring.
- Added 'HeatTotalLapTime' to the xls export so that you can export the total lap time for a driver in a given heat.
- Updated the modify heat screen to use drag and drop which should make it a bit easier to use.
- Removed step up number from the modify heat screen. Step numbers are now auto-assigned making it easy for multi-step up races to be setup.
- Changed season export to use the last known name of a driver. This means that if a driver changes their name, the season stats will show the name they are currently/last used.
- Fixed rare case in which adjusted lap counts could be mis-handled. This only occurred if you stood on one foot and did the hokey pokey after resetting a heat. This fix will correct 99% of these cases as well (only step up races can't be 100% fixed). Simply re-export the race and the corrections will be automatically handled.
- Fixed crash if you setup demo mode to have a time that is not greater than 0 for any of the fields.
- Fixed startup crash if you have no races in your database.
- Fixed bug preventing in race exporting of the xls data at the end of the race.
- Fixed stat export in which a new lap record would not be displayed as **\*\*\*New\*\*\*** in some cases when the record indeed was set in the race being exported.
- Fixed stat export in which season summary sometimes displayed 0 season points for a drivers race when they actually earned points for that race.
- Fixed missing image in the parallel port configuration.
- v1.5.0.0
  - **\*\*\* Removed support for all previously saved race stats. If you want the stats, export them to excel before updating to this version. Once updated you will**

**no longer have access to the statistics saved from older versions of RC \*\*\***

- Fixed rare crash at the end of a race that had no counted laps completed if that race was the first race ever run to completion.
- Fixed rare crash when hitting the track call button just after the race ends.
- Fixed heat standings if you modify a heat that is not the current heat.
- Minor spelling fixes in the different Wizard files.
- Fixed late joiner screen so that you can better see your selections. The screen still isn't great, but now you don't have to guess as to what you have selected.
- Made late joiner screen resizable
- Fixed bug with adding late joiners to a race that is not the first race in an event (first race in an event worked, all further races may have had problems).
- Fixed potential startup crash regarding saved settings
- Updated all audio resources to sounds we have legal rights to release.
- Added ability to export current race progress from the race itself
- Added confirmation dialogs for deleting race stats
- Added default selection in the Expert setup screens
- Added ability to select drivers heat lap count and bring up an add lap/sections dialog that will only allow you to edit that drivers sections for that particular heat
- Added more 'Are you sure' dialogs for race director options that may not be the desired action
- Tried to clarify the race directory start and restart options
- Made it so a race or an event can be selected, but not both. This will hopefully help with the confusion regarding running a race versus an event.
- Alphabetized all the database views that show elements with names. Things like race lists, driver lists, car lists, etc.
- Made first lap include reaction time (if applicable). This makes it so that for the first lap, the average lap time, best lap time, and lap time are all the same.
- Added option to discount any lap times that contained 'drifting' as a best/record



lap. This is useful if your cars can drift a large distance and you use a lot of yellow flags. This only affects records, driver bests, and F1 style bonus points. Total lap counts and total times are not affected.

- Changed precision display on the practice race day files. RC is accurate to 0.001 seconds so displaying any more decimal places than that is really meaningless.
- Added automatic race backup. At the end of every heat the race is automatically saved to disk. If RC shutdown artificially (computer crash, power outage, etc) when you try to run the same race or event again you will be prompted to load the saved race. If you complete the race or end it in any fashion manually the saved backup is removed
- Added the ability to save a paused race to disk. The file management for the saved race is 100% up to you so the saved race can be restored at any time as many times as you like. However, stats for any race are only saved once. If you finish a saved race a second time, the second race will overwrite the first races stat data
- Added ability to add/remove lap sections after the race is over and exited. Simply load the stat file as a race, make your changes and close it out.
- Added fully customizable excel stat exporting. You can now specify exactly how you want the export to occur by setting up an export template. A default template has been provided which exports the stats much like previous versions.
- Added virtually all stats RC holds as possible values to export
- Added reasonable default file names to xls export based on race name and time.
- Added ability to pause/start races with one key click (default is the space bar). In situations where both the race director/start and race director/pause menu options are available (like during the auto start period), the toggle feature will pause first. By default it's the space bar. Any user using a custom XAML file will need to update their custom file to gain this support.
- Added season support capable of assigning points to each driver in the race and being able to add bonus points based on the F1 style point system already in place for race scoring. This is just the tip of the iceberg with this feature. Suggestions welcome
- Added more COM ports to the Trackmate interface as a temporary hack to get people with a lot of ports working.

- Added more buttons to the gameport configuration. At least one user needed buttons 10 and 12.
- Added Options/Reset Windows to the Race Day setup screen so you can reset window positions should the need ever arise.
- v1.4.5.0
  - Fixed crash in parallel port interface if you have setup less pit stop pins than lap pins (very rare) and then you pause the race
  - Fixed timing issue relating to drift after the heat has ended
  - Fixed support for car filters. Cars that do not pass the filter are no longer allowed to be dragged into the driver for the race.
  - Fixed default file extension for exporting race stats to .xls
  - Fixed default file extensions for custom rotation files to .txt
  - Fixed bug on practice screen in which after selecting a button to clear a lane, the space bar and enter keys would then trigger that button.
  - Fixed bug in which if the first lap a driver recorded was a drift lap, the reaction time calculation would be incorrect causing the next lap run by the driver to be very large.
  - Fixed bug in which if the first actual lap a driver recorded was a drift lap and the option 'Start at Current' was set, the lap time for the drift lap would be too large.
  - Fixed bug in which if the final lap of the race was a drift lap, it would not be exported in the xls stats, however it would be correct in the RMS display.
  - Fixed debug message when a xaml file doesn't load to include the programmer reason why. Hopefully this will be enough to fix the problem. It's better than it was at least.
  - Added new friendly Round Robin heat rotation. This rotation is friendly to late joiners because it rotates in the highest seeded drivers first. This means that if a late joiner comes in, you can add them to the race and use the 'automate' button to re-setup the heat structure much longer because the lowest seed doesn't rotate in in the second heat. If you don't care what the heat structure really looks like for a round robin, this is your best option if you expect people to show up late to the race.

- Added ability to count a lap when using the parallel port and alternate pit lane sensors. This allows your pit row to bypass the actual start finish line if you want.
  - Added ability to setup step up lanes in the Modify Heat dialog. Be careful, invalid step ups will result in empty lanes or worse!
  - Made the Race Start lamp window resizable.
  - Added race half over male/female audio files for callouts. They're not used by the application by default, but they can be added just like any other callout.
- v1.4.4.0
    - Fixed bug in which under specific fuel race configurations, laps would continue to be counted for drivers that ran out of fuel.
    - Fixed bug with lap based races and the Allow Finish property in which each driver needed to reach the lap count rather than complete just one more lap
    - Fixed bug in which a driver that ran out of fuel when the End Heat property was set would not correctly mark itself as having finished the heat. In races in which the Allow Finish property were set this would cause the heat to never end. In this case the Race Directory would have to manually skip the heat.
    - Fixed bugs with the parallel port configuration screen in which the check boxes on the screen were not initialized to the proper values based on the current settings.
    - Fixed add laps dialog to allow scrolling if the track has more than 5 lanes. This allows selection of those extra lanes
    - Fixed pit started callout to only play if the car is allowed to refuel
    - Fixed false starts so that only one per heat can occur. This should prevent a weird case in which a car can receive more than one false start penalty on the same false start
    - Fixed potential issues with F1 style scoring and the bonus point system.
    - Fixed issues with adding late joiners causing the loss of heat data. This only caused problems when the race data was exported.
    - Fixed issues with late joiners being added in ways that altered heats that were already completed.
    - Fixed corrupt race stats when using Operating Systems setup with a foreign

language that uses commas (,) for decimal points. Not only would the saved data be corrupt, it would crash RC when trying to export it.

- Fixed overall stat export lap totals to include the adjusted lap count and penalty laps assigned to a driver.
- Optimized parallel port configuration screen allowing slower computers to access the screen without hanging the application
- Moved the database and log directory to the systems APP DATA folder. This is done so that newer operating systems like Windows7 do not require administrative privileges to run the app. This also helps guarantee that debug or crash logs can be created.
- Changed gap calculation to be more useful especially when the two drivers being compared are not on the same lap.
- Updated webcam.exe to accept '-noerrordisplay' which will disable all error messages. This is useful if you want to run the web camera interface but don't always run RC.
- Added 6 lane race day xaml files.
- Updated Race Results xaml to a better design. All the same information is there, but it is presented much nicer.
- Added group support to single lane and single lane solo heat structures
- Added step up support to single lane heat structures.
- Group races can now be setup to rotate groups on and off the track for each heat
- Added an option to replicate the heat structure as many times as you want for a race. This means you could setup a round robin, but have the race run with every driver running on every lane two or three times. A lot of people were using custom rotation files to accomplish this, and now that is no longer necessary.
- Added ability to display driver total lap time and total lap count even if these are not the methods used to score the race. Use "TotalLaps" and "TotalTime" to display these values.
- Added ability to display the extended time a driver gets when a time based race also uses the allow finish option. This value is the time used to determine the tie breaker between two drivers that have the same lap count in races that are

scored by lap count. Use “TotalExtendedTime” to display this value

- Added ability to display the heat finish value. For lap based races this would be the lap count the drivers need to reach and for time based races this would be the total time the heat lasts for. Use “EndHeatValue” to display this value.
  - Added ability to display the heat lap count in the same way the race timer works. That is, in a lap based heat it counts down and when at 0 the heat is ending, for time based races, it counts up showing the heat leaders lap count. Use “RaceLaps” to display this value.
  - Added confirmation for quitting the race. This prevents accidental exits.
- v1.4.3.0
    - Fixed crash watch dog to detect all crashes, not just the ones it felt like
    - Fixed Race Wizard to account for new Race Day Xaml Prefix properties
    - Fixed rare crash in the parallel port configuration screen
  - v1.4.2.0
    - Fixed crash when removing a driver from a heat after the race has started
    - Fixed crash when setting up initial fuel level greater than fuel capacity
    - Fixed bug in which you could not modify the current heat you were on
    - Fixed automatic heat setup when modifying heats after a race starts
    - Fixed team scoring
    - Fixed drifting at the end of a heat/race
    - Fixed heat audio callouts when running lap based races
    - Simplified Track user interface by removing the ability to add/remove tracks. A common confusion was why track settings were not taking affect when it was because a new track was added, but no race actually used that track. If you have created more than one track, your extra tracks will still be usable, however you cannot create more.
    - Added all the missing stock race day files. There are now files for 8, 4, 3, and 2 lane tracks for every combination of fuel, image, and static displays.
    - Changed how the Race Day XAML File property works:

- It can now be a prefix in which #L, static, images, fuel will be automatically appended to in an attempt to use the best fit screen. On the UI setup screen there are now properties to control what gets appended onto the file prefix.
    - Anything after the first \_ in the file path will be removed and everything before it will be the prefix
    - You can still manually type in your full path name and force a specific file to be used. However this should not be needed. Even if you create your own custom screens, if you name them with the same convention used by the stock files, you can then use your prefix and get things working.
  - Added seed display ability. This is the ranking a driver started a race in. By default it only shows up as a (#) value in the top5 window and the leader board, but any user can create a custom race screen and add it there as well if they wanted.
  - Optimized Race Day display for better performance on lower end computers.
- v1.4.1.0
  - Fixed drift support which was broken by the fuel changes
  - Fixed Track management so that when you configure your interface it shows the current settings rather than the defaults. \*\*\*Note, that if you have just selected an interface it will wipe out any old settings and change to the defaults to the new type selected.
- v1.4.0.0
  - Fixed bug in Trakmate interface in which if you parked the car under the IR sensor you would continuously record laps at a min lap time rate
  - Fixed potential crash if sound drivers aren't setup properly. Now you just get no sound.
  - Fixed issue with updating a Track in the Expert Track Setup in which after the update the wrong track would be selected.
  - Fixed an invalid race configuration from causing a crash when running the bogus race as part of an event.
  - Fixed bug with total time race standings display

- Fixed lots of issues with the Driver Wizard not setting up audio properly.
- Fixed crash caused by assigning a custom rotation file that was built for fewer lanes than the track being used. Now if the custom rotation file lane count does not match the track lane count it is ignored.
- Fixed issue with track interfaces sticking around after going into the track interface configuration.
- Added Game Port support which includes the Carlson USB Sensor Trak.
- Added analog fuel support
  - Fuel is consumed based on lap time (configurable)
  - Refueling occurs by parking at the track sensor or alternate sensors that represent pit bays
  - Unknown how well it works for things like the webcam interface and deadstrips.
- Added beep and zoom sounds to practice laps. These are NOT currently configurable, but use the default lap/best lap sounds provided upon install.
- Made the Add Sections dialog a bit more usable by making the driver selected stand out better
- Added drag and drop race day setup.
- Added tips of the day
- Added different levels of race wizard setup. The goal is to slowly introduce the user to the various features of RC by order of importance/relevance/common usage.
- Added 2 and 3 lane practice xaml files
- Added Top 5 race leader display with scalable window.
- Reworked parallel port interface setup
  - Support separate pit sensors and lap sensors.
  - Pin based setup making it easier to see what your configuration is.
  - LED light bridge support enabled
    - Light bridge support is currently custom tailored to my bridge

but can easily be changed upon request.

- Added variable font sizes to the Race Day driver data so that the text could be as large as possible making it easier to read from a distance.
- Added load percentage counter on the splash screen.
- v1.3.1.0
  - Fixed bug that allowed a heat to be restarted even though it wasn't already in progress. This would cause either a loss of heat data or a crash if it was done in the first heat.
- v1.3.0.0
  - Fixed problem in track manager in which the wrong interface was being displayed as selected after certain track changes.
  - Fixed Trakmate interface to properly support dead strips and reed switch track interfaces
  - Fixed minor issue with global standings when dropping lowest heat values
  - Fixed Trakmate configuration invert relay to be consistent in the setup screen and during race day (they were backwards of each other).
  - Fixed Race Wizard Audio Start Prefix which wasn't being changed if the text box was double clicked.
  - Fixed rare case in which a car drifting over the start/finish line could affect the heat standings and store incorrect stats.
  - Fixed crash bug that occurs when you close the main window while the start countdown is running
  - Fixed crash bug that occurred if the decimals field was not supplied for timing data in xaml files
  - Fixed bug with custom rotation files in which a less than optimal rotation file could sometimes be selected as the rotation to use.
  - Fixed loading custom file in which the heat rotations would be incorrect unless you load the file two consecutive times
  - Fixed Race Manager custom rotation listbox to properly put scroll bars up
  - Added auto heat advancement timer so you can race through the heat rotations



100% hands free including not using the track call button to advance the heats.

- Added multi-selection to the Race Setup Driver listbox
  - Optimized Race Day rendering slightly to make the heat results window possible.
  - Re-disabled saving stats for practice races
  - Added ability to display laps in the order they occurred as well as starting from the most recent lap completed.
  - Added new heat results screen that gives stats relating to the current heat including 50 laps worth of timing data for each driver.
  - Added new race results screen that gives a summary of the race. It is similar to the leader board but with a lot more information on it.
  - Added record score to the stats stored per race and now display the score that won a particular race in the manage/stats screen.
  - Added yellow coloring in wizard table of contents to identify which wizard properties have been changed
  - Added check in wizard to see if any properties have been changed and only if so prompt the user to make sure they want to discard the changes
  - Added more text to some of the wizards in an attempt to improve their usefulness
  - Added ability to start in front of or behind the lap sensor.
  - Added new best lap callouts for practice races so that drivers can gauge their performance against themselves as they run laps.
  - Made radio buttons and checkboxes more viewable. Some systems could not see what was selected at all.
  - Cleaned up and simplified Trakmate code.
- v1.2.1.0
    - Fixed crash in the race wizard Image Setup screens
    - Moved Track Wizard Interface configure button so it is on screen without resizing the window.

- Fixed Track Wizard to invalidate itself after creation because the track name is now a duplicate name.
- Bullet proofed DirectSound resource loading as there have been rare reports of problems related to loading sound files
- Fixed issues with Custom Heat validation
- Fixed bug in which stopping the re-start countdown twice in a row would reset the race time
- v1.2.0.0
  - Fixed crash bug when removing a driver from a heat after the race has started.
  - Fixed Manage/Race screen so that the heat view will display properly even if there is no race selected (ie: creating a new race from scratch).
  - Fixed 'start at current' option to start a drivers first heat behind the lap sensor.
  - Fixed 'start at current' option so that it respects the 'Min Lap Time' option on the first lap of each heat.
  - Fixed crash when using Track Wizard
  - Fixed bug that allowed invalid race setups by incorrectly identifying invalid step-up values
  - Fixed bug with race events in which driver data would sometimes carryover from one event to the next
  - Fixed bug in custom heat generation in which the 'round robin' template setup for a 2 person step up race.
  - Fixed custom heat validation
  - Fixed late joiner automation to properly check for heat changes (may not have been needed but the code is now cleaner)
  - Fixed bug in manage/event screen preventing track names to show up in the 'current race' list box.
  - Fixed bug with 'quick changing' driver lane assignments before a heat begins
  - Fixed timing issue in which reaction time was not being counted towards a drivers total time

- Fixed resetting race stats to also reset race records and best performance tracking
- Bullet proofed window placement in the event the width/height is somehow read in as less than or equal to 0
- Made database update failures a fatal error. They really are fatal and can hide problem until much later. RC will now crash out if this happens to draw attention to the application error.
- Made all management windows resizable and save/load current window position. They will all default to open up full screen, but can be changed.
- Added support for special characters like single quotes in database entries (like names, file paths, etc.)
- Dramatically improved performance of data base updates
- Added basic team support to the race configurations
- Added ability to add/remove sections to any heat already completed or currently being run. No more waiting until the end of the heat.
- Changed race day windows to have the light blue background which is constant with the default race day screens.
- Made track call button abort the race start if pressed during the start sequence
- Minor changes to the provide Race UI (user interface) xaml files
  - Changed 'start race' to 'start heat' in all the provide interface screens
  - Changed on deck text color from white to black to make it visible on a white lane
- Disabled stat saving for practice races
- Updated demo mode to trigger 8 laps
- Added info on splash screen helping to display load progress
- Added race winner to the stat display for a race
- Added new overall race record stats and new current race stats
- Added new audio callouts for the race
  - Callouts for overall fastest lap and overall fastest laps per lane

- Callouts for best lap during a race and best lap per lane during a single race
    - Callouts for new race and heat leaders
  - Added new F1 style scoring bonus points
    - Bonus points for best race lap time
    - Bonus points for best race lap time per lane
    - Bonus points for best heat lap time
  - Added drift support enabling laps to be counted for a brief time after the heat has ended or a yellow flag has occurred
  - Added European Round Robin as a default template for custom heat generation
  - Added European Round Robin as a default template when a late joiner arrives
  - Implemented support for un-checking 'Balance Seeds' for group racing. This enables a 'winners' and 'losers' type bracket if desired.
  - Added links to the About screen.
  - Added 3 lane xaml files to the install
  - Added 8 lane xaml file to the install
  - Added the Kitchensink\_4l.xaml file with comments
- v1.1.2
  - Fixed crash bug with false starts.
- v1.1.1
  - Fixed bug with parallel port setup preventing anything but the default values from being used.
- v1.1.0
  - Fixed bug in which Race Management would undo group settings when selecting a new race.
  - Fixed crash bug when using certain group settings and having no drivers added in the Race Day Setup Screen
  - Fixed all windows that display lanes and lane colors to draw at the same length

- Added Webcam support through a separate executable named 'Webcam.exe'. All operating systems including WindowsXP require RC to be run with administrative permission for this track interface to work. Simply configure your track with a webcam interface and run the Webcam.exe with the following command lines:
  - -callbutton
    - Signals RC that a callbutton has been pressed
  - -lap [#]
    - Signals RC that a lap has occurred on the specified lap (#). Laps can range from [1, 8].
- Added 'On Deck' driver support
- Added Group text to heat lists for better display
- Added saving/restoring window sizes and position
- Added new Heat Rotation formats:
  - Solo/Any Lane allows the racer to race alone on the track and record laps/times on any lap the track has
  - European Round Robin allows racers to drive on every lane on the track while never racing alongside the same driver twice.
  - Custom Round Robin allows the user to specify a sequence of numbers specifying the lane each driver will drive on for the heat. This is exactly the same behavior as the Trakmate 'custom rotation'
- v1.0.0
  - Initial non-beta release