

Using the Richta Rallymaster App for TSD Rally

Author: Jay Johannes, Milwaukee Region SCCA

Version 1.0

Comments welcome at jayjohannes49@gmail.com

Table of Contents

- 1. Overview 5**
 - 1.1. What is the Richta Scoring System? 5**
 - 1.2. Available Downloads 5**
 - 1.3. How Does the System Work? 5**
 - 1.4. A Note About GPS Accuracy 6**
- 2. Types of Richta Controls 7**
 - 2.1. Time Check 7**
 - 2.2. Restart Controls 7**
 - 2.2.1. Start at Car Zero Time (24 hour Format) 7
 - 2.2.2. Flying Start 7
 - 2.2.3. Flying Start – Truncate to 00:30 8
 - 2.2.4. Flying Start Next Minute 8
 - 2.3. Timing Controls 8**
 - 2.3.1. Timed from Previous Checkpoint 8
 - 2.3.2. Timed From Previous TOD Restart or Flying Start 9
 - 2.4. Course Following Control Types 10**
 - 2.4.1. On Course 10
 - 2.4.2. Off Course 10
- 3. Steps to Creating the Rally Using Richta Rallymaster 11**
 - 3.1. STEP 0: Study the Materials and Create a Practice Event 11**
 - 3.2. STEP 1: Download the App 11**
 - 3.3. STEP 2: Contact Rich Bireta to Register Your Event 11**
 - 3.4. STEP 3: Initialize the Rallymaster Event 11**
 - 3.5. STEP 4: Lay Out Your Course 12**
 - 3.6. STEP 5: Place GPS Controls 13**
 - 3.7. STEP 6: Measure the Course and Validate GPS Control Locations 13**
 - 3.8. STEP 7: Calculate Leg Times for Each Control 14**
 - 3.9. STEP 8: Prechecks 15**
 - 3.10. STEP 9: Adjust Control Locations and Leg Times Based on Precheck 15**
 - 3.11. STEP 10: More Prechecks 16**
 - 3.12. STEP 11: Rally Day 17**
 - 3.12.1. Before Registration 17
 - 3.12.2. During the Event 17
 - 3.12.3. At the Finish Location 17
 - 3.13. STEP 12: Post Event Reporting and Accounting 18**
- 4. Validation Report 19**
- 5. Use of Open/Close Time and Bearing Settings 20**
- 6. Timing Calculations 21**
- 7. Duties of the Monitoring and Scoring Staff 23**
 - 7.1. Making Adjustments During the Rally 23**
 - 7.1.1. Rescoring a Leg 24
 - 7.1.2. Discarding a Leg 24
- 8. Using the Map Capability in Rallymaster 25**

- 9. Use of the Scoreboard Function..... 26**
- 10. Advanced Menu..... 27**
- 11. Exporting and Displaying Scores..... 28**
- 12. Technical Details on Data Transfer Between the Competitor and Rallymaster App..... 30**
- 13. Use of Backup Devices in the Competitor App..... 31**

Table of Figures

Figure 1: Event Main Menu 11

Figure 2: Event Settings 12

Figure 3: Change CP Location 13

Figure 4: Change CP Time 14

Figure 5: Checkpoints Display 16

Figure 6: CARS Screen 17

Figure 7: CARS Detail Screen 17

Figure 8: Validation Report 19

Figure 9 Compass Bearing 20

Figure 10: Map Display 25

Figure 11: Scoreboard 26

Figure 12: Advanced Menu 27

Figure 13: Example Scoresheet 28

Figure 14: Sorted Scoresheet 29

1. Overview

This guide's purpose is to help a rallymaster get started creating a TSD rally using the Richta scoring system. The guide assumes that the rallymaster is familiar with the basic process of writing a TSD rally, so it will not cover fundamentals such as choosing a route, pacing, calculating leg times, etc. There are other guides online for this purpose, and any new rallymaster should study those before reading this guide.

1.1. What is the Richta Scoring System?

The Richta Scoring System is a set of apps developed by Rich Bireta (rbireta@gmail.com) to aid in creating, timing and scoring a TSD rally. The apps have been developed to run on both Android and Apple iOS devices. They may be downloaded from the Apple App store (Apple) or the Google Play store (Android).

1.2. Available Downloads

- *Richta Rallymaster* (Android), *Rally Master* (iOS)– App used to create a TSD Rally. It will place checkpoints and set times during layout. It will also monitor progress and do scoring during the event. The use of this app is the primary subject of this guide
- *Richta Competitor* – App used by contestants to detect and transmit individual scores during the rally. Each entrant needs to be running the Competitor app on a dedicated device during the event. The use of the Competitor app is described in a separate guide.
- *Richta Scoreboard* – App displays a real time scoreboard of all contestants during the event. Mostly for competitor use, as the function is also built into the Rallymaster app.
- *Richta Odometer* – App provides an accurate odometer with measurements displaying mileages to .001 mile (Android) or .01 mile (iOS) resolution using GPS. The app is not quite as accurate as a dedicated wheel based rally odometer, but can be run with any vehicle without physical connections. It runs on any Android or Apple device and is very cost effective (\$5.00 download on Android, \$10 on iOS).
- *Richta Simple Rally Computer* – App provides a good emulation of a traditional rally computer, using GPS instead of wheel driven measurement techniques.
- *Richta Rally Calculator* - App provides the calculating capability from *Richta Simple Rally Computer*, but mileages must be manually entered from an external source. Used for limited class, or when an odometer is not permitted.
- *Richta Checkpoint Clock* - App provides a GPS sourced real time clock with split and logging capabilities.

1.3. How Does the System Work?

The rallymaster uses the *Rallymaster* app to precisely place a number of timed checkpoints using the GPS capability built in to the smartphone. The rallymaster then calculates the perfect time to each checkpoint using the normal techniques and enters this timing information into the *Rallymaster* app for each checkpoint.

The competitor registers for the rally by entering the car number and class when initializing the *Competitor* app. As the competitor runs the course, the competitor will encounter the checkpoints using GPS positioning data. When a checkpoint is encountered, the competitor app will compare the time it entered the checkpoint to the perfect time stored in the event database. It uses this information to generate a score for that leg. It also keeps a running total of all leg scores and displays the overall contestant score. Finally, it transmits this information back to a central database which is read by the *Rallymaster* app to provide scoring.

The *rallymaster* app has additional features on event day that can provide real time adjustment of leg timing, contestant missed checkpoints, and discarding a leg. It can also display a map showing all checkpoint locations and the positions of vehicles on the course. Finally, it can provide downloadable information that can be downloaded to a spreadsheet for a friendly scoreboard display at the finish.

1.4. A Note About GPS Accuracy

The GPS built in to most smart phones (4g and above) will generally provide an accuracy of better than 20 meters, depending on cloud cover and foliage. In order to achieve best possible accuracy, you should always position the receiver (external or smartphone) such that it has a good view of the satellites. Placing it on the dashboard near the windshield is ideal. When setting checkpoint locations, it is best to take time to let the reading settle before marking the spot, and it is best to hold it outside the window for a few seconds. This will generally provide an accuracy of under 4 meters unless there is extreme blockage of the sky (e.g. overhead trees, tunnel). The GPS accuracy reported by the app is the CEP – Circular Error Probability. This is a computer measure of accuracy and is useful to indicate the approximate accuracy of the measurement, e.g. a circle of the specified radius around the reported position.

2. Types of Richta Controls

There are several types of checkpoints defined in the Richta system. Some are used to provide assurance you are on course. Some are used to begin a segment that contains one or more timing checkpoints. Finally, there are timing checkpoints where the vehicles are scored against a calculated perfect time.

2.1. Time Check

The Time Check checkpoint simply sounds a chime and displays the current time. It is not scored. This checkpoint is generally used in 3 places. First, it is generally placed as CONTROL 1 somewhere in the registration area so contestants and organizers can verify that the competitor app is properly configured for each contestant.

Second, if the rally allows for flexible start times, a time check is placed at the start of the odometer check as CONTROL 2. This alerts contestants that the event has started. It may also be placed at the end of the odometer check to allow some space before the restart for the first leg. This allows competitor feedback for the end of the odo check and provides some space before encountering the first restart checkpoint.

Finally, it can be placed anywhere on the route where an assurance is needed that the contestant is still on course. Common usage is at break locations.

2.2. Restart Controls

Restart Controls are used to set an out time to begin a segment that contains one or more timed checkpoints. There are multiple ways to configure a restart, but it is recommended that the number of ways be limited for any particular event.

2.2.1. Start at Car Zero Time (24 hour Format)

In this variation the rallymaster enters a time for the checkpoint in 24 hour format (e.g. 13:26:00) Segment timing begins at the time specified plus the contestant car number in minutes. For example car #6 would be assigned an out time of 13:32:00 when they passed this checkpoint. If the rally is running on a fixed schedule, this form of checkpoint would be placed at the start of the odometer check.

The best strategy is to place the actual checkpoint location 50 to 100 feet before the instructed restart point. You still measure and calculate times from the instructed restart point. Contestants will then receive confirmation of their actual out time before leaving the start point.

2.2.2. Flying Start

In this variation, the contestant is assigned an out time exactly equal to the time they passed the checkpoint. For example 13:26:23.3 as an out time. Timing is from that point. This style of restart is uncommon, except for performance events where the car needs to be up to speed before timing starts, or the rare event where a competitor clock is not allowed.

2.2.3. Flying Start – Truncate to 00:30

In this variation the contestant is assigned an out time equal to the :00 or :30 time prior to crossing the line. For example, a car passing this checkpoint at 14:12:13 would have a start time of 14:12:00. A car passing this checkpoint at 14:11:58 would have a start time of 14:11:30.

This is the common restart used for events that allow contestants to restart at the top of any minute.

The best strategy is to place the actual checkpoint location about 0.1 miles beyond the instructed restart point. You still measure and calculate times from the instructed restart point. This avoids the issue of contestants starting a few seconds early to get up to speed ending up timed from the previous half minute.

2.2.4. Flying Start Next Minute

In this variation the contestant is assigned an out time equal to the :00 time after crossing the line. For example, a car passing this checkpoint at 14:12:13 would have a start time of 14:13:00. A car passing this checkpoint at 14:11:58 would have a start time of 14:12:00.

This is the common restart used for trap rallies to give the contestant an extra minute to find and reason out the trap.

The best strategy is to place the actual checkpoint location about 0.1 miles beyond the instructed restart point. You still measure and calculate times from the instructed restart point. This avoids the issue of contestants starting a few seconds early to get up to speed ending up timed from the wrong minute.

2.3. Timing Controls

There are two styles of timed checkpoints available. The difference is whether the timing is based on when the competitor left the previous Restart point, or left the last checkpoint. There are advantages and disadvantages to each style, but it is important that only one of these styles be used for an event. The event general instructions should specify the type being used. Note that cars do not stop at timed checkpoints, but continue on at assigned speed to the next checkpoint or restart.

2.3.1. Timed from Previous Checkpoint

Cars are timed from the previous checkpoint (or restart point for the first checkpoint after a restart). The car is timed in and scored based on the perfect time from the previous checkpoint. This implies that the car leaves each leg with a fresh start and is on time heading for the next checkpoint.

This system favors the vehicles running seat of the pants as they don't need to worry about making up or losing time if they were early or late at the previous checkpoint. However, it is more involved for teams using computers, as they need to adjust their computers to compensate for the error at the previous checkpoint. This is not trivial as they need to factor in both their displayed error and the error reported by the app. For example, a car enters a checkpoint of this sort. The computer indicates he is 3 seconds early and the App reports a 2 second early penalty. The navigator would need to subtract the indicated time from the reported error and add a gain or pause to compensate. In This case $+2 - +3 = -1$, so he would need to adjust by adding a gain of 1 second before the next checkpoint.

Example calculation for Timed from Previous Checkpoint Controls

Inst #	Instruction	Official Miles	Elapsed Time for instruction	Leg time
34	Restart at 'FRED' CAST 30	0.00	0.00	0.00
35	PAUSE 15 sec at 'GINGER'		0.25	0.25
36	CONTROL 6	1.500	3.00 (1.5*60/30)	3.25= 3:15
37	CONTROL 7	3.000	3.00 ((3.0-1.5)*60/30)	3.00 (a new leg started, so pause and prev time don't apply)

2.3.2. Timed From Previous TOD Restart or Flying Start

Cars are timed from the previous restart point. Each leg is computed using the total distance, CASTs and pauses that have occurred since the last restart. A car that has a timing error at a checkpoint carries that error into the next leg. NOTE: CAST is an acronym for Change Average Speed To, i.e. the assigned speed for the competitor.

This style favors events that attract computer class vehicles, as they do not need to do the fancy timing adjustments at each checkpoint in a series. This style is the norm for SCCA national championship events. However, seat of the pants teams need to adjust on the fly to pause or gain time based on their score at the previous checkpoint.

There is one additional drawback for rallymasters using this style. Since the timing is cumulative, an erroneous perfect time in one leg will affect the timing of all subsequent legs until the restart. Even if the error can be localized, contestants will have lost confidence in the timing for the segment and will be trying to compensate on subsequent checkpoints. There is at least one case where the rallymaster lost 6 legs over a simple math error in the first checkpoint of a segment. For this reason, it is recommended that events using this style of timed checkpoint should limit the number of checkpoints between restarts to 3 or 4 at most.

Example calculation for Timed from previous TOD restart or Flying Start checkpoints

Inst #	Instruction	Official Miles	Elapsed Time for instruction	Leg time
34	Restart at 'FRED' CAST 30	0.00	0.00	0.00
35	PAUSE 15 sec at 'GINGER'		0.25	0.25
36	CONTROL 6	1.500	3.00 (1.5*60/30)	3.00+0.25=3:15
37	CONTROL 7	3.000	3.00 ((3.0-1.5)*60/30)	3.00+3.25=6:15 (the time and pause from prev leg added to calc time)

2.4. Course Following Control Types

These checkpoints are used on trap TSD or on gimmick rallies to determine whether a car used an on course or off course loop after a course directing trap. The vehicles are not timed, and the score is either a 0 or a max.

Rallymasters should be careful about following this type of trap with a timing checkpoint as it would potentially introduce double jeopardy with the contestant receiving 2 max scores for a single leg.

2.4.1. On Course

Cars that encounter this checkpoint receive a score of 0. Cars which do not encounter this checkpoint receive a maximum penalty.

2.4.2. Off Course

Cars that encounter this checkpoint receive a maximum penalty. Cars which do not encounter this checkpoint receive a score of 0.

3. Steps to Creating the Rally Using Richta Rallymaster

The following steps provide a general overview for creating an event and for running it on rally day

3.1. STEP 0: Study the Materials and Create a Practice Event

When using the Richta System for the first time, it is advisable to study this document thoroughly before proceeding. In addition, a wealth of information can be found at www.richtarally.com under the Documentation tab. Two especially good presentations are:

- [Getting Started With GPS Checkpoints](#)
- [Best Practices and Unusual Situations](#)

In addition, create a practice rally where you can experiment with the concepts. Rich will happily create a practice event for this purpose. Create this event close to home and limit it to just a few checkpoints. Be sure to use all the types you plan for the event. Once you have created it, checkrun it like you would for the main event. This will save time and effort in the long run.

3.2. STEP 1: Download the App.

The apps can be downloaded from the Apple App Store (Apple) or the Google Play Store (Android). The main app is named *Rallymaster – Richta GPS Checkpoints*. You will also need to download *Competitor – GPS Checkpoints* in order to perform checkruns and to validate your checkpoints. Both of these apps are free to download. The other apps are optional and may have a moderate cost. (Note: Names are slightly different in iOS version)

NOTE: Multiple copies of the *Rallymaster App* may be run at the same time. This is especially useful on rally day where the app may be run by lead, sweep and the scoring committee.

3.3. STEP 2: Contact Rich Bireta to Register Your Event.

You can contact Rich via email at rbireta@gmail.com. You will need to provide him with the event name, the event date and the entry fee. While there is no cost to register the event, Rich does charge a per car fee after the event is run. The fee varies based on entry fee. As of this writing, the fee is \$3/car if the entry fee is under \$50, or \$5/car if the event fee is higher.

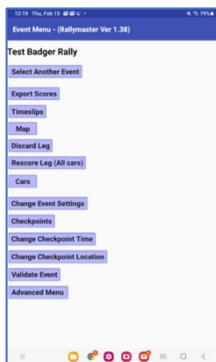


Figure 1: Event Main Menu

Rich will reply with a pointer to the event name in his database, as well as a rallymaster password unique to this event. Do not publish this password. You will have an opportunity to create a different user password for the competitors.

3.4. STEP 3: Initialize the Rallymaster Event

When you first run the event, you will receive a page asking you to agree to the privacy policy. Once this is accepted, the page shown to the left will be displayed. Choose *Select Another event*, and choose the event name Rich has assigned. Enter the password he assigned you.

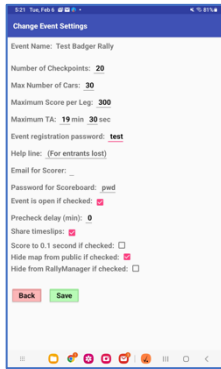


Figure 2: Event Settings

Next step is to select *Change Event Settings*. This will bring up a second screen as shown on the left. You should create an event registration password, which will be published with your general instructions, or mailed to competitors with their car numbers. Once the password has been distributed, you should leave the *Event is Open* selection unchecked except as needed for prechecks. This prevents users from registering their vehicles before the database is finalized (more on this later). Set the number of checkpoints to a large number for now (you will eventually set it to the actual number of checkpoints in your event). You should also set the max number of vehicles to a number higher than you expect to be used. This prevents headaches later where

people with high car numbers can't register the app.

Other fields can be filled in or changed based on the event requirements. Make sure that maximum score per leg matches what you published in your general instructions. I recommend that it be set to a lower value for trapless tours, perhaps 30 or 60 seconds. The scoreboard password enables competitors to access the scoreboard app, and should be left blank if you choose to share the scoreboard. The precheck delay field is only used during prechecks and should normally be set to 0. Normally the event will report errors of 1 point per second up to the max error. However, checking the score to 0.1 second box will report errors at 0.1 points per 0.1 second error. This produces higher scores for your event, but eliminates virtually all ties.

3.5. STEP 4: Lay Out Your Course

Next lay out your proposed course to get a rough idea where you are going to go. A good method is to use Google Maps for this first pass. It is fairly accurate, and has the advantage of showing some potential roads you may overlook. There are several things you need to consider that are unique to the Richta system. First, be very careful about having the route loop back on itself. A later checkpoint can be triggered early if the route passes that checkpoint before you planned it. In the same way, you need to be careful about placing checkpoints near intersections where rally cross traffic might trigger the checkpoint. For example, a checkpoint placed at a STOP sign could be triggered by a competitor going by the intersection earlier in the event. It is best to place checkpoints at least $\frac{1}{4}$ mile before the intersection. There are methods available to handle these situations if it becomes necessary. These methods are covered in Section 5.

Next, you need to consider restart locations. These locations provide an opportunity for the rallyists to catch their breath and resynch for the next batch of checkpoints. There is a necessary restart at the end of the odometer check and at the end of each rest break. Other restart points should be considered to avoid checkpoint overload. Make sure that each restart location has ample room for a number of vehicles to pull over and wait.

Now is a good time to also scout out locations to place the timing checkpoints. It is easier to find them on subsequent runs if you use hard landmarks. These can be signs, back sides of signs used by oncoming traffic, a large distinctive tree or even a field entrance. Just make sure you write them down in your notes and/or preliminary route instructions. Save actually placing them until the second run. It is also a reasonable idea to scout out some extras that can be eliminated before step 5.

3.6. STEP 5: Place GPS Controls

The course should now be run to set the checkpoint locations. Run the course as normal. At each location, you will set the location in *Rallymaster App* as follows:

1. Stop the car even with the landmark and wait 5-10 seconds for the GPS to stabilize.

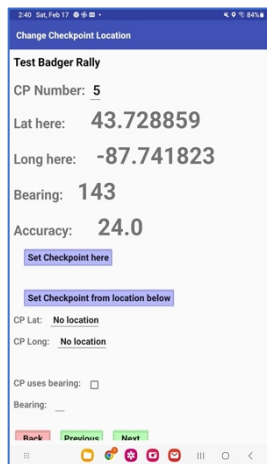


Figure 3: Change CP Location

2. From the main menu, select *Change Checkpoint Location*. You will see the screen displayed on the left.

3. Verify the checkpoint number. Note that every Richta checkpoint is a unique number, including time checks, restarts, and timing checkpoints. The typical rally uses a *time check* at registration, a *time check or Restart at Car Zero Time* at the start of the odo check, and often a *time check* at the end of the odo check. Thus, the first restart point is often checkpoint 4 and the first timed checkpoint is checkpoint 5.

4. Check the accuracy of the location which is shown on the screen. This example (from my indoor office) shows an accuracy of 24 meters. To get better accuracy, you should move the smartphone out of the vehicle and allow time to stabilize. (see section 1.4)

5. Once you are certain of the location, press the *Set Checkpoint Here* button. The app will automatically go to the next checkpoint. To verify, go back to the main menu and re-enter the *Change Checkpoint Location* screen. The app should now display the CP Lat and Long you set. Make sure that the *set bearing* box is unchecked unless needed. See Section 5 for a description of the set bearing checkpoints.

6. Proceed to the next checkpoint location and repeat.

Note: The checkpoint type is set in Section 3.8. It is located on the *Set Checkpoint Time* page.

3.7. STEP 6: Measure the Course and Validate GPS Control Locations

Once you have placed all the checkpoints you should rerun the course. This is a good time to take a first measurement as well as verifying proper checkpoint operation.

As a minimum, measurements should be taken to the hundredth of a mile, and preferably to the thousandths. If a dedicated rally odometer is unavailable, the Richta GPS based *Odometer* app will produce acceptable results.

Mileages should be reset at each restart location, and can be reset anywhere by using a route instruction Instructing you to zero your odometer (xx.xx/0.00). A good practice is to zero about every ten miles or less because contestants using a stock odometer will tend to drift off of the official mileage over long distances.

As you run the course, use the competitor app to monitor checkpoint locations. verify that all the checkpoints trigger in sequence and at the right locations. If a checkpoint is triggered early and out of sequence, you have probably either introduced a loop or the checkpoint is on a cross street or parallel road that is near enough to trigger it. The solution is to move the offending checkpoint further back or outside of the looped course area. You can also change the course to avoid the loop. If you absolutely need to use this location, there are some tools available, covered in section 5.

3.8. STEP 7: Calculate Leg Times for Each Control

Once the checkpoints have been placed and a measurement run has occurred, the leg times should be calculated for each checkpoint. Perfect time should be calculated outside of the app and entered manually. Procedures for doing this are covered in section 6. Note that the leg times must be expressed in HH:MM:SS.mmm format. Calculating and entering to at least the hundredth of a second will avoid some errors in scoring and can act as a tiebreaker. If you are timing to the tenth of a second, you must specify to at least the tenth.

Once the leg times have been calculated, they can be entered into the *Rallymaster App*. Open the *Change Checkpoint Time* screen from the *main menu* (See [Figure 4](#)). A screen will appear asking you to select a checkpoint number. Select the checkpoint number and press GO.

A screen to set checkpoint time will appear as shown in [Figure 4](#).

Figure 4: Change CP Time

For *Time check* checkpoints, flying restarts and on course/off course checkpoints, you only need to enter the checkpoint type as there is no timing data associated with this style checkpoint.

Start at Car Zero Time Checkpoints have a field for a time of day. You must enter the time in 24-hour format. For example, a CZT of 2:25 PM would be entered with 14 in the hours field, 25 in the minutes and the seconds and hundredths would be left blank. Be careful when setting time of day restarts to allow enough time for contestants to traverse the distance from the last time checkpoint plus at least 2 minutes. Break time should also have a little extra slack.

For *Timed checkpoints*, you need to enter the time you have calculated in hours, minutes, seconds and milliseconds. Note that if you enter a 1 for milliseconds, it will be 1 mS, not a tenth of a second. To choose a tenth, you need to enter 100. For *Timed from previous checkpoint* checkpoints, you would enter the calculated leg time for that checkpoint. For *Timed from Previous TOD Restart or Flying Start* checkpoints, you should enter the sum of all leg times from the previous restart to that point.

You have the option to enter a short critique message. This is displayed in the competitor app if they click on any checkpoint which has been scored. Some rallymasters display the official mileage for the leg. On a trap rally, this can contain a brief description of the trap.

Do not enter any data in the *Open or Close Time* fields. These will be covered as special cases in Section 5.

Once the timing data is correct, press the *Save and Show Next Leg* button and repeat the process.

When the entire event is entered, go to the *main menu, event settings* and set the *number of checkpoints* field to the highest number checkpoint used. Next select *Validate event*. When the number of CPs read field stops changing, press *Report*. The next screen will ask for an email address for the report. Email the report and open it. The validation report is described in Section 4.

3.9. STEP 8: Prechecks

Once all the checkpoint data has been entered, you are ready for your first precheck. The first precheck should be run by the rallymaster in the passenger seat and a driver who has not seen the course. Have the driver run the event using the route instructions at rally speeds. It is helpful to use a rally computer for this run. If access to a dedicated computer is not available, the app store has the *Richta Simple Rally Computer*. It runs on any tablet and functions much like a Timewise rally computer, using GPS for measurements.

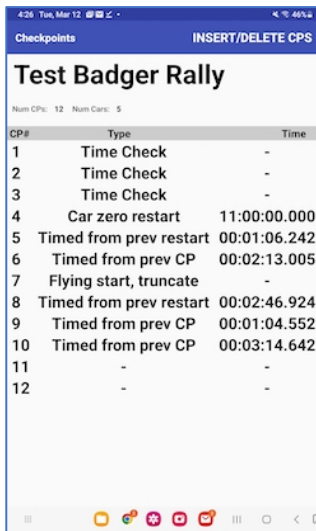
The rallymaster should verify the following items during the precheck

- The route instructions are complete and accurate. Rely on the driver to find issues.
- The speeds are appropriate for the roads.
- The checkpoints are detected in the proper order.
- The mileages are accurate.
- The leg timings are about what is expected. It is difficult to expect to remain on time during this first precheck as issues come up which need to be addressed.

3.10. STEP 9: Adjust Control Locations and Leg Times Based on Precheck

Issues will always happen during that first precheck. The assigned CAST may be too high or low. A checkpoint that looked good on paper is now in a bad location, or was triggered too early. The highway department is closing a road. Simply go back, reset checkpoint locations if necessary, make the changes to the route instructions. Then recalculate the affected legs and possibly adjust any time of day restarts.

3.11. STEP 10: More Prechecks



CP#	Type	Time
1	Time Check	-
2	Time Check	-
3	Time Check	-
4	Car zero restart	11:00:00.000
5	Timed from prev restart	00:01:06.242
6	Timed from prev CP	00:02:13.005
7	Flying start, truncate	-
8	Timed from prev restart	00:02:46.924
9	Timed from prev CP	00:01:04.552
10	Timed from prev CP	00:03:14.642
11	-	-
12	-	-

Figure 5: Checkpoints Display

Repeat prechecks as necessary. Make corrections and rerun until everything is running smoothly.

All rally routes, calculations and Richta Checkpoint locations and timing should be frozen before the official precheck. Reset the max number of checkpoints in the Richta settings menu to the highest checkpoint number shown in the checkpoints screen (available from the main menu). In the example on the left, the last checkpoint used was 10, so set *Max Checkpoints* in the *Change Event Settings* menu to 10.

Plan to have an official precheck about two weeks before the event. This precheck should be run with people who have not seen the course before. It should be run on the same day of the week and at the same time, so you can best verify traffic conditions on rally day. Any issues that are found on the official

precheck should be handled through an addendum. Do not change route instructions unless absolutely necessary.

After the official precheck is complete, you can enable the *Event is Open* by checking the field in the *Change Event Settings* screen, assign car numbers and notify drivers to download and initialize the competitor app.

Even though everything is perfect, you should still run a final precheck the day before the rally. Things happen. Roads get closed for construction. Signs get knocked down or erected. An unanticipated event, such as a bicycle race or street fair might block a portion of the route. Knowing ahead of time allows a chance to reroute and recover. It is not necessary, but is helpful to run the *Competitor app* for this final precheck.

If something changed that might affect timing at a checkpoint, it is often better to simply convert that checkpoint to a time check. Do not add or delete checkpoints as this will affect both route instructions and calculations. It also greatly increases the chances of introducing new errors that can't be prechecked. It is best to do as little as possible to address the situation.

Run the validation report again and verify that all errors and warnings have been addressed.

3.12. STEP 11: Rally Day

Car	Class	Team	Ver	Ready?
1	EXP	Smith/Smith	1.26	Not Ready
2	NOV	Fred/Ginger	1.26	Ready
3	EXP	fred/barney	1.26	GPS
4	COM	rocky/bull	-	CPs GPS
5	-	-	-	-

Figure 6: CARS Screen

Figure 7: CARS Detail Screen

3.12.1. Before Registration

Before registration opens, run the *Rallymaster App* and verify that contestants have registered the app. This is done by selecting *CARS* from the main screen. An example of the Cars screen appears in [Figure 6](#). Edit individual entries by selecting the name on the screen. A new menu will appear as shown in [Figure 7](#).

Fill in any blank names and classes as this will help with scoring later. Note that *Class* names should be consistent. They are used during scoring to sort the results by class. There is no way to enforce class names in the competitor app, so adjustments need to be made by the rallymaster app for consistency.

If registration has occurred via the competitor app, a version number will be displayed. The rallymaster should ensure all entrants are running the most current version of the app (see www.richtarally.com for the most current version).

A red background indicates an issue. A red background to CPs indicates all the CP data has not been read by the app. A red background indicates the *Competitor* app has not reported its position in the last hour. Red here may indicate the phone's GPS and location services are not working. Not Ready indicates that the competitor has not triggered at least one checkpoint.

Be sure to change *Max Cars* in the *Event Settings* menu to the highest registered car number.

3.12.2. During the Event

The rallymaster should remain free to run a lead car as a final verification of roads, signs, checkpoints and any on course workers. Event monitoring should be handled by a dedicated person or group as described in Section 7.

3.12.3. At the Finish Location

The rallymaster should be the primary contact with the competitors. Decisions to rescore or discard a leg, or to form a protest committee should be performed by the rallymaster.

Scoring should be performed by the dedicated group and is described in Section 7. This group handles posting scores, handling of any missed checkpoints, leg timing errors and or protests.

3.13. STEP 12: Post Event Reporting and Accounting

After the event concludes, the club should contact Rich Bireta with the number of vehicles that actually started the event, and ask for an invoice. Please pay promptly when received.

The final validation report should also be archived with the scoreboard and other rally materials that get saved.

4. Validation Report

```

Event name: Test Badger Rally

The maximum score on each leg is 300.

The event has 20 checkpoints.

3 checkpoint documents were read.

ERROR: The event setting for number of checkpoints does not match
the number of checkpoints read.

The event is timed to the second

There is no precheck offset

The maximum time allowance is 19 minutes 30 seconds.

The maximum number of cars is 30

Checkpoint 1:
Checkpoint is a time check.
Cars are timed as they pass.
All competitors receive a score of 0 at this checkpoint.
Location (lat/long) is 43.729005 -87.741875
No open time set.
No close time set.
Critique message is: test
Checkpoint is not sensitive to the direction in which it is passed.
Warning: Distance between CP 1 and CP 2 is 0.0 meters.
Warning: Distance between CP 1 and CP 3 is 10.3 meters.

Checkpoint 2:
Checkpoint is a time check.
Cars are timed as they pass.
All competitors receive a score of 0 at this checkpoint.
Location (lat/long) is 43.729005 -87.741875
No open time set.
No close time set.
Critique message is: null
Checkpoint is not sensitive to the direction in which it is passed.
Warning: Distance between CP 2 and CP 1 is 0.0 meters.
Warning: Distance between CP 2 and CP 3 is 10.3 meters.

Checkpoint 3:
Checkpoint is a time check.
Cars are timed as they pass.
All competitors receive a score of 0 at this checkpoint.
Location (lat/long) is 43.728926 -87.741940
No open time set.
No close time set.
Critique message is: null
Checkpoint is not sensitive to the direction in which it is passed.
Warning: Distance between CP 3 and CP 1 is 10.3 meters.
Warning: Distance between CP 3 and CP 2 is 10.3 meters.

End of report.
6 warnings.
1 errors.

```

Figure 8: Validation Report

The validation report provides a sanity check on the values entered for the event. It also provides a record of the event for archiving by recording all checkpoint locations and timing information. A validation report is generated by selecting *Validate event* from the main menu. After waiting for the display to stabilize, press *report*. The app will ask for an email address. Enter a valid email and press *send*. A text file will be sent via email.

An example of a validation report is shown on the left in [Figure 8](#).

The example to the left indicates one error and a number of warnings. The error is that the number of checkpoints placed (3) does not equal the number of checkpoints defined (20). This is a common error for new users of *Rallymaster* and will prevent proper operation of the *competitor* apps.

The warnings are due to the fact that checkpoints were defined too close together. (I created this set of checkpoints by walking around in my house, so no surprise)

Another error could be that a Time of Day restart has an earlier time specified than for a previous restart. There are other errors and warnings available, but they are generally self-explanatory.

The archive record is the settings for the event at the top including name, number of checkpoints, timing info, precheck offset and maximum number of cars.

The archive record for each checkpoint shows settings for checkpoint type, how cars are scored, location (latitude, longitude), opening and closing times, the critique message, and potentially a bearing that the cars must approach from to enter the checkpoint.

Some items do not contain warnings, but should be verified before the event is run. If the Open time or bearing has been set, check carefully to ensure they are intended to be used. These settings are uncommon and only used in special circumstances.

All errors must be resolved. Warnings should be resolved or understood.

5. Use of Open/Close Time and Bearing Settings

Each checkpoint has the ability of specifying an opening time, a closing time and a bearing of approach. They are intended to be used in special cases where a rally course must overlap. These settings can be used to prevent premature triggering of a checkpoint arrival. It is recommended to not use these settings unless needed as unintentional errors can occur.

The open and close time fields are accessed from the *Change Checkpoint Time* screen (see [Figure 4](#)). The fields should be entered in 24 hour format HH:MM (i.e. 14:30). Vehicles reaching this checkpoint before the specified opening time will not be detected, but will be detected when re-encountered after the specified time. The close time field works in the opposite manner and vehicles will not be detected after the specified close time. The two fields can be used together to provide a timing window. The close time field is very uncommon and it is difficult to define where it would be used on a normal TSD event.

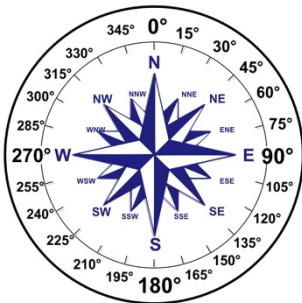


Figure 9 Compass Bearing

The bearing field is accessed from the *Set Checkpoint Location* Screen. The bearing setting specifies the compass bearing that a competitor would approach the checkpoint from. Thus, a checkpoint where competitors are traveling west would have a bearing of 270 degrees. A graphic of bearings is shown in [Figure 9](#). When setting or updating a checkpoint location a suggested bearing will be displayed. To use this bearing simply check the *CP Uses bearing* checkbox as shown in [Figure 3](#) and enter the bearing. The competitor app has a fairly wide window for accepting a bearing (+/- 45 degrees), but it should be set as close as possible.

Cars approaching from the other direction, or passing a checkpoint from a different heading (i.e. cross street) would not trigger checkpoint detection. This is one way of avoiding early detection, but should be used as a last resort.

A common source of error is having the *CP Uses Bearing* setting checked with the default bearing of 0. This error will not be flagged in the validation report, so care should be taken during manual review.

6. Timing Calculations

Rally calculations are performed using the basic formula

$$Time = \left(\frac{60}{CAST} \right) * Distance$$

Where:

- Time is in the format MM.mmm (Minutes and fractions of a minute)
- 60 is the number of minutes in an hour
- CAST is in Miles per Hour. CAST stands for Change Average Speed To and is the assigned speed for this calculation.
- Distance is measured in miles.hundredths


The units work as follows

$$Minutes = \frac{\frac{Minutes}{Hour}}{\frac{Miles}{Hour}} * Miles$$

The miles and the hours cancel, leaving minutes for the unit.

This calculation is performed for every piece of the leg. CAST changes introduce a new calculation, and pauses must be added. Finally, the hundredths of a minute must be converted to seconds and fractions of a second.

Example: Using the following Route Instructions

NRI	Off Miles	CZT/ CAST/ Pause/ CP	INSTRUCTION
1.	/0.00	CP 7 CAST 30	STOP. RESTART AT THE TOP OF ANY MINUTE
2.	0.933	PAUSE 15	STRAIGHT AT STOP.
3.	1.438	CAST 40	STRAIGHT AT STOP ON NEUHEISEL.
4.	2.144	CP 8	
5.	3.211		RIGHT ON HAZELNUT.
6.	3.500	CAST 42	
7.	4.301	CP 9	“STOP AHEAD”

Calculate CP 8

NRI 1-3 (CAST Change)	$(60/30)*1.438$	2.876
NRI 2	$Pause\ 15 = 15*(1/60)$	0.25
NRI 3-4 (CP 8)	$(60/40)*(2.144 - 1.438)$	1.059

Leg time CP 8 is $2.876 + 0.25 + 1.059 = 4.185$

Convert fractional minutes into seconds and add integer minutes

$$4:00 + 0.185(\text{frac minutes}) * \left(60 \left(\frac{\text{sec}}{\text{fract}} \text{minutes} \right) \right) = 04:11.100 \text{ MM:SS.mmm.}$$

Next CP 9

NRI 4-6 (CAST Change)	$(60/40)*(3.500 - 2.144)$	2.034
NRI 6-7 (CP 9)	$(60/42)*(4.301-3.500)$	1.144

Leg time CP 9 is $2.034 + 1.144 = 3.178$

Convert to MM :SS.mmm $3:00 + (0.178 * 60) = 3:10.680$

If using the *scoring from last Checkpoint*, the calculated values would be entered in the change checkpoint time for these checkpoints.

If using the *scoring from last restart*, checkpoint 8 would remain at 04:11.100, but the time entered for checkpoint 9 would be the sum of the legs since the restart, or

$$4:11.100 + 3:10.680 = 7:21.780$$

It is advisable to use a spreadsheet for the calculations. The reason is that a spreadsheet can easily be updated when changes are made during prechecks or when scoring errors are found during the event.

7. Duties of the Monitoring and Scoring Staff

It is strongly recommended that the rally committee allocate resources for a dedicated person or group for real time monitoring of the app and performing scoring at the finish. Doing this frees the rallymaster to manage the flow of the event, run lead, and address problems as they occur. Duties of this person/group include:

- Helping competitors initiate the app at registration.
- Monitor and track down competitors who have not properly initialized the app.
- Monitor the map and checkpoints for competitor progress through checkpoints and early detection where an issue has halted the event (remote sweep).
- Monitor the map to detect off course vehicles and potentially contact them to get back on course.
- Act as a central point of contact for competitors (breakdowns, off course, abandoning the rally, etc.), freeing the rallymaster from dealing with minor issues.
- Coordinate as necessary with the rallymaster via phone to address major issues.
- Manage the scoring during the event by periodically downloading the scores via the *Export Scores* function on the main menu. Executing this function will prompt for an email address and will send an enclosed spreadsheet in CSV format of the scores reported by the competitor apps.
- If a majority of scores are unexpectedly high for a particular leg, examine the calculations for a potential scoring error. If found, perform the following
 - Coordinate with the rallymaster and calculate a new official leg time.
 - Enter the new official time for the checkpoint using the *Change Checkpoint Time* function.
 - Wait until all cars are at the finish location, then use the *rescore leg (all cars)* function to rescore the leg. If this is done early, some competitors may not properly update to the new time, and this will need to be repeated.
 - If the timing issue cannot be resolved, use the *discard leg* function to remove it from scoring (converts all scores to zero for that leg)
- Post scores and allow for comments.
- Address competitor issues such as a leg not being detected, resulting in a bad score. Cars that drop out early will show an artificial low score as the app will show a score of 0 for the missed checkpoints. Deal with the issue according to the event generals by either deleting the cars from scoring or converting the missing scores to a max penalty. This can easily be done by selecting the *Missing Scores given Max* checkbox in the *Export Scores* function

7.1. Making Adjustments During the Rally

Sometimes it is necessary to make adjustments to checkpoints during the event. It may turn out that a leg was incorrectly timed and needs to be adjusted. Sometimes an error was made that cannot be recovered, and the leg needs to be discarded. This section addresses both cases.

7.1.1. Rescoring a Leg

The new leg should be rescored outside of the *Rallymaster* program, using the new data applied to the original scoring calculations. Examples include a missing or extra pause, or a CAST change applied at an incorrect mileage.

In this case recalculate the leg time. Use the *Change Checkpoint Time* function in the main menu to update the time in the database. Once you are certain that all cars have cleared that checkpoint, select *Rescore Leg (All Cars)*. It is safest to do this post event. This will update scores for all the competitors as well as for the *export scores* function for scoring.

7.1.2. Discarding a Leg

Once it has been determined that the leg needs to be discarded (rallymaster agrees), use the *Discard Leg* function in the *advanced menu*. This will convert all competitor leg scores to 0 for that leg. Again, this should be done after all competitors have cleared that checkpoint. Discarding a leg may be undone by rescoring the leg (see section 10).

8. Using the Map Capability in Rallymaster

Rallymaster has the capability to display a real time map during the event. You can select the *map* display from the main menu. The map that appears uses Google Maps with overlays for checkpoint positions and appears as in the example in [Figure 10](#):

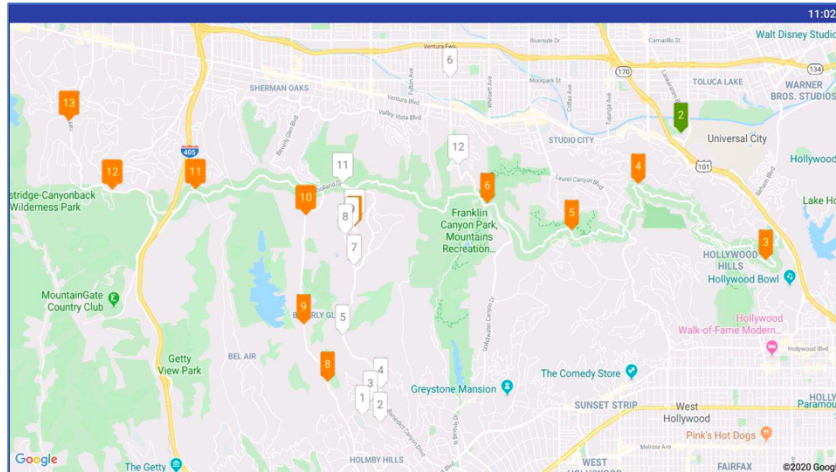


Figure 10: Map Display

The map can be zoomed and moved as the rally progresses to get the most accurate snapshot. Time Check locations appear in blue, restart locations appear in green, timed checkpoints appear in orange, and the rally vehicles appear in white. The map will update every couple of seconds to show where the competitors are on the course. In the example above, the rallymaster can see that car 6 and possibly car 12 are off course. This information can be relayed to the sweep car, or if mobile phone numbers were entered for the contestant, they can be called directly and be directed back on course.

Cell coverage dead zones can affect individual vehicles from updating their position. The car will appear to be frozen in one spot until coverage is reestablished, and will then jump to the new location. This is normal and commonly observed. Dead zones may occur for different vehicles in different places depending on cell carrier, type of phone, placement in car, etc.

9. Use of the Scoreboard Function



Scoreboard - Ver 4
17:26:07
Son of Sno*Drift XXII
Num CPs: 85 Num Cars: 26 CPs Read: 2046

Car	Cls	Team	CPs	Score
1	E	Jeff & Nic Boris	85	33.0
2	E	Layton/Layton	85	264.9
3	E	Harkcom/Jean-Franco	85	94.7
4	E	Ken & Dennis Wiedbu	85	384.1
5	E	Kay/Bennett	85	34.0
6	E	Fishbeck/Dawson	85	38.0
7	L	Harkcom/Lester	85	75.2
8	L	Line/Line	85	279.4
9	L	Adams/Piotr	85	87.4
10	L	Steve & Sean Riddell	85	171.5
11	L	Oxendine/Swarm	65	184.5
12	-	-	-	-
13	N	Everett/Seeger	85	317.5
14	N	Harwood/Dorella	85	805.0
15	N	Velarde/Romero	85	140.5
16	N	Rotsaert/Zimmerman	85	194.1

Change Event 111

The *Scoreboard App*, available as a separate download (see Section 1.2) can also be used to track contestant progress on rally day. A rallymaster can look at each entry and instantly see the number of checkpoints that have been visited. The number includes all checkpoints, including timed checkpoints, restarts and time checks. You also have a display with the max number of checkpoints and total number of vehicles. The display will scroll down to see the higher car numbers. Note that in the example to the left, all vehicles that are visible have reached all 85 checkpoints, except for car 11. That car is either running far behind or has dropped out. The rallymaster needs to monitor this situation to determine if he can declare the scores complete.

Figure 11: Scoreboard

A more detailed set of leg scores is available through the export scores function in the main menu. This function exports a spreadsheet of competitor leg and total scores via an email link and can be used to detect issues in scoring. The process of creating a scoring spreadsheet from this data is shown in Section 11.

10. Advanced Menu

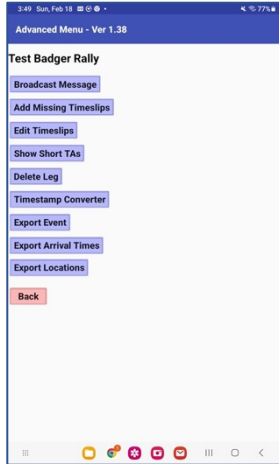


Figure 12: Advanced Menu

The *Advanced menu* provides access to less commonly used functions. The menu is shown in [Figure 12](#).

The following functions are available:

- **Broadcast Message:** Allows entry of a message which is then immediately sent to all competitors. This generally only happens when something drastic occurs that prevents competitors from completing the course.
- **Add Missing timeslips:** If a competitor is using a backup device and the primary competitor device is missing or shows an incorrect value for a checkpoint, this function will copy the data from the backup device.
- **Edit Timeslips:** This provides a method of correcting an individual leg score for one competitor. Select a car number and a checkpoint number. Select *GO*. You can then edit the cars timeslip to change the score. When you are done, select *Rescore*, then *Save*. Press *Back*. You can then select *Send Timeslips* so the corrected score appears on the contestants competitor app. Edit Timeslips can be used to add a missing timeslip (as in the case of missing a restart CP). The calculated score can also be overridden before saving it.
- **Show Short TAs:** This will display all the 10 and 20 second TAs taken by all competitors.
- **Delete Leg:** At the rallymaster option, this function will cause a specified leg to be discarded. It will still show up in the scoring, but all competitors will show a penalty of 0 for the leg. (Note: This function may be removed in future releases. Legs can be deleted by editing the *Checkpoint* screen).
- **Timestamp Converter:** a very rarely used utility to convert the internal time formula entry to a readable Time of Day. This is only used when it is necessary to decode entries from a database.
- **Export Event:** This function will export ALL data stored in the database and email it in spreadsheet format (csv). It is used to backup the entire event.
- **Export Arrival Times:** This will export via email a spreadsheet showing the arrival times for all competitors at each checkpoint. Occasionally useful in decoding timing issues.
- **Export Locations:** This will export via email a spreadsheet showing the GPS coordinates of all checkpoints. Useful in verifying checkpoint locations against the rallymaster notes and when a checkpoint doesn't trigger as expected during a precheck.

11. Exporting and Displaying Scores

The *Rallymaster App* has the capability to generate a scoring file suitable for display at the finish. The basic procedure is to transfer the raw score file to a spreadsheet (often Excel) and then manipulate the data to display the scores.

Step 1 – Export the scores

This is done by using the *export scores* function available in the *main menu*.

- Wait for *CPs Read* and *Timeslips read* fields to stop changing
- If all cars have finished or dropped out, select *Missing Scores given max*
- Press *Report*
- A Dialog will appear to generate an email. Enter a valid email address and send the email

Step 2 - Receive the scores

An email will be received containing an attached spreadsheet as a comma separated values (.csv) file. Save this file to the scoring computer.

Open the score file using a spreadsheet program. It should look something like this:

Car	Team	Class	Total	4	5	6	7	9
1	SmithSmith	COMPUTER	4.3	0.8	1	0.6	0.4	1.5
2	HarryTonto	NOVICE	69.4	8.6	5.5	30	12.1	13.2
3	ThelmaLouise	EXPERT	78.8	12.5	30	11.3	9.5	15.5
4	BurnsAllen	COMPUTER	5.2	1.2	0.4	0.8	0.7	2.1
5	RockyBull	EXPERT	37.2	3.7	6.6	5.9	11.4	9.6
6	FredBarney	NOVICE	110.2	14.5	18.4	22.2	30	25.1

Figure 13: Example Scoresheet

Note: The above example is for a fake event with 6 entries, 3 classes and 5 timed checkpoints. CP1 and 2 were timechecks. CP3 and 8 were restarts. These are not exported for the score file.

The heading row shows the CP number for each TIMED Checkpoint and the total shows the sum of the leg scores.

Step 3 – Sort and Publish

Use the capabilities of the spreadsheet program to sort by *class* and then by *total score*. This should generate the following final results sheet that can be displayed and published:

Car	Team	Class	Total	4	5	6	7	9
1	SmithSmith	COMPUTER	4.3	0.8	1	0.6	0.4	1.5
4	BurnsAllen	COMPUTER	5.2	1.2	0.4	0.8	0.7	2.1
5	RockyBull	EXPERT	37.2	3.7	6.6	5.9	11.4	9.6
3	ThelmaLouise	EXPERT	78.8	12.5	30	11.3	9.5	15.5
2	HarryTonto	NOVICE	69.4	8.6	5.5	30	12.1	13.2
6	FredBarney	NOVICE	110.2	14.5	18.4	22.2	30	25.1

Figure 14: Sorted Scoresheet

12. Technical Details on Data Transfer Between the Competitor and Rallymaster App

All Rally data is stored in a central database hosted by Google Cloud Services. The database used is a No SQL database called Google Cloud Firestore.. The database contains all the event general info as well the checkpoint times and locations. It also contains all the competitor info entered from the competitor app, such as names, class and reporting info.

The rally info is sent to the database as it is being created in the *Rallymaster app*. Therefore a record would be sent to the database every time a checkpoint location or time is entered or changed.

When the competitor initializes the *Competitor app* the demographic data is sent. It would also be resent any time info on the user config page is saved.

It is important to note that all location and timing data is downloaded to the competitor app when the app is started and the proper event is selected. It is NOT updated during the event. This is important because and changes you make to checkpoint timing or location will not be captured to any running competitor app. To update, the app should be halted and restarted.

When a checkpoint is entered, the competitor app will send the timing data to the database. If cell coverage is unavailable, the data is held and transmitted when coverage is available. However, the competitor receives his score and any critique when entering from the local stored data.

Rescoring a leg and discarding a leg should be performed after all competitors have completed the course. It can be done early with no damage, but must be done again after everyone has arrived.

13. Use of Backup Devices in the Competitor App

A convention has been established for competitors choosing to use a backup device running the competitor app. Doing so provides a duplicate record of checkpoint entry data and can be used to resolve scoring disputes. A competitor wishing to run a second *Competitor app* should register the backup device as normal, but use car number plus 100 as the car number. Thus car number 6 would use 106 for the backup device.

Worker cars such as lead and sweep can use any number higher than max cars. We often use 61 for lead and 62 for sweep.