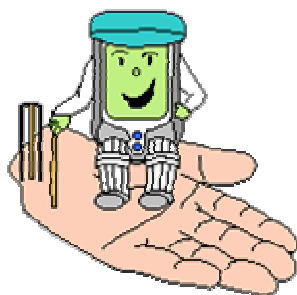


Palmericket User Manual



Version 1.50

HTML Help File

February 2015

Table of Contents

1. [Introduction](#)
 2. [Features](#)
 3. [Initial Installation](#)
 4. [Installing Upgrades](#)
 5. [Starting and Running Palmericket](#)
 6. [The main Screen and scoring Overview](#)
 7. [Registration](#)
 8. [Starting a New Match](#)
 9. [Notes on Match and Team Details](#)
 10. [Clocks](#)
 11. [Batter on Strike](#)
 12. [Scoring Normal Runs](#)
 13. [Confirmation Screens](#)
 14. [Batting Accolades](#)
 15. [Correcting Scoring Errors](#)
 16. [Scoring Extras](#)
 17. [Recording Dismissals](#)
 18. [Ending an Over](#)
 19. [Mid-over Bowling Changes](#)
 20. [Correcting Bowler Selection Errors](#)
 21. [Breaks in Play](#)
 22. [Ball-by-ball Edits](#)
 23. [On-line Help](#)
 24. [Batting Summaries](#)
 25. [Time Spent at the Crease](#)
 26. [Bowling Summaries](#)
 27. [Innings Timing](#)
 28. [Utilities](#)
 29. [Match Completion](#)
 30. [Match Result and Notes](#)
 31. [Conclusion](#)
-

• [Introduction](#)

A Palm™ handheld computer represents a perfect solution for scoring cricket matches. These wonderful devices are inexpensive, light and durable, fit in one's shirt pocket and have a battery life of weeks, if not months. Palm handheld computers come with abundant memory, sufficient to store a large number of cricket matches, with enough memory left over for normal day-to-day diary and

appointment functions. The Palm operating system (PalmOS) is "event-driven", i.e. the Palm handheld executes instructions by responding immediately to stylus taps and button presses initiated by the user.

If one thinks about it, cricket is also an "event-driven" sport, comprising many different events: the toss, the start of the match, a ball being bowled, a wide ball, a batter being dismissed, the end of an over, a drinks break, a declaration, a change of bowling, etc. In fact, the PalmOS is so tailor-made for scoring cricket matches it almost seems that it was designed specially just for cricket!

Although a few cricket scoring programs have already emerged in the market, Palmcricket is the first serious attempt to utilise the PalmOS capabilities to the full. It caters for all conventional cricket scoring requirements, from 5-day test matches and one-day internationals, to week-end club matches and junior cricket played on a Saturday morning.

So pack away those bulky, expensive, power-dependent lap-top computers and discard the old archaic paper score-sheets. Change to a Palm handheld with Palmcricket and immediately start scoring cricket matches with ease, anywhere, anytime!

- **Features**

Palmcricket is a cricket scoring program that is packed with a rich selection of scoring functions and analyses capabilities:

- **Specially designed for Palm™ handheld computers:**

Palmcricket works on any Palm™ or equivalent handheld computer with 300k free memory running on Palm OS® 3.5 or later. The program is ideal for cricket coaches, scorers and umpires who need true mobility on cricket fields without access to electrical power. With Palmcricket, one can easily score while umpiring (a real advantage for overworked school and club coaches who have to do everything during a cricket match). Cricket enthusiasts can also score international one-day matches and test cricket from the comfort of their stadium seats or home lounge chairs. Professional and media scorers at international matches can now also benefit by replacing bulky and expensive PC scoring systems with Palm handheld computers running Palmcricket.

- **Score all types of conventional cricket matches:**

Use Palmcricket to score 5-day tests, one-day internationals, club and school matches on any Palm handheld. The program allows for up to two innings per team, per match, with unlimited overs per innings. It handles all possible events covered by the latest 2000 cricket laws, including allowance for batting and fielding penalties. It also caters easily for different bye-laws and interpretations.

- **Simple operation, with one-tap scoring:**

No irritating confirmation screens and multiple buttons after every event, just clean and effective scoring, with the minimum number of taps. The speed and simplicity allow one to score while umpiring. Captains and players love having such up-to-date analyses available right at the stumps!

- **Ball-by-ball database:**

Palmcricket automatically stores every scoring event to the built-in database, in real time. There is thus no need to "back up" the match after every ball or over, in case of mistakes. Effortlessly change to other Palm applications, and return to the exact match position at any time, even change batteries during a match. Mistakes are easily corrected either immediately using a backspace button, or later using the comprehensive ball-by-ball edit facility. Correct

any ball in any innings, at any time.

- **Print score sheets at the match, using infra-red:**

No need to hurry back to the office to transfer the match data for printing from one's desk-top. Built-in support for Stevens Creek Software's print server allows one to print score-sheets to portable battery-driven printers such as the Hewlett Packard™ Deskjet 350, anywhere, any time. This works especially well at isolated venues without electrical power.

- **Handle all stoppages effortlessly:**

Drinks, lunch, tea, close of play, declarations etc are all catered for and recorded in the database, with special allowance for early breaks after fall of a wicket. Palmcricket also accommodates mid-over bowling changes easily, with bowler partial overs recorded correctly.

- **Archive completed matches:**

Store up to 10 completed matches directly on the Palm handheld, to restore/view/delete when needed. Achieve complete mobility by going on tour with one's Palm handheld, and scoring all the tour matches, without coming near a desktop pc. Transfer data to the desktop at one's leisure.

- **Beaming capabilities:**

Beam team sheets and archived matches to other handheld devices using infra-red. This allows easy interchange of data between scorers before and after a match, right on the field of play.

- **Tight, fast, effective coding:**

Palmcricket was developed using the gnu open source C compiler for native code interface to Palm OS. It needs no bloated run-time modules, leaving more space for matches. The complete program occupies only 200k. Download the software and updates in minutes, no huge Mbyte overhead.

- **Initial Installation**

To install Palmcricket on the hand-held, do the following in sequence:

- Download the latest revision from the [Palmcricket web-site](#) and unzip the file into the folder **C:\Program Files\Palm\Add-on** or equivalent. On older installations, this folder may be named **C:\Palm\Add-on**. On more recent systems, it could be **C:\Program Files\PalmOne\Add-on** (Tungsten E2 models).
- Use the Install (or Quick Install) button on the Palm Desktop to add the file:

Pcri.prc

- The installation package includes team sheets for the 14 teams that competed in the 2003 Cricket World Cup. Adding some or all of the team sheet files listed below could be useful for familiarising oneself with team sheet procedures:

Australia.pdb
Bangladesh.pdb
Canada.pdb
England.pdb
India.pdb
Kenya.pdb

Namibia.pdb
New Zealand.pdb
Pakistan.pdb
South Africa.pdb
Sri Lanka.pdb
The Netherlands.pdb
West Indies.pdb
Zimbabwe.pdb

- Initiate the HotSync(R) function.

No run time modules are required, and Palmcricket will create its own score-sheet database (PcriDB.PDB). The Palmcricket module provides complete scoring functionality, with the exception of infra-red printing. The following are supplementary but not essential for correct operation:

- **SCSPrint.prc:** A print server third-party add-on available from Stevens Creek Software ([click here to link to their download pages](#)). It allows one to print score sheets directly from the Palm handheld via infra-red interface to a printer. We've used the Palm M100 successfully in the field with the HP Deskjet 350, and a desktop HP Laserjet 5P, but haven't tested any other printers as yet. This feature allows one to score and print matches at venues without any electrical power whatsoever, if one has access to portable battery driven printers such as the 350.
- **Palmcricket Manual:** The complete Palmcricket user documentation (this file, along with 109 images in GIF format, [download from here](#)), which is recommended reading for new users.
- **Palmcricket Viewer:** The free Windows companion to Palmcricket which allows one to view and/or print completed score-sheets in conventional or summarised format, or export to Excel, on desktop computers running Windows 95 or higher. [Click here to go to the Palmcricket web-site.](#)

- **[Installing Upgrades](#)**

Upgrades and improvements are made to Palmcricket continually. Whenever a new revision is available it can be [downloaded](#) from the Palmcricket website and installed in exactly the same manner as described above under [initial installation](#).

Note: Deletion of the Palmcricket application by using the PalmOs menus before installing an upgrade is **not recommended**, as the current match, archived matches and registration details will all be deleted along with the program. By simply installing only the file **Pcri.prc**, all data will be retained on the handheld, with the exception of the old program module, which PalmOs will automatically replace with the new.

- **[Starting and Running Palmcricket](#)**

Palmcricket is launched from the Palm handheld's home screen by tapping on the Pcricket icon, shown circled below:



Every time Palmcricket is started, the current score sheet is opened and re-built to the exact point where it was left previously. Although Palmcricket has built-in archiving and restoring facilities, there is no need to be concerned about exiting and re-entering the application during a scoring session. No information will be lost, and there is no need for intermediate back-ups.

• The Main Screen and Scoring Overview

When Palmcricket is launched by tapping on it's icon, the main scoring screen shown here is displayed. It looks very complicated, but one soon gets used to it. The various components are described below:

In progress		11:51:19	11:01:23	ⓘ
Australia		134/0		
1	☐	Justin Langer	56	
2	☒	Matthew Hayden	70	
		1 w 4 n 1 b 2 L Ext	8	
Ov		32.6 @ 4.06 pf 0 pb 0		
South Africa		277/10		
5	☐	Cloud Henderson	2.6-11/0	
		1.L1.n1.w1.b1..4.. = 7+ 2		
▪	1	2	3	Match Edit ? Help
4	5	6	>6	Team1 Team2 Bat
w	n	b	L	Over Bowler Bowl
←	n+	pb	pf	Out Stop Utils

Match status: At the very top, the screen title displays the match status, in this case "In progress". Various messages indicating breaks in play, end of an innings, match completion, etc. will be displayed here to show the current status of the match. Buttons and events will behave differently, depending on the match status. The clock timer on the left in the title line displays elapsed time for the match, day, innings or session, as per preference. In this case, the "i" indicates the elapsed time for the innings is 1 hour, 51 minutes and 19 seconds. The clock timer on the right displays the actual time of day, i.e. the time is just after 11 o'clock in the morning. Clocks are updated every second as the match progresses (but not on older handheld models that do not support sophisticated clock handling). On the extreme top right, the small information icon ⓘ can be tapped to display an "about" screen that displays website/e-mail addresses, a copyright message and the program version and date.

Batting team: Immediately below the title, the screen displays 4 lines of batting details for the team currently batting. The first line, in bold type, shows the team name (e.g. Australia) and total score (134 runs, 0 wickets). The next two lines show the the batting order, names and scores of the two batters currently at the crease (Justin Langer on 56 and Matthew Hayden on 70). The batter on strike (Hayden) is indicated by a tick in the block to the left of the batter. The last line of batting detail gives a breakdown of total wides (1), no-balls (4), byes (1) and leg-byes (2), with the total extras (8) shown on the very right.

Batting information: A detailed ball-by-ball analysis of the current batting performances can be obtained immediately by tapping on the numbered batting order rectangle to the left of either of the batter's names. This will bring up the batting information window shown under [batting summaries](#)

Changing the strike: The batter on strike can be changed by tapping anywhere inside the

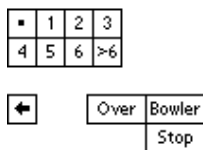
batter's name or tick-box. Pay careful attention to which batter is on strike. Normally, the strike would change intelligently, except in the case of no-balls, wides and dismissals, where the scorer may need to override the setting by tapping on the correct batter's name.

Completed overs: The next line shows the total overs completed (32.6), the run rate per over (4.06), penalty runs awarded to the fielding side (pf) and penalty runs (pb) awarded to the batting side.

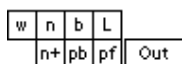
Bowling detail: The middle part of the screen shows bowling statistics for the team currently bowling. The team name (e.g. South Africa) is displayed in bold type-face. Batting scores for all innings completed previously by this team are also shown on the same line (277/10). The next line shows the number, name and bowling analysis for the current bowler (5. Claud Henderson 2.6 overs 11 runs for 0 wickets). The next line shows the current over in progress. To the left of the equal sign the ball-by-ball breakdown for the over is shown (1.L1.n1.w1.b1.o.4.o). The total runs conceded by the bowler (7) is displayed between the equal sign and the plus sign, whereas other runs not credited to the bowler, i.e. byes and leg-byes, (2 in total) are displayed to the right of the plus sign.

Bowler information: Details of the current bowler's statistics can be obtained immediately by tapping on the numbered bowling order rectangle to the left of the bowler's names. This will bring up the bowling information window shown under [bowling summaries](#)

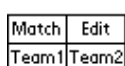
Buttons: The 29 buttons at the bottom of the screen can be divided into 4 main groups: scoring buttons, shift buttons, edit buttons and miscellaneous buttons, as described below:



Scoring buttons: The numbered buttons shown left are the active scoring buttons. One tap on any of these buttons will immediately credit the runs to the batter currently on strike, shown by a tick next to the batter's name. See [scoring normal runs](#) for more detail. The left arrow, also referred to as the back-space or undo button, shown at the bottom left of the main screen, allows one to delete the last event recorded. It deletes all traces of the event, so normally one would use this to correct mistakes immediately after tapping the wrong button. There is no re-do facility, any events thus erased can be re-instated by normal scoring. All events in the ball-by-ball database can be erased backwards, right up to the start of the match, if desired (see also [correcting errors](#)). The [Over](#) button is used to indicate the end of a normal over, whereas the [Bowler](#) button allows one to make a mid-over bowling change and to [Correct a wrong bowler](#) selected at the start of the over, without having to re-score the over. [Breaks in play](#) can be started and completed at any time by tapping on the **Stop** button. During stoppages, the label of this button changes to **Start**. This button is also used to complete an innings and start a new one. One can score up to 4 innings in a match. There is no restriction on number of overs (within reason, i.e. it can handle all 4 innings of a test match comfortably). Equally, Palmercricet does not restrict which team bats in which innings, as it will comfortably allow the same team to bat in successive or all innings, or any combination of team and innings.



Shift Buttons: These buttons all operate as shift keys: on their own they do nothing, except to supply a different context to the scoring buttons. When any of these buttons are "on", they will be highlighted. Tapping one of the run buttons will then trigger the highlighted event, eg with the **w** button highlighted, a tap on the **1** button will record one wide. Similarly, when the **Out** button is on, a tap on any run button (**dot**, **1**, **2**, **3**, **4**, **5**, **6** or **>6**) will trigger a dismissal event and the **Out** window will pop-up. The **Out** button can be on simultaneously with any of the extras buttons, in order to trigger dismissals with extras, whereas only one extras button can be on at a time. If the shift button was tapped by mistake, simply tap it again to remove the shift effect. Refer to the sections on [extras](#) and [dismissals](#) for more detailed descriptions of these functions.



Edit buttons: The Match, Team1 and Team2 buttons are normally used at the start of the match to edit the match and player details. They may also

be used at any time during the match, to save a team sheet, correct player names etc. (see [step 4](#) under the section [starting](#) a new match). The **Edit** button allows one to correct minor scoring mistakes anywhere in the match (see [ball-by-ball editing](#)).

? Help
Bat
Bowl
Utils

Miscellaneous buttons: The buttons on the right of the scoring block are used for various queries, analyses and non-scoring activities. At the top, are two [Help](#) buttons. The [Bat](#) button displays battings summaries for any innings, whereas the [Bowl](#) button provides bowling statistics. A number of additional utilities are accessed by tapping on the [Utils](#) button. These include printing, registration, archiving, cumulative graphs, Manhattan skylines and target settings.

- [Registration](#)

As from version 1.50, Palmcricket is available free of charge and registration of a Palmcricket licence is no longer necessary. Although free, P-Viewer has not been placed in the public domain. Copyright remains vested with J deV Oberholzer and Multi-Media Publications (Pty) Ltd.

- [Starting a new match](#)

Directly after installation, the current score-sheet is initialised and stored in the default Palmcricket database, (PCriDB.PDB). At this stage, the match status will be displayed as **Not Started**. Here one can skip steps 1 to 3 and proceed directly with step 4 below.

Once a match has been completed, the current match must be reset, using the following steps 1 to 4:

Step 1: Complete the current match and record the result, as per the sequences described in [Match Completion](#) and [Match Result and Notes](#).

Step 2: Optionally save the match, using the **Utils** button, as described in [Utilities](#).

Step 3: Tap on the **Start** button to reset the match status to **Not started**. Since the current match will now be erased completely, the program will ask one twice to confirm this action, as shown. Tap the **Restart** button on the two successive confirmation screens. The current match will be erased, and the match status will be set to **Not started**.

Match Completed ⓘ South Africa 5/255 ☒ 3. Jacques Kallis 108 ☐ 7. Lance Klusener 26 Restart the Match ⚠ Restart is irreversible! You should archive the current match first. All current scoring will be lost without a trace. Are you sure? Cancel Restart	Match Completed ⓘ South Africa 5/250 ☐ 3. Jacques Kallis 104 ☒ 7. Lance Klusener 25 6 w 5 n 0 b 1 L Ext 12 Restart the Match ⚠ Last chance to change your mind! All current scoring will be lost without a trace. Are you sure? Cancel Restart	Match not started ⓘ Australia 0/ 0 ☐ 1. Adam Gilchrist 0 ☒ 2. Mark Waugh 0 0 w 0 n 0 b 0 L Ext 0 0 v 0.0 @ 0.0 pf 0 pb 0 South Africa 1 Herschelle Gibbs = 0 + 0 <table border="1"> <tr> <td>▪</td><td>1</td><td>2</td><td>3</td><td>Match</td><td>Edit</td><td>? Help</td></tr> <tr> <td>4</td><td>5</td><td>6</td><td>>6</td><td>Team1</td><td>Team2</td><td>Bat</td></tr> <tr> <td>w</td><td>n</td><td>b</td><td>L</td><td>Over</td><td>Bowler</td><td>Bowl</td></tr> <tr> <td>←</td><td>n+</td><td>pb</td><td>pf</td><td>Out</td><td>Start</td><td>Utils</td></tr> </table>	▪	1	2	3	Match	Edit	? Help	4	5	6	>6	Team1	Team2	Bat	w	n	b	L	Over	Bowler	Bowl	←	n+	pb	pf	Out	Start	Utils
▪	1	2	3	Match	Edit	? Help																								
4	5	6	>6	Team1	Team2	Bat																								
w	n	b	L	Over	Bowler	Bowl																								
←	n+	pb	pf	Out	Start	Utils																								

Step 4: Complete the forms on the match details and player names by tapping on the Match, Team 1 and Team 2 buttons in succession. These windows appear as shown below. Changes should be made as described under [Notes on Match and Team Details](#). Up to 15 player names can be entered per team. Use the Next buttons to switch to players 12 to 15.

Edit Match Details

Descr ODI
 Team 1 Australia
 Team 2 South Africa
 Venue Perth
 Date 3 February 2002
 Ump 1 Steve Davies
 Ump 2 Bob Parry
 Ump 3 Darryl Hair
 Scorer JDO
 Toss South Africa
 Batting First ☒ Team 1 ☐ Team 2
 (Done) (NbRule)

Edit Team 1 Players

1 Saeed Anwar
 2 Taufeeq Umar
 3 Yousuf Youhana
 4 Inzamam-ul Haq v
 5 Shahid Afridi
 6 Younis Khan
 7 Rashid Latif w
 8 Abdul Razzaq
 9 Azhar Mahmood
 10 Waqar Younis c
 11 Wasim Akram
 (Done) (Next) (Clear) (Fill) (Save) (Fetch) (Imp SD)

Edit Team 2 Players

1 Herschelle Gibbs
 2 Gary Kirsten
 3 Jacques Kallis
 4 Boeta Dippenaar
 5 Jonty Rhodes
 6 Lance Klusener
 7 Mark Boucher w
 8 Shaun Pollock c
 9 Nicky Boje
 10 Allan Donald
 11 Makhaya Ntini
 (Done) (Next) (Clear) (Fill) (Save) (Fetch) (Imp SD)

The match status will still display **Not started**. Tap on the **Start** button, confirm the restart and start scoring.

The **Clear** and **Fill** buttons on the team sheets above allow one to delete all names in the player sheets, and to fill the roster with dummy player names. These buttons only appear when the match has not yet been started, so that names are not erased by accident during the course of a match.

The **Save** button on either of the team 1 or team 2 edit screens above allows one to save the 15 player names comprising the displayed team to a team sheet. Up to 100 team sheets can be stored on the handheld.

The **Fetch** button allows one to retrieve a team sheet that was previously stored on file. The stored team will overwrite the currently displayed team 1 or 2. Also, since batting and bowling orders are linked to the position of the players name in the team list, retrieving a stored team to overwrite current team 1 or 2 must be used with caution once scoring is in progress. From the team selection screen, one can retrieve (tap on the **Fetch** button), delete, beam or export the selected team sheet. Once the limit of 100 stored team sheets has been reached, teams no longer in use can be removed by tapping on the team name to highlight the selection, followed by a tap on the **Del** button. The **Beam** button allows one to beam the selected team across to another handheld device. This can save time at the start of a match when both teams playing in a particular match use Palmcricket for scoring and team sheets have been prepared beforehand. For this action to be effective, the infra-red "eyes" of the beaming and receiving device must be in visual contact with each other.

These team sheet actions are illustrated below:

Save a team sheet

Team sheets limited only by memory
 Team sheets on file = 14
 Save to team sheet name:
 South Africa
 (The name cannot be blank,
 avoid punctuation characters)
 (Save) (Cancel)

Select Team Sheet

Australia
 Bangladesh
 Canada
 England
 India
 Kenya
 Namibia
 New Zealand
 Pakistan
 South Africa
 Sri Lanka
 (Fetch) (Cancel) (Del) (Beam) (Exp SD)

Select Team Sheet

Australia
 Bangladesh
 Canada
 England
 India
 Beam
 Searching...
 (Cancel)

The **Exp SD** button on the Fetch window allows one to export the current team sheet to SD card, from where it can be imported again into the teams database using the **Imp SD** button on the Edit Team form. Importing a team from SD card will not overwrite the current score-sheet teams, but will replace teams in the internal database teams on confirmation, if they already exist. Teams are stored on the SD card in the folder **Palmericket/Teams/** and can be removed from the SD card using the **Delete** button on the team import form.

These SD team sheet actions are illustrated below:



Additional cautionary notes on team sheets: The tiny application (pcrteam.prc) that previously bound all team sheets into a single application is no longer required to activate team sheets. However, if pcrteam.prc is still present on the handheld from a previous version of Palmcricket, the normal PalmOs menu option can be used to delete the pcrteam application. This is not recommended before doing a back-up to a desktop computer via hotsync or SD card, since all team sheets on file will be deleted in the process.

- **Notes on Match and Team Details**

Match and team details as shown in [step 4](#) under the section [starting](#) a new match) can be entered as per the normal Grafitti or keyboard convention for field entries. Double and triple taps, pen dragging, cut and paste etc. are also possible. For cutting and pasting, use the correct diagonal stroke on the Grafitti area to access the cut/paste icons. The match details and player names are retained throughout, even when starting a new match. Additional buttons on these screens allow one to clear all details, and/or to fill blank details with dummy information (useful when opposing team players are unknown and the match starts in a hurry). The scorer can then fill in the correct details during play, as the details become available.

Note that the team batting first must be indicated on the match detail screen before the start of the match (Match status = **Not started**). Once the match is in progress, neither the team batting first nor the no-ball rules may be changed, since this would invalidate any scoring events already recorded. If one makes a mistake in selecting either or both of these options and the match has already been started, then one would need to reverse the start of the match by using the backspace button. Restart the match again after correcting the match settings. Other match details and player names can be changed at any time, bearing in mind that the batting and bowling orders selected during the course of the match cannot be changed, except by using the backspace (see the note below regarding [player order](#)).

Although Palmcricket allows for storage and retrieval of separate team sheets, one can also restore any archived match and then clear only the details that have changed, should the same teams play each other in a new match. Palmcricket never loses the current match details or player names unless the scorer deliberately clears these items. Thus, if one is using Palmcricket to score a full ODI series (e.g. Australia vs. South Africa), where many matches between the same two teams are played in succession, it becomes a simple matter to retain most of the same information, making only minor changes between each match.

Player Order: The order of team names is important in that they track the order selected when batting or bowling. Once names have been entered, and the match is in progress, the name order should remain unchanged. Batter number 1 may be linked to the 5th name, whereas the same player may bowl later as the 3rd bowler. Thus if one changes name number 5, this will be reflected in the batting results at Batter 1, but also at Bowler 3, i.e. the actual order "points" to the correct names in the list. One can of course start off with blank names, and fill them in as the match proceeds. The trick is to allocate player names in logical batting or bowling orders, i.e. always select the batters from player 1 downwards and bowlers from player 11 upwards, even if the names are wrong or blank at first. One gets used to it soon enough. Even with pre-entered names, one can still "cut" and "paste" player names and do some shuffling during the first innings of the match. Avoid using player names 12 to 15, unless there are

actually more than 11 players in the match (a common occurrence with very young teams).

Palmercricket is also quite lenient with regard to shuffling the batting order in any completed or current innings. What often happens in club matches is that the team captain specifies a batting order to the scorer at the start, but then changes the order during the game. Invariably, the scorer is not informed until much later. The [shuffle](#) option allows one to replace any player in any batting position, with another player. Palmercricket automatically corrects all related scores whenever this function is activated.

- [Clocks](#)

Two real time clocks are provided on the top line of the main screen, with elapsed time shown on the left and actual time of day on the right. These clocks are updated every second. The focus of the elapsed time clock can be changed by tapping on the clock and then selecting the focus, as shown in the screen sequence below:

In progress i 0:01:19 9:01:23 ⓘ									
Australia 0/ 1									
☐ 1. Adam Gilchrist w 1									
☒ 2. Matthew Hayden 0									
0 w 0 n 0 b 0 L Ext 0									
Ov 0.2 @ 3.00 pf 0 pb 0									
India									
1 Zaheer Khan 0.2 ov 0/1									
1.1 = 1 + 0									
▪	1	2	3	Match	Edit	? Help			
4	5	6	>6	Team1	Team2	Bat			
w	n	b	L	Over	Bowler	Bowl			
←	n+	pb	pf	Out	Stop	Utils			

In progress i 0:04:06 9:04:10 ⓘ									
Select Clock Display									
☐ Match 00:04:06									
☐ Day 00:04:06									
☒ Innings 00:04:06									
☐ Session 00:04:06									
Done Cancel									
4	5	6	>6	Team1	Team2	Bat			
w	n	b	L	Over	Bowler	Bowl			
←	n+	pb	pf	Out	Stop	Utils			

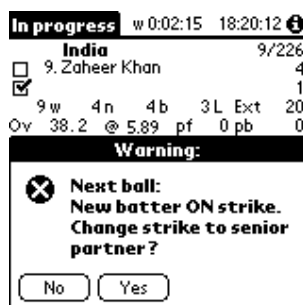
The prefix to the elapsed time clock indicates the focus, where m = match, d = day, i = innings and s = session. The elapsed time clock is still somewhat experimental, and depends on the duration of breaks in play for its accuracy. As at revision 1.29, it is not yet responsive to backspace actions that remove breaks. Incorrect timing calculations may result when breaks are reversed. Further sophistication will be introduced in revision 1.30 to handle timing adjustments to the elapsed time clocks when making backspace corrections to breaks.

Another elapsed timer clock, indicated by the prefix w (for wicket), comes into focus automatically directly after a dismissal. This clock indicates the time elapsed between the dismissal and the next ball bowled, as shown below. It can be used to assist the umpire in deciding upon appeals for "timed out" dismissals.

In progress w 0:01:09 10:03:30 ⓘ									
Australia 1/ 6									
☐ 2. Matthew Hayden 1									
☒ 3. Ricky Ponting 0									
0 w 0 n 0 b 0 L Ext 0									
Ov 0.5 @ 7.20 pf 0 pb 0									
India									
1 Zaheer Khan 0.5 ov 1/6									
1.1.4.W = 6 + 0									
▪	1	2	3	Match	Edit	? Help			
4	5	6	>6	Team1	Team2	Bat			
w	n	b	L	Over	Bowler	Bowl			
←	n+	pb	pf	Out	Stop	Utils			

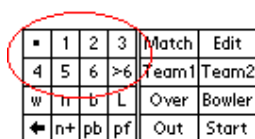
- [Batter on Strike](#)

On the main screen, the striker is "ticked". If for any reason this is incorrect, e.g. after a dismissal), then just tap the correct batter. Generally, the strike changes intelligently, but one needs to keep one eye on the tick at all times, batters get quite upset when one records their boundaries to another player. Note that after dismissals, the program assumes which batter will take strike, but will still pose a question to allow the scorer to override this assumption, as illustrated:



Similarly, when the batters physically run another run on a wide or no-ball, the strike will change automatically on *even* runs, on the assumption that one extra is the default and that a score of say 2 wides implies that the batters crossed once only. Some league bye-laws require double the extras, i.e. an ordinary wide or no-ball scores 2 runs, but this is the exception rather than the rule. After such multiple wides or no-balls, the program will pop up a reminder as to what happened to the strike. If incorrect, the strike must be changed manually, using a single tap on the correct batter.

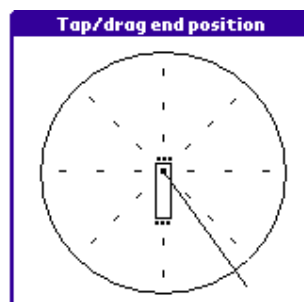
- **Scoring Normal Runs**



A scoring event is recorded by a single tap on any of the numbers (shown circled) in the scoring block located at the bottom left hand corner. The buttons labelled with **dot**, **1** to **6** and **>6** are scoring triggers, i.e. when tapped, they immediately record the score to the batter on strike, the bowler, the batting team's total runs, the number of balls etc. They also record a scoring event in the ball-by-ball database. The **dot** button records a valid ball bowled, without any runs being scored.

Note that the numbered scoring buttons **1** to **6** and **>6** also act in conjunction with eight "shift" buttons. These are labelled **w**, **n**, **b**, **L** and **n+** for normal extras, **pb** and **pf** for penalty runs awarded to the batting and fielding sides, and **Out** for dismissals. The shift buttons can be toggled on or off with successive taps. When "on", they are shown highlighted, i.e. with colours reversed. To activate the shift events, one of the scoring buttons, i.e. **dot**, **1** to **6** and **>6**, must be tapped while a shift button is highlighted (see also the sections on [extras](#), [dismissals](#) and [shift buttons](#)).

Recording wagon-wheel strokes: If the wagon-wheel option has been set in the Utils->Prefs screen, then the wagon-wheel screen will appear after each scoring shot (and dot-balls, if set as well in Prefs). The end-point of the batting stroke can then be recorded by tapping on the oval at the correct point as shown:

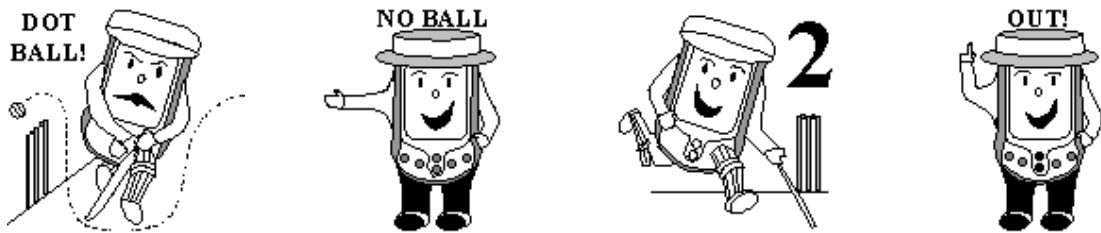


It is up to the scorer to ensure correct positioning and orientation of the stroke. The oval is always viewed from the perspective of the umpire behind the stumps, i.e. long off and on will be at the **bottom** of the screen, whereas third man and fine leg will be at the **top** of the screen.

Wagon-wheel for individual batters can be viewed from the [Batting Summaries](#) screen.

- **Confirmation Screens**

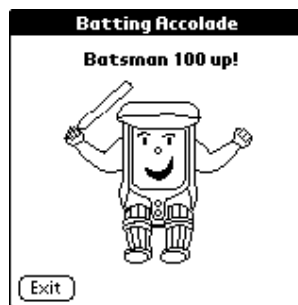
After every scoring event, a suitable confirmation screen pops up for one second. These screens depict Crico, our Palmericket character, in various action poses. A few samples are shown below:



Although we enjoyed conceiving Crico and doing the artwork, the main reason for these pop-ups is to indicate to the scorer that the event was recorded successfully. They also provide a slight pause during which scoring buttons are disabled. This prevents multiple button taps being recorded by mistake (due to the quick program response).

- **Batting Accolades**

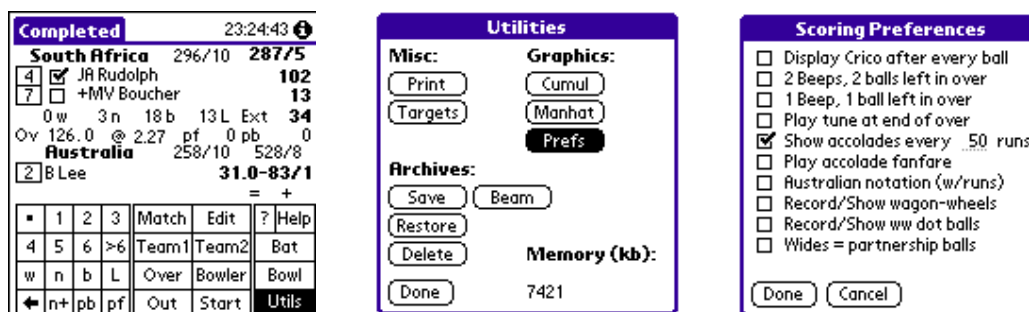
The Crico character makes his appearance with somewhat of a fanfare to award a batting accolade whenever a batter, the current partnership or the total score reaches a milestone in multiples of 50 runs, i.e. at 50, 100, 150 ... etc.



The fanfare sound volume is linked to the game sound volume control accessed via the PalmOs **Prefs** button and can be set to any of the four options: off, low, medium and high. The high setting is recommended for scoring a match while umpiring, whereas the off to medium settings are more appropriate when scoring outside the field of play.

The default accolade interval is 50. For matches involving very young players, intervals of 10, 15 or 20 runs would be more appropriate. The desired accolade interval can be entered by tapping on the **Utils** button to call up the Utilities screen, then tapping the **Prefs** button, entering the desired number of runs and finally tapping the **Done** button. Thereafter, accolades will be triggered at the newly entered run interval.

If desired, accolades can be switched off entirely, using the same sequence: Tap on the **Utils** and **Prefs** buttons, then the checkbox to the left of 'Show accolades every ..50.. runs.', followed by the **Done** button. Alternatively, one could elect to show the accolades, but suppress the fanfare, by checking the accolades box, but unchecking the accolade fanfare, on the **Prefs** screen, as shown in the sequence below:



• [Correcting Scoring Errors](#)



Errors can be corrected immediately using the left arrow (shown circled at the bottom left of the scoring block). This button is also referred to as the "back-space" or "undo" button. It will always reverse the last recorded event, except at the very start of a match. Since all events are stored in the ball-by-ball database, the back-space button can successively reverse every single event, including runs or extras, dismissals, breaks in play, start or end of an innings, or the match completion. In fact, if desired, an entire test match can be "undone" by tapping the back-space button about 2700 times! Such self-inflicted torture would never be necessary, however, since other features allow one to archive the current match and start a new match, at any time, using the same or different players.

Generally, the back-space button should be used only for correcting errors directly after the event. Since there is no "redo" facility to reinstate an event that has just been deleted, the correct event must be recorded using the normal scoring procedure. To correct an error that occurred, say 3 balls ago, would require three taps to reverse the last three balls, then three correct scoring events. In such cases, it would be more efficient to use the [ball-by-ball edit](#) facility, to correct only the specific error. Ball-by-ball edit are preferred for correcting all minor and most major scoring errors, and should always be used as the first choice.

A number of functions are available to correct player selection errors after a dismissal or the end of an over. Refer to the [shuffle](#) option on the batting summary screen for details on correcting batter selection errors and the [mid-over bowling changes](#) or [bowler selection errors](#) for details on correcting bowler selection errors.

• [Scoring Extras](#)



The Extras buttons (shown circled) act as shift buttons, i.e. on their own they do nothing. A single tap on any of these buttons will highlight the button, but the extras are only recorded after tapping one of the run buttons, with one of the extras buttons highlighted.

The extras buttons, clockwise from top left on the scoring block, are defined as follows:

Button	Type of extras
w	Wides
n	No-balls
b	Byes
L	Leg-byes
pf	Penalty awarded to fielding side
pb	Penalty awarded to batting side
n+	No-ball plus runs off the bat

For the latter case, the number of no-balls added to extras as specified under "Match", "No-

Ball rules", the default of 1 is the norm.

Penalty extras awarded to the fielding side are recorded automatically in the previous or next innings, as per the recommendations in Law 42.17 (d), MCC Laws of Cricket, 2000. These will be handled correctly for normal innings sequences, and also when a team follows-on. Fielding penalties are not recorded in the previous or next innings in very rare cases where the same team bats twice in succession, (other than for the standard follow-on). This could only happen if a team forfeits an innings, Law 14.2, or during special coaching matches. In such cases, the penalty runs awarded to the fielding side must be entered manually during the fielding team's next batting innings.

Note the difference between the no-ball and no-ball plus buttons. Runs tapped with the "n" button highlighted are all scored as no-balls (example: batter take a run on a no-ball, without the ball touching the bat: tap "n" then "2" to record 2 no-balls). Runs tapped with the "nb+" button highlighted are scored to the batter on strike, and 1 no-ball is recorded to extras (example: batter hits a no-ball for 4: tap "n+" button followed by the "4" button, this will score 4 runs to the batter, and 1 run to no-ball extras).

The number of no-balls recorded to extras default to 1, but this can be changed at the start of the match to 0 or 2, using the Match button, followed by the nbRule button.

Examples:

▪	1	2	3	Match	Edit
4	5	6	>6	Team1	Team2
w	n	b	L	Over	Bowler
←	n+	pb	pf	Out	Start

To score one wide: tap the **w** button, then button **1**, as shown.

▪	1	2	3	Match	Edit
4	5	6	>6	Team1	Team2
w	n	b	L	Over	Bowler
←	n+	pb	pf	Out	Start

Extra runs on dismissals (eg a stumping on a wide) are recorded by first highlighting the two shift buttons (tapping on buttons **w** and **Out** in any order) and then triggering the event by tapping button **1**, as illustrated.

2 runs to batsman

▪	1	2	3
4	5	6	>6
w	n	b	L
←	n+	pb	pf

1 run to no-ball extras

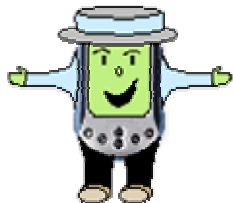
Runs to the batter off a no-ball are recorded by first tapping the **n+** button and then triggering the event by tapping the number of runs scored, i.e. two runs. The no-ball is recorded to no-ball extras, and the batter is credited with the 2 runs, as shown.

A question frequently asked is: "How does Palmcricket score byes on a wide or no-ball?"

The answer of course is that byes cannot be scored off a wide or a no-ball. In terms of the MCC Laws of Cricket 2000, Law 25.6 (Runs resulting from a Wide - how scored) states: "All runs completed by the batters or a boundary allowance, together with the penalty for the wide, shall be scored as Wide balls". This also holds for no-balls (law 24.13).

The umpire should indicate such cases correctly, for example if a wide is missed by

everybody, and it goes over the boundary behind the wicket, the correct score is 5 wides: 1 for the penalty + 4 for the boundary. Also, when the batters run one run on a wide ball, the correct score is $1+1 = 2$ wides (and a change of striker). With Palmcricket, one would tap "w" then "5" for the first example, and "w" then "2" for the second example. In cases where no additional runs are scored, one would tap "n" then "1" or "w" then "1", for no balls or wides, respectively.



Remember: All the extras buttons operate as shift keys: on their own they do nothing, except to supply a different context to the scoring buttons. One of the run buttons must be tapped last in the sequence to trigger the scoring of the extras. If an extras button was tapped by mistake, simply tap it again to remove the shift effect.

• Recording Dismissals

Wickets are recorded by tapping the **Out** button, then a scoring button (**dot**, **1-6** or **>6**). Mostly, the **dot** button will be used to trigger a dismissal where no runs were scored off the dismissal ball. In other cases, simultaneous events can be recorded, eg, stumping off a wide ball: tap **Out**, then **w**, then **1**.

Match in progress										i
South Africa										5/242
☒	3. Jacques Kallis									98
☐	7. Lance Klusener									23
6 w		5 n		0 b		1 L		Ext	12	
Ov	49.1		@ 4.92		pf		0 pb		0	
Australia										7/283
3	Andy Bichel									8.1 ov 1/30
										n. = 0 + 0
1	2	3	Match		Edit		Help			
4	5	6	>6	Team1	Team2		Bat			
w	n	b	L	Over	Bowler		Bowl			
←	n+	pb	pf	Out	Stop		Utils			

In this example, first tap the **Out** button, then the **dot** button, to record a straight-forward dismissal without any runs.

Match in progress										i
South Africa										5/242
☒	3. Jacques Kallis									98
☐	7. Lance Klusener									23
6w		5n		0b		1L		Ext	12	
Ov	49.1		@ 4.92		pf		0 pb		0	
Australia										7/283
3	Andy Bichel									8.1 ov 1/30
										n. = 0 + 0
1	2	3	Match		Edit		Help			
4	5	6	>6	Team1	Team2		Bat			
w	n	b	L	Over	Bowler		Bowl			
←	n+	pb	pf	Out	Stop		Utils			

This example illustrates a typical case of two leg-byes being scored, with a run-out occurring on the third run. Here, the tap sequence is **Out**, **L**, **2**. The event that triggers the dismissal is always one of the run buttons, tapped last in the sequence.

After triggering the dismissal with a run button, the **Out** window will be displayed, as shown below. First tap on any of the allowed methods of dismissal. Secondly, for score-sheet print-outs, it is important to record the correct fielder. This is done by tapping on the **down arrow** button to the left of the word **Unknown** (shown in red), which will "pop" up a list of fielders from which to select the correct player. If uncertain about the identity of the fielder, leave the fielder as "Unknown", since the correct player can be selected at any later stage by using the [ball-by-ball edit](#) facility. In cases where either batter can be given out, just tap on the correct player under the **Who was out** section. Once all the dismissal details are correct, tap the **Done** button.

The Out Window, where the method of dismissal, fielder involved (if any) and the batter dismissed can be selected. For most types of dismissals, the batter dismissed will be the striker, and the program will not allow a change.

After each dismissal, one has the option of selecting the next batter, or recording a break in play. Select the batter from the list by tapping on the player name followed by **Done**. If the dismissal was followed by a break in play, tap **Break**, to bring up the Break Window, as shown..

Batters who are dismissed as "retired" or "retired injured" receive special consideration in that such dismissals are not treated as real wickets, but pseudo wickets. Such batters can bat again, with balls faced and runs carried over from the previous position. Batting summaries, fall of wickets and partnership displays cater for the difference between pseudo and real wickets, since the returning batter will establish new partnerships with other batting partners (or even with the original batting partner).

Palmcricket also places no restriction on players batting again in the same innings after being dismissed normally, since such events occur often in junior cricket teams comprising younger players. The batter is treated as a new player, in such cases. The only restriction is the limit of 15 batting positions.

In terms of Law 9 (b), a retirement without injury constitutes a real dismissal, and such a batter cannot return to resume batting without the consent of the opposing captain. Palmcricket treats this ambiguity automatically by assuming that a batter retiring (without injury) is treated as "retired out", unless the batter bats again later in the same innings. On the resumption of a retired batter's innings, the batter's first session is then treated as "retired not out", and the previous runs are added to the batter's current batting session. Palmcricket is thus quite lenient in allowing any batter to bat more than once in the same innings, regardless of the manner of dismissal. Rigid enforcement of such rules can be applied easily by the umpires, it falls outside the control of the scorer.

Dismissal errors can be reversed immediately by using the back-space button. Errors made during the recording of the dismissal (but only discovered later) can also be corrected via the ball-by-ball edit at any later stage during or after the match. Corrections to the method of dismissal, the batter dismissed or the fielder are possible. Extra care is still necessary when recording a dismissal, since the ball-by-ball edit cannot remove a dismissal, once recorded. Only the back-space button can reverse a dismissal completely.



Remember: The "Out" button operates as a shift key: on its own it does nothing, except to supply a different context to the scoring buttons. One

of the run buttons must be tapped last in the sequence to trigger the dismissal. If the "Out" was tapped by mistake, simply tap it again to remove the shift effect.

- **Ending an Over**

After 6 or more valid balls have been bowled in an over, Palmcricket will give a warning sound, but won't restrict the number of balls. The scorer must indicate the end of over by tapping the **Over** button. Then one can select a new bowler, or record a break in play.

Match in progress

South Africa 5/250
☒ 3. Jacques Kallis 104
☐ 7. Lance Klusener 25
 6 w 5 n 0 b 1 L Ext 12
 Ov 49.6 @ 5.0 pf 0 pb 0

Australia 7/283
 3 Andy Bichel 8.6 ov 1/38
 4.1.1.1.1. = 8+ 0

1	2	3	Match	Edit	? Help
4	5	>6	Team1	Team2	Bat
w	n	b	L	Over	Bowler
← n+	pb	pf	Out	Stop	Utils

The end of the over is triggered by tapping on the **Over** button (shown circled).

After triggering the end of the over, one has the option of selecting the next bowler, or recording a break in play, as shown. Normally, the bowler who bowled the over before the over just completed will be shown highlighted. If there's a change in bowling, tap on the correct player name. Select the bowler from the list by tapping the player name followed by **Done**. If the end of the over was followed by a break in play, tap **Break** to bring up the Break Window, as shown.

Select Next Bowler

Adam Gilchrist
 Mark Waugh
 Ricky Ponting
 Damien Martyn
 Steve Waugh
 Michael Bevan
 Darren Lehman
 Shane Warne
 Brett Lee
 Andy Bichel
 Glenn McGrath

Done Cancel Break

Stop Play

Intervals:
☐ Drinks
☐ Lunch
☐ Tea
☐ Close of Play
☐ Rain
☐ Injury
☐ Lost Ball
☐ Other

End of Innings:
☒ End of overs
☐ Time up
☐ All out
☐ Declaration

End of Match:
☐ Completed
☐ Abandoned

Done Cancel

Palmcricket does not restrict the number of balls per over. The over can be completed after one ball or several, and this is under full control of the scorer. Most cricket matches abide by Law 22.1, which specifically defines an over as consisting of 6 balls. Palmcricket provides only a gentle reminder (a quick series of sounds) after 6 or more balls have been counted. This allows flexibility with regard to umpires miscounting, junior matches where a maximum of 3 wides are allowed, or special club rules which use 8 balls per over, say.

The only restriction is that the current over's ball-by-ball breakdown must not exceed 100 characters. This field appears on the main scoring screen just above the scoring block, and scrolls to the left as more events are added. For example, in the screen at the start of this page, the ball-by-ball breakdown is shown as **4.o.1.1.1.1.** (12 characters all told, including the periods).

A series of quick musical notes are sounded after every event from valid ball 6 onwards. The sound volume is linked to the game sound volume control accessed via the PalmOs **Prefs** button and can be set to any of the four options: off, low, medium and high. The high setting is recommended for scoring a match while umpiring, whereas the off to medium settings are more appropriate when scoring outside the field of play.

- **Mid-over Bowling Changes**

Tap the **Bowler** button followed by the [Mid-over bowling change (injury)] option to change a bowler who cannot complete an over, as shown in the sequence below. This allows partial overs per bowler.



A mid-over bowling change should be used only when **Law 22.8 (Bowler incapacitated or suspended during an over)** applies. Note that the bowler selection screen does not allow a break to be indicated. If there was a substantial time loss due to injury or umpires conferring, use the **Stop** button to record the break before applying a mid-over bowling change.

Don't use the **Bowler** button for normal bowling changes at the end of overs, but the **Over** button (see [ending an over](#)).

- **Correcting Bowler Selection Errors**

Tap the **Bowler** button followed by the [Correct player selection error] option to change a bowler during the current over, as shown in the sequence below.



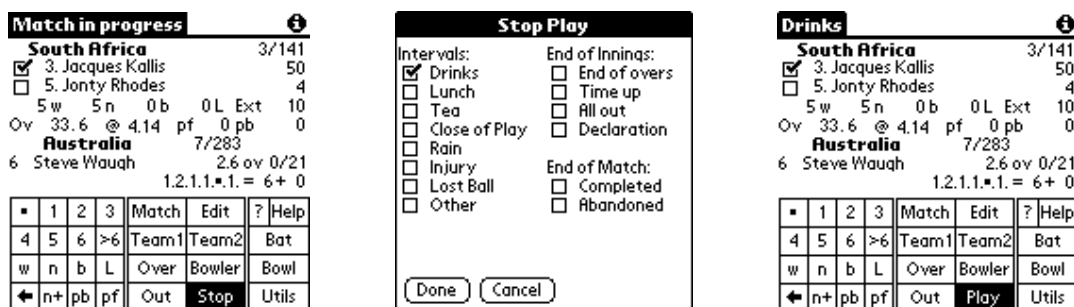
When a different player is selected, all the runs, extras, wickets, etc. which were allocated to the wrong bowler during the current over will be reversed automatically and credited to the correct player.

Bowler selection errors in other overs or innings can be corrected via the ball-by-ball edit screen, as described under [ball-by-ball edits, correcting bowler selection errors](#).

Don't use the **Bowler** button for normal bowling changes at the end of overs, but the **Over** button (see [ending an over](#)).

- **Breaks in Play**

The **Stop/Start** button handles all activities related to starting new matches, breaks in play, ending an innings and completing a match. The sequence below illustrates a typical drinks break.



During the drinks break, the button label changes to **Play**. Tap the button once to resume the game.

A new match can be started at any time during a current match by tapping the **Stop** button and recording the break as **Match completed**. The match status will then change accordingly and the Stop/Start button will read **Start**. Refer to [starting a new match](#) for more details.

Normal breaks in play are handled by tapping **Stop** and selecting the correct break from the **Stop Play** screen. The main screen displays the break, scoring buttons are disabled, and the Stop/Start buttons displays **Play**. Tap **Play** to resume scoring after the break. Breaks in play can also be selected directly after an over has been completed or after a dismissal. This follows the same process as above, except that when scoring resumes after completion of the break, the next bowler or batter must first be selected from the player list. Breaks between innings should be indicated on the **Stop Play** screen by tapping the appropriate end of innings break. Generally, if the end of the innings or match was reached simultaneously with the end of the over (as in the case of limited overs matches), then the end of over need not be indicated first, as the bowler statistics will fix up the partial overs with 6 or more balls bowled.

When play resumes after the end of innings, options are presented which allow for different possibilities (next team bats, follow on etc). Although 4 separate innings are allowed, Palmcricket will not restrict which team bats next, so that it is possible for the same team to bat 4 times, or the first team to bat again immediately, or any other combination within the 4 allowed innings. This provides some flexibility, particularly for coaching purposes, or when the batting positions exceed 15. Coaches can come up with innovative training matches played over a number of days or even weeks, even when scoring other matches between training sessions. The archiving and restore facilities allow such innovations with ease.

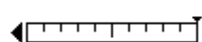
• [Ball-by-ball Edits](#)

The **Edit** Button allows full access to every event during the entire match. This means that one can use the Ball-by-ball edit to correct mistakes much later, e.g. a disputed 4 which was actually a 6, 5 overs ago.

Navigation: The ball-by-ball navigation controls allow one to scroll backwards and forwards by ball, over, or innings in order to position the focus of the edit on the exact ball where the error was made. Their operation is described below:



The navigation section of the ball-by-ball edit screen consists of the top row of buttons, which moves the focus of the current ball as follows in button order from left to right: start of innings, previous over, previous event, next event, next over, end of innings.



The ruler section allows one to position the current focus in larger jumps. Just tap anywhere inside the ruler to move the focus to say half way, quarter way etc. To the left and right of the ruler are arrow buttons which allow one to move the focus to previous or next innings, either or both of these will only appear if there are previous or next innings recorded. Normally, by tapping inside the ruler and using the next

over/previous over buttons (the double arrows in the row of buttons), it should take only about 3 to 4 taps to locate the right over. Then tap on the next ball/previous ball (the single arrows in the row of buttons) to position the focus on the right event, as shown by the over, ball and event indicators at the top left of the screen.

Corrections affecting current ball: To access any over in the current or previous innings, navigate to the correct over and innings as described above. Changes to the current ball are made by tapping the required buttons, and/or striker. If, for instance, a run was scored to the wrong batter, just tick the correct batter. Extras can be switched on or off, in order to correct the type of extra, or to change it to a normal ball by tapping on the **ball** button. In the following example in the 50th over, ball 49.6 is changed from a single by batter 7 to 3 runs by batter 3.

Match in progress

Australia 0/134
☐ 1. Justin Langer 56
☒ 2. Matthew Hayden 70
 1w 4n 1b 2L Ext 8
 Ov 32.6 @ 4.06 pf 0 pb 0

South Africa 10/277
 5 Cloud Henderson 2.6 ov 0/11
 1.L1.n1.w1.b1.4. = 7+ 2

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2
w	n	b	L	Over	Bowler
n+	pb	pf	Out	Stop	Utils

Ball-by-Ball Edit

Ovr 49.6 Evt 311 Inn 2
 3. Andy Bichel
☐ 3. Jacques Kallis
☒ 7. Lance Klusener

Scoring record: (7): 1
 Time 12:29:36

1 2 3
 4 5 6 >6
 w n b L
 ball n+
 Exit

Ball-by-Ball Edit

Ovr 49.6 Evt 311 Inn 2
 3. Andy Bichel
☒ 3. Jacques Kallis
☐ 7. Lance Klusener

Scoring record: (7): 1
 Time 12:29:36
 Change to: (3): 3
 (cancel) Upd

Dismissal events provide additional information regarding the method of dismissal and the fielder involved, if any. These can also be corrected when they appear on screen. To correct a fielder selection error, tap on the fielder name (Mark Waugh in the example below) and select the correct player from the pop-up list as shown.

Ball-by-Ball Edit

Ovr 104 Ball 5 Inn 1
 3. Andy Bichel
☐ 8. Shaun Pollock
☒ 11. Nanti Hayward Caught

1 2 3
 4 5 6 >6
 w n b L
 ball n+
 Exit

▼ Mark Waugh
 Scoring record: (11):
 Time 15:37:45
 Event 634

Ball-by-Ball Edit

Ovr 104 Ball 5 Inn 1
 3. Andy Bichel
☐ 8. Shaun Pollock
☒ 11. Nanti Hayward Caught

1 2 3
 4 Unknown
 w Substitute
 Justin Langer
 ba Matthew Hayden
 Ricky Ponting

To correct the method of dismissal, tap on the incorrect "how out" field and select the correct dismissal from the drop down menu. The correction of the method of dismissal from "caught" to "stumped" is illustrated below:

Ball-by-Ball Edit

Ovr 1 Ball 5 Inn 2
 1. Glenn McGrath
☒ 1. Sachin Tendulkar Caught

Bowled
 LBW
 4 Hit wick
 Caught
 w Stumped
 ba Run out
 Obstruct

Once any changes have been made to the current ball, the **Upd** (update) button will appear. All changes that affect a single ball are only effected after tapping the update button.

More wide-ranging changes that affect more than one ball are also possible. These corrections relate to the bowler, batter dismissed and ball insertions/deletions, as described below. These

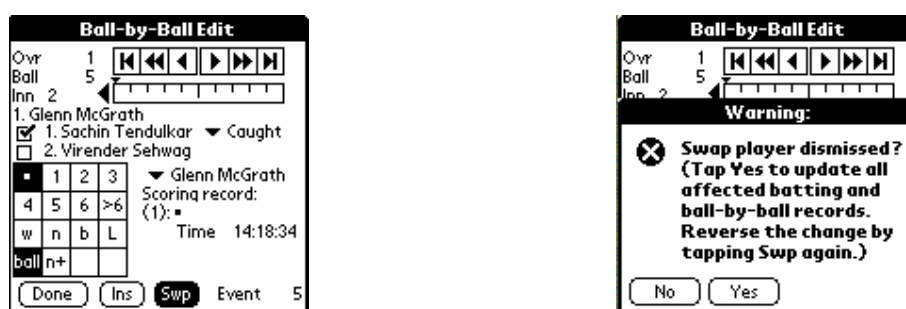
wider changes are effected directly after confirmation. The update button is not applicable in these cases and thus does not appear. Although these changes are far-reaching, they are easily "undone" by repeating the correction using the original bowler or batter dismissed, or by applying the opposite action for dot ball insertions and deletions.

Bowler corrections One can also use the ball-by-ball edit screen to correct a bowler player selection error made during the current over, or any previous over in any innings. Position the ball-by-ball focus on any ball in the over of interest, tap on the bowler name and follow the prompts, as shown in the sequence below:



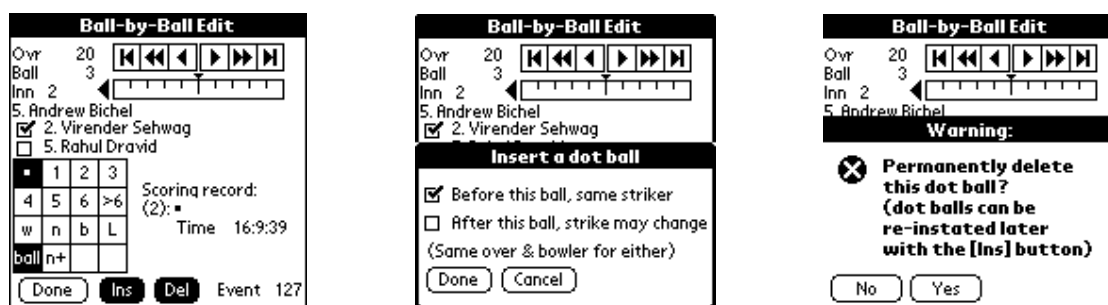
The bowler for the current over in the current innings can also be changed directly from the main screen [Bowler] button as described in [Correcting Bowler Selection Errors](#).

Batter dismissed: Very occasionally, the wrong batter is dismissed and the mistake is only discovered much later, after a number of balls have been bowled or the innings or match has been completed. From a player and scoring perspective, this is quite a significant error, since all runs scored subsequent to the dismissal will be allocated incorrectly to the player that continues to bat and the unfortunate batter dismissed in error will be deprived of those runs. The ball-by-ball edit overcomes this scoring predicament by allowing one to swap the dismissed and not-out players for any dismissal ball in the innings. This correction can be applied at any time. All subsequent balls and all affected batting records in the innings are adjusted automatically and instantaneously. Such corrections are triggered easily by tapping on the **Swp** button and confirming the correction, as shown below:



This correction applies only to correcting the *dismissal* of the wrong batter. It must not be confused with the [shuffle](#) action that can be triggered from the batting summary to correct errors made when selecting the *next player coming out to bat*.

Dot ball insertions and deletions: These functions allow one to add or erase dot balls anywhere in a score-sheet. Indirectly, this allows one to insert or delete any valid ball other than dismissals, since dot balls can be converted to other scoring balls and vice versa, using the normal ball-by-ball edit functions. Such corrections are triggered easily by tapping on either the **Ins** or **Del** buttons and confirming the correction using the **Yes** button, as illustrated below:

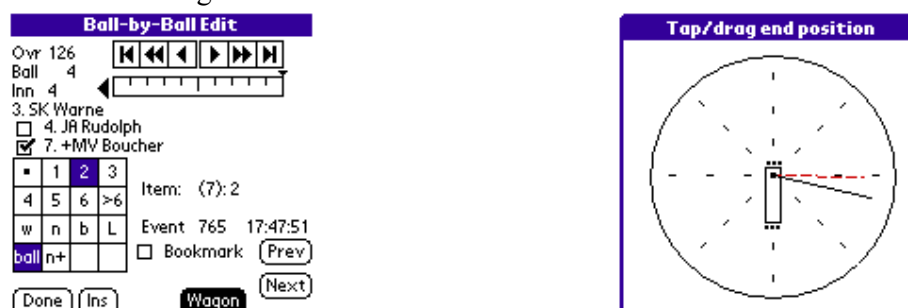


Dot balls are inserted with an event time one second earlier or later than the current event's recorded time.

Ball-by-ball edits allow correction of all minor and most major scoring mistakes, quickly and effectively, and is the preferred approach. The advantage of making corrections in this way is that the time for each ball, as recorded originally, is retained. [In contrast, the [left arrow button](#) effects a backspace deletion, which loses the previous ball. Hence if one goes too far back, one must re-instate the in-between balls. Not only is it quite difficult to reconstruct events accurately when many different actions took place, but the original timing for every ball re-scored is lost.]

Although ball-by-ball edits are quite powerful in correcting runs, extras, batter on strike, bowler, fielder, method of dismissal or batter dismissed, extra care is still necessary when recording dismissals. *A dismissal cannot be erased by a ball-by-ball edit.* This can be achieved only by using the left arrow and then re-scoring the dismissal from scratch, directly after the event, as explained under [correcting scoring errors](#) and [dismissals](#) above.

Correcting wagon-wheel strokes: If the wagon-wheel option has been set in the Utils->Prefs screen, then any recorded wagon-wheel batting stroke can be corrected by tapping on the **Wagon** button and correcting the stroke as follows:



Bookmarks: Bookmarks can be set at any event in the ball-by-ball database by tapping on the **Bookmark** checkbox, as shown:



At any subsequent stage during or after the match, one can scroll sequentially backwards or forwards through all recorded bookmarks by tapping on the **Prev** and **Next** buttons. In the example above, a bookmark is set at the dismissal of JM Kemp, perhaps because of uncertainty whether the dismissal was a bat-pad catch or LBW. One can then return later to

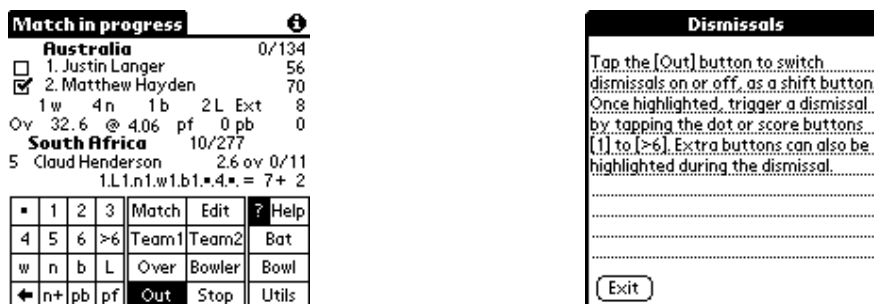
this bookmark using the **Prev** button, so that the dismissal can be corrected.

• [On-line Help](#)

The Help button covers 20 help topics, with each topic providing abbreviated information, useful when one is at the match, far away from a desktop. Tap on the **Help** button, followed by the topic button as shown below:



The **?** button is a "context sensitive" shift key, i.e. when highlighted, it can be used to access additional help screens directly related to each scoring button. This works by tapping the **?** button, followed by a tap on the scoring button of interest.



Again, the information is somewhat abbreviated, but there's enough to resolve most queries that arise while scoring in the field.

• [Batting Summaries](#)

The **Bat** button is used to display batting summaries at any stage, for all completed innings. This is quite useful when players want to know what their figures during or after the match, or when scorers need to reconcile score sheets.

Match in progress

Australia0/126

1. Justin Langer56

2. Matthew Hayden66

0w3n0b1LExt4

Ov32.1@3.91pf0pb0

South Africa10/277

6 Lance Klusener0.0ov0/0

1.= 1+

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Stop Utils

India 10/234

No	Batter	Out	Min	Balls	Runs
1.	Sachin Ten	c	3	5	4
2.	Virender Se	r/o	102	80	83
3.	Sourav Gar	c	43	25	24
4.	Mohamma	c	1	2	0
5.	Rahul Dravi	b	86	57	47
6.	Yuvraj Sinc	c	44	35	24
7.	Dinesh Mor	c	14	12	12
8.	Harbhajan	c	12	8	7
9.	Zaheer Kh	c	21	8	4
10.	Jayaqal Sri	b	5	4	1
11.	Ashish Nehr	*	6	4	8
Extras (9w 4n 4b 3Lb Op)					20

Done

FoW

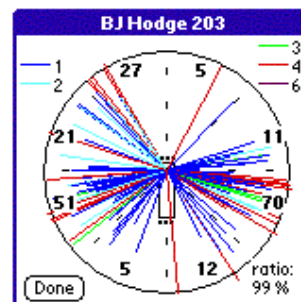
Part

Shuf

On the summary screen, the left and right arrow buttons allow one to scroll to the previous or next innings. The up and down arrows scroll the screen up or down by four players, so that one can view statistics for all 15 positions (normally only applicable in junior cricket matches involving very young players, 9 and 10 year olds).

To see a detailed ball-by-ball analysis of the player's batting performance as shown below, tap on any player's name. The wagon-wheel chart will only be shown if the correct option was set in Util->Prefs and the scoring strokes were recorded during the match.

Jacques Kallis 38			
caught Adam Gilchrist			
bowled Andy Bichel			
*****2*****1*****			
*****4*****2*****3*****2*****			
*****2*****23*****			
*****14*****>>			
Time in: 16:25 Time out: 11:55			
1170 - 998 = 172 minutes			
125 balls: 108 dots, 6 x [1], 6 x [2], 2			
x [3], 2 x [4], 1 x [6]			
Exit			



[Note: Refer to [time spent at the crease](#) for an explanation of how batting times are calculated].

Fall-of-wicket and batting partnership statistics can also be viewed by tapping the **FoW** or **Part** buttons on the batting summary screen.

South Africa FoW				
No	Batsman	Over	Evt	Runs
1.	Herschelle Gibb	14.2	89	24
2.	Gary Kirsten	18.2	115	36
3.	Boeta Dippenaar	30.4	189	59
4.	Jacques Kallis	59.3	365	131
5.	Lance Klusener	59.4	366	131
6.	Mark Boucher	77.5	476	198
7.	Neil McKenzie	89.1	545	220
8.	Claud Henderso	89.6	550	225
9.	Alan Donald	91.6	562	233
10.	Nanti Hayward	103.5	634	277
Extras (0w 7n 1b 10Lb)				18
Done ▶ ▼ Bat Part				

India Partnerships					
No	Bat1	Bat2	Min	Balls	Runs
1.	Virender Se	Sachin	3	5	4
2.	Virender Se	Soura	43	57	54
3.	Virender Se	Mohar	1	4	1
4.	Rahul Dravi	Virenc	54	80	88
5.	Yuvraj Sinc	Rahul	32	48	40
6.	Dinesh Mor	Yuvra	10	19	21
7.	Harbhajan	Dinest	3	3	1
8.	Zaheer Kh	Harbh	8	11	14
9.	Zaheer Kh	Javaq	5	7	3
10.	Ashish Nehr	Zahee	7	6	8
Done ◀ ▼ FoW Bat Shuf					

In the case of batters batting again, having retired or been dismissed earlier, the returning batter will occupy another batting position. Since the database is limited to 15 players and 15 batting and bowling positions, this becomes a constraint when the combination of total players and returning batters exceed 15 (again only applicable to junior cricket).

The **Shuf** button is used to correct a batter selection error in any innings, at any time after the error. This allows an incorrect batting order to be "shuffled" by selecting the correct player to replace an incorrect player at any completed or current batting position. This requires only four quick steps. First tap the **Shuf** button, then tap on the incorrect batter, next select the correct batter and finally, tap the **Done** button.

Tap on Batsman to replace				
No	Batsman	Out	Balls	Runs
1.	Gary Kirsten	bowled	52	10
2.	Herschelle Gibb	caught	42	14
3.	Boeta Dippenaar	caught	52	26
4.	Jacques Kallis	caught	125	38
5.	Neil McKenzie	lbw	160	67
6.	Lance Klusener	caught	1	0
7.	Mark Boucher	caught	65	43
8.	Shaun Pollock	*	94	42
9.	Claud Henderso	run out	3	5
10.	Alan Donald	caught	6	0
11.	Nanti Hayward	caught	22	14
Done ▶ ▼ FoW Part Bat				

Replace 2. with...	
Gary Kirsten	
Herschelle Gibbs	
Jacques Kallis	
Neil McKenzie	
Boeta Dippenaar	
Lance Klusener	
Mark Boucher	
Shaun Pollock	
Claud Henderson	
Alan Donald	
Nanti Hayward	
Done Cancel	

All runs, balls faced and event timing are credited to the correct player. The incorrect player is placed back in the "hut", with status set to "did not bat".

This correction applies only to correcting an error made originally when selecting the *next player coming out to bat*. It must not be confused with the correction of a [dismissal](#) error, available under the ball-by-ball edit.

• Time Spent at the Crease

Palmcricket uses the handhelds' real-time clock to store, to the nearest minute, the start and end of a batter's innings when the player is selected and dismissed, respectively. For not-out batters, the end time is taken as that of the last ball bowled in the innings. The hand-held clock's date and time should of course be set correctly before the match commences, for such timing to be effective. Also, for greater accuracy, batters should be selected only when they start walking onto the field of play.

In addition, the time for each event is recorded in the ball-by-ball database. The program automatically determines significant breaks in play (whether these were recorded by the scorer or not) when the time between successive balls bowled exceeds 3 minutes. Time spent by batters at the crease is calculated as the number of minutes that have elapsed between the recorded start and end times. During the batter's innings, breaks in play are allowed for by deducting all delays longer than 3 minutes and replacing each of these with a nominal 40 seconds of play.

Typical calculations of the time spent at the crease are illustrated in the following batting examples from the 2nd Test, Australia vs South Africa, played at Melbourne on 26 December 2001. The start of the first day, scheduled for 11:00, was delayed for 79 minutes due to rain. Play resumed at 12:20, with a break for lunch at 13:03, a few minutes after the normal scheduled time. After lunch, there were two more interruptions for rain (101 minutes and 35 minutes), and a drinks break of 3 minutes. The usual tea-break fell away and the play ended at 18:00.

Day 2 started half an hour earlier at 10:30 and there were no further delays caused by rain. Drinks were taken at 11:49, 14:44 and 17:15, and lunch started at 13:00. South Africa were bowled out at 15:38, two minutes before tea. The change of innings thus overlapped into the tea-break (refer to the example under [innings timing](#) for more complete details). Play resumed at 15:59 and continued until 18:23, when bad light stopped play, with Justin Langer and Matthew Hayden still at the crease.

Herschelle Gibbs 14	
caught Ricky Ponting	
bowled Glenn McGrath	
*****1***2*****4****1**14****	
1>>	
Time in: 11:00 Time out: 15:33	
273 - 219 = 54 minutes	
42 balls: 35 dots, 4 x [1], 1 x [2], 2 x [4]	
Exit	

Neil McKenzie 67	
bowled Brett Lee	
*****41***2****4*****	
*****3****1**2****114****1**1	
*1***4*****112*****1**	
121***3***>>	
Time in: 17:18 Time out: 14:29	
1271 - 1035 = 236 minutes	
160 balls: 127 dots, 17 x [1], 6 x [2], 2 x [3], 8 x [4]	
Exit ▲ Top	

Matthew Hayden 66*	
not out	
*****4***214*****1**2**1*****4	
*****1*****1***3***41141**	
4*****24*****144*2**	
Time in: 15:58 Time out: 18:23	
145 - 3 = 142 minutes	
100 balls: 74 dots, 9 x [1], 5 x [2], 1 x [3], 11 x [4]	
Exit	

The 219 minutes deductions for breaks during Herschelle Gibbs' innings consisted of a rain delay of 79 minutes, 39 minutes for lunch and another rain delay lasting 101 minutes. (Herschelle incidentally lost his wicket in the 2 minutes of play between the second and third stoppage for rain). Neil McKenzie started his innings at 17:18 and continued batting into day 2. His 1035 minutes of deductions for breaks comprised 989 minutes overnight, lunch of 39 minutes, two drinks breaks of 3 and 2 minutes duration and an unexplained 2 minute delay, just before the last drinks break. (The latter delay was caused by the scorer's lapse in concentration). The last session on day 2 had only one stoppage of 3 minutes for a drinks break, as illustrated on Matthew Hayden's batting summary.

[Note: Refer also to [batting summaries](#), section for guidelines relating to the display of batting details shown above. The example score sheet is supplied with the Palmcricket Viewer download and can be viewed and printed on a Windows desktop. The file can also be installed on the handheld by copying it to the installation directory, and then invoking the Palm Hotsynch procedure. From there, it can be copied into the current match within Palmcricket by

restoring it from the archive, in order to explore the example in more detail.]

- **Bowling Summaries**

The **Bowl** button is used to display the bowling summaries at any stage, for all completed innings. To see a detailed ball-by-ball analysis by over, tap on any players name on the summary screen shown below in the middle. This will bring up a screen similar to the one shown below on the right, where the over number, individual balls, analysis by over and running analysis are displayed on each line.

Match in progress

Australia0/126

☐ 1. Justin Langer56

☒ 2. Matthew Hayden66

0w 3n 0b 1L Ext 4

Ov 32.1 @ 3.91 pf 0 pb 0

South Africa10/277

6 Lance Klusener0.0 ov 0/0

1. = 1+

Australia Bowling

Bowler	O	M	R	W	Econ
GD McGrath	24	11	39	0	1.6
B Lee	31	9	83	1	2.7
SK Warne	46.5	21	87	3	1.9
NW Bracken	19	5	37	1	1.9
A Symonds	3	0	6	0	2.0
BJ Hodge	2	0	8	0	4.0

Done

▼ Det

Tim

w/n

1. Glenn McGrath

[1] [0-0] 0-0

[3] 1 [2-0] 2-0

[5] 1 [1-0] 3-0

[7] [0-0] 3-0

[9] 4 [4-0] 7-0

[13] 4 [4-0] 11-0

[15] W [2-0] 14-1

[17] [5-0] 19-1

[19] W [0-1] 19-2

[21] 2 [2-0] 21-2

Exit

▼

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Stop Utils

Bowling economy, in average runs per over is normally displayed in the far right column. Tap on the **w/n** button to switch to displaying wides and no-balls.

On the summary screen, the left and right arrow buttons allow one to scroll to the previous or next innings. The up and down arrows scroll the screen up or down by four players, so that one can view statistics for all 15 positions (normally only applicable in junior cricket matches involving very young players, 9 and 10 year olds). On the detailed ball-by-ball screen on the right, up and/or down arrows will appear when there are more than 10 overs. This allows one to scroll through all possible overs bowled by the specified bowler.

The **Detail** button above is used to display the bowling analysis for the entire innings, for each over in sequence. As for the analysis by bowler, scroll buttons allow one to view any batch of approximately ten overs. The **Top** and **Bottom** buttons allow one to move to the first and last over in the innings.

Over Details		
[20] 1.....	[1-0] 73-0	Jacques
[21]	[1-1] 75-0	Shaun P
[22] 4.....	[3-1] 83-0	Jacques
[23] 1.....	[4-1] 89-0	Shaun P
[24] 1.....	[3-0] 92-0	Alan Do
[25] 4.1.2..	[7-0] 99-0	Shaun P
[26] 4.....	[4-0] 103-0	Alan Do
[27] 1.....	[1-0] 104-0	Shaun P
[28]	[2-0] 106-0	Alan Do
[29] 2.....	[2-0] 108-0	Claud H

- **Innings Timing**

One can easily examine the real-time progress of any innings by tapping on the **Timing** button on the bowling summary screen, as illustrated in the sequence below. In this example, drawn from the 2nd test between Australia and South Africa played at Melbourne on 26 December 2001, the first day's play was interrupted by rain on 3 occasions, hence the overflow of events onto a second screen.

Australia Bowling									
Bowler	O	M	R	W	w	n			
Glenn McGrath	26	8	70	2	0	1			
Brett Lee	31	10	76	3	0	0			
Andy Bichel	19.5	6	45	3	0	5			
Shane Warne	19	3	56	0	0	1			
Mark Waugh	8	1	19	1	0	0			

Innings 1 timing									
11:00:00 Batsmen walk on.									
11:00:00 - 12:20:04 Rain (79:24)									
12:20:04 First ball									
13:03:31 - 13:42:22 Lunch (38:11)									
13:51:06 - 15:32:53 Rain (101:07)									
15:34:29 - 16:10:54 Rain (35:45)									
17:36:44 - 17:40:51 Drinks (3:27)									
18:00:16 - 10:30:29 Close (989:33)									
11:49:10 - 11:52:48 Drinks (2:58)									
13:00:36 - 13:40:31 Lunch (39:15)									

Innings 1 timing									
14:40:00 - 14:43:01 Delay (2:21)									
14:43:55 - 14:47:01 Drinks (2:26)									
15:37:45 Last ball									
1718 - 1294 = 424 minutes									
103.5 overs									
Over rate: 14.7 per hour									
(40 sec play assumed per break)									

These summaries are particularly useful when submitting match results for club leagues where timing calculations and over rates are required.

(Refer also to [Time Spent at the Crease](#) to see how these relate to individual batters).

• Utilities

The **Utils** button brings up the Utilities screen that offers a number of facilities. Currently, the active buttons are **Print**, **Targets**, **Cumul**, **Manhat**, **Prefs**, **Save**, **Restore**, **Delete**, **Beam**, **Export** and **Import**, as shown:

Completed 14:56:29 ⓘ

India 10/234

☐ 11. Ashish Nehra 8

☒ 9w 4n 4b 3L Ext 20

Ov 39.2 @ 5.94 pf 0 pb 0

Australia 2/359

1 Glenn McGrath 8.2 ov 3/52

■.W■. = 0+

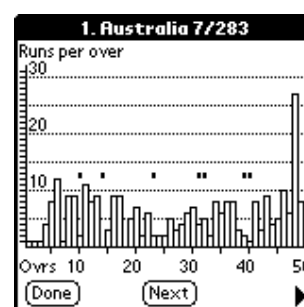
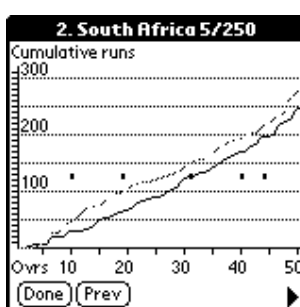
■	1	2	3	Match	Edit	?	Help		
4	5	6	>6	Team1	Team2			Bat	
w	n	b	L	Over	Bowler			Bowl	
←	n+	pb	pf	Out	Start			Utils	

Utilities									
Misc					Graphics				
Print					Cumul				
Targets					Manhat				
					Prefs				
Archives					SD Card				
Save					Export				
Restore					Import				
Delete									
Done					Free RAM: 7744 kb				

The **Print** button will print the full score sheet directly to a printer via the infra-red interface. It requires the Stevens Creek Software print server to activate the infra-red protocols (see notes at the start of this manual).

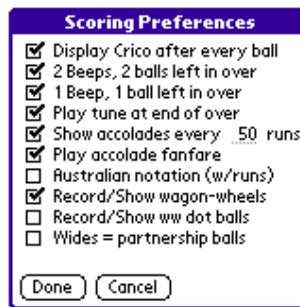
The **Targets**, **Cumul** and **Manhat** buttons will display screens similar to the samples below:

Innings Target									
Total runs required 284									
Total overs available 50									
Overall rate required 5.68									
Runs behind 70									
Remaining overs 4.2									
Current run rate 4.69									
Remaining run rate 15.90									
Done Recalculate									



The dots on the graphic screens indicate fall of wickets. **Prev** and **Next** buttons allow one to view other innings in the match, if any. Left and right arrows scroll the graphic screens horizontally, so that one can view the graphs over the entire range of overs. Due to pixel limitations, only 50 overs fit on the screen.

The **Prefs** button allows one to set various parameters that control Palmcricket's behaviour during scoring, as illustrated:



Most of these are self-evident, from the check-box descriptions. Tap on each item to enable or disable the required behaviour. Accolades are described in more detail under [Batting Accolades](#)

The **Registration** button is no longer visible. As from version 1.50, Palmcricket is available free of charge and registration of a Palmcricket licence is no longer necessary. Copyright, however, remains vested in J deV Oberholzer and Multi-Media publications.

Save, **Restore** and **Delete** buttons allow one to manage match archives, by saving from or restoring to the current match, or deleting older archives no longer required, as per the following screens:



When saving to archive, the name must be unique without any punctuation characters, otherwise the save will be rejected. When restoring from or deleting a specific archive, the archives on file will be displayed. Tap on the archive name, followed by the **Restore** or **Delete** buttons to execute the command.

The full current match is always stored automatically throughout, which means there is no need for constant "back-ups". One loses nothing when accessing other applications on one's handheld. For example, it's useful to record memos during the course of the match. The odd back-up during a match can be done at any time, but it's not critical.

When using the Palm Hotsync utility, the current and all archived matches are transferred into the \Palm\Username\Backup directory, from where they must be transferred manually using the Windows Explorer (R) to a folder where they will not be affected by subsequent Hotsync procedures. These match files can be viewed or printed using the Palmcricket Viewer (available as a separate download). This is explained in more detail in the readme.txt file in the viewer download.

The **Beam** button allows one to beam an archived match across to another handheld computer using infra-red communication, as illustrated in the following sequence:



For this technique to be effective, the infra-red "eyes" of the beaming and receiving handheld devices should be in direct visual contact with each other.

The SD Card **Export** and **Restore** buttons allow one to manage the transfer of match archives to and from SD memory cards, as per the following screens:



When exporting or importing, a list of match names will be displayed. Tap on the match name, followed by the **Export** or **Import** buttons to execute the command.

To delete a score-sheet on the SD card, select the score-sheet, tap the **Delete** button and confirm the deletion as shown:



SD cards allow one to transfer matches to and from Windows desktop computers that can read and write SD cards, without having to rely on the Palm Hotsync process. Score-sheets are stored on the SD card in the folder Palmcricket/Matches.

- **Match Completion**

Tap the **Stop** button on the main screen, and then tick the Completed box in the **Stop Play** window, followed by **Done**. This will change the match status indicator, top of the main screen to **Completed** as shown.

In progress m 06:59:44 16:23:20

India 10/234

☐ 11. Ashish Nehra 8

☒ 9 w 4 n 4 b 3 L Ext 20

Ov 39.2 @ 5.94 pf 0 pb 0

Australia 2/359

1 Glenn McGrath 8.2 ov 3/52

W* = 0+

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Stop Utils

Stop Play

Intervals: ☐ Drinks ☐ End of Innings: ☐ End of overs

☐ Lunch ☐ Time up

☐ Tea ☐ All out

☐ Close of Play ☐ Declaration

☐ Rain

☐ Injury

☐ Lost Ball

☐ Other

End of Match: ☒ Completed ☐ Abandoned

Done Cancel

Completed 14:56:29

India 10/234

☐ 11. Ashish Nehra 8

☒ 9 w 4 n 4 b 3 L Ext 20

Ov 39.2 @ 5.94 pf 0 pb 0

Australia 2/359

1 Glenn McGrath 8.2 ov 3/52

W* = 0+

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Start Utils

Record the result and any additional notes as explained below in [Match Result and Notes](#). At this point, it would be useful to save the match to archives, so that one can restore it later to a desktop computer and publish the match results. This is achieved by tapping the **Utils** button, followed by the **Save** button on the Utilities window. Enter a name for the archived match and tap **Save**. After completion of the save to archive, tap the **Done** button to return to the main screen. This sequence is illustrated below:

Completed 14:56:29

India 10/234

☐ 11. Ashish Nehra 8

☒ 9 w 4 n 4 b 3 L Ext 20

Ov 39.2 @ 5.94 pf 0 pb 0

Australia 2/359

1 Glenn McGrath 8.2 ov 3/52

W* = 0+

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Start Utils

Utilities

Misc:

Graphics:

Archives:

Memory (kb): 1304

Done

Save current match to archive

Up to 10 matches can be archived.
Current archive count = 4

Destination archive name:
2003 World Cup Final

The name must not be blank,
avoid punctuation characters,
(eg !@* / - . ? etc)
use only alphanumerics.

Save Cancel

• [Match Result and Notes](#)

Upon completion of a match, the result and two additional notes can be entered by activating the button sequence shown below:

Completed 14:36:14

India 10/234

☐ 11. Ashish Nehra 8

☒ 9 w 4 n 4 b 3 L Ext 20

Ov 39.2 @ 5.94 pf 0 pb 0

Australia 2/359

1 Glenn McGrath 8.2 ov 3/52

W* = 0+

1	2	3	Match	Edit	? Help
4	5	6	>6	Team1	Team2 Bat
w	n	b	L	Over	Bowler Bowl
←	n+	pb	pf	Out	Start Utils

Edit Match Details

Descr World Cup Final

Team 1 Australia **Result**

Team 2 India

Venue Wanderers

Date 23 March 2003

Ump 1 David Shepherd

Ump 2 Steve Bucknor

Ump 3 Rudi Koertzen

Scorer JDQ

Toss India

Bat 1st ☒ Team 1 ☐ Team 2

Done

Match Result and Notes

Result: Australia win by 125 runs

Note 1: Man of the match: Ricky Ponting

Note 2: 2003 World Cup champions: Australia

(Max char = 46, 38 and 38)

Done

These notes will be stored in the database, and will be saved and restored during any match archive operations.

• [Conclusion](#)

Although these guidelines are fairly extensive, in reality scoring is much easier than it may appear from reading these notes, and the best way to learn is to score a real match. No doubt there are many more questions, a FAQ (Frequently Asked Questions) list is available on the Palmcricket web-site.

The web-site is at: <http://www.palmcricket.com>

Support, questions or suggestions via e-mail: support@palmcricket.com