

# Glucotools™

*By Tony Cureington*

**Copyright © 2002-2003, James A. Cureington**

[tonycu@users.sourceforge.net](mailto:tonycu@users.sourceforge.net)

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

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

You may also find a copy of the GNU General Public License at <http://www.gnu.org/copyleft/gpl.html>

GlucTools™ is a trademark of James A. Cureington. All other trademarks are the property of their respective owners.

*To my loving and supportive wife, Judy.*

# Table of Contents

Terms of Usage.....	5
Introduction.....	6
What is GlucoTools?.....	7
PDA Requirements.....	7
Update Notices.....	7
Version Differences.....	8
Configuration and Customization.....	9
Setup Meal Bolus.....	9
Setup Correction Bolus.....	11
Setup Preferences.....	12
The Main Screen.....	14
Meal Bolus.....	14
Snack Bolus.....	16
Correction Bolus.....	16
BG Reading.....	17
Correction Carbs.....	17
Bolus History.....	18
More Tools.....	19
Blood Glucose Averages.....	19
Misc Info.....	20
Estimated HbA1c.....	20
Export.....	21
Frequently Asked Questions.....	22
Getting Support.....	24
How Do I Get Help?.....	24
I Found A Bug, What Should I Do?.....	24
How I Can Help?.....	25
Feature Creep.....	26

# Terms of Usage

---

You must agree to the following terms before using GlucoTools:

1. I will NOT make any insulin dosage or other decision based on GlucoTools
2. I use GlucoTools at MY OWN RISK and will consult my physician before making any changes whatsoever
3. I have read and agree to the terms of the GNU GPL located at <http://www.gnu.org/copyleft/gpl.html>

If you don't agree to the above terms, don't use GlucoTools.

# Introduction

---

In March of 2002 I began using an insulin pump to better manage my diabetes and allow more flexibility with meal times. I quickly discovered computing insulin dosages based on the number of grams of carbohydrates in the meal and my current blood glucose reading was a bit more than I could do without a pencil and paper. I searched for a tool that would allow me to compute meal and correction boluses using different insulin to carbohydrate ratios for different times of the day, but found none. Over the next couple weeks I pondered what such a tool should do and began sketching various user interface screens.

On April 21, 2002 I registered the GlucoTools project at SourceForge and shortly thereafter began the software development life-cycle. Even though I had never written a PDA application before, in November I released version 1.0 of GlucoTools.

GlucoTools is feature complete and stable. While there are a few other elaborate features that could be added, they are not currently a priority for the project - see the *Feature Creep* section for details. The main priority now is creating a desktop JavaScript application to display the information contained in the GTLS-BH.pdb history database. Using JavaScript will allow it to be OS independent. If you are a JavaScript hacker and would like to contribute, see the *How Can I Help?* section. All the code is open source released under the GNU GPL and available from <http://glucotools.sourceforge.net>.

If you have ever contemplated a pump, I would encourage you to discuss the benefits with your physician and attend a pump support/user meeting. If I had attended one of these meetings earlier in Austin, Texas, I probably would have started pumping a long time ago.

If you received a copy of this documentation from anyplace other than <http://glucotools.sourceforge.net>, please download a copy from <http://glucotools.sourceforge.net>. This is the only mechanism I have of determining how many individuals are using the tool since the program is beamable.

**The only way others find out about GlucoTools is through word of mouth. If you know someone who may benefit from GlucoTools, please give them the link.**

Don't beam a newer version of GlucoTools over an older one. When I beamed a version from my Visor Neo to my Palm III, it deleted all GlucoTools databases on the Palm III. It is fine to beam GlucoTools to someone that does not already have it installed. None of GlucoTools databases are beamed, only the application itself. I've not seen any problems when upgrading from one version to another using the cradle.

## What is GlucoTools?

GlucoTools is a set of tools for the Palm Pilot, HandSpring Visor, and other PalmOS based PDAs that assist pumpers (diabetics using insulin pumps) with managing diabetes. GlucoTools is handy to double check meal and correction boluses.

Some of GlucoTools features are:

- Blood glucose values can be entered in mg/dL or mmol/L
- Computes meal boluses based on different insulin to carb (grams of carbohydrates) ratios for different times of the day - enter the number of carbohydrates and get the insulin dosage to cover the carbohydrates at the specific time of day.
- Computes correction boluses based on different blood glucose targets and insulin to blood glucose ratios for different times of the day. It even computes negative correction boluses when below the target to offset the meal bolus.
- Estimates HbA1c.
- Computes average units of insulin used per day for boluses.
- Computes average carbohydrates consumed per day.
- Computes average blood glucose tests per day.
- Records blood glucose readings.
- Averages blood glucose readings for different times of the day.
- Able to specify number of days for which averages and estimated HbA1c are computed over.

## PDA Requirements

GlucoTools requires the following:

- PalmOS version 2.0 or later. I've only tested this on a Handspring Visor Neo running PalmOS v3.5.2H3.0, and limited testing on a Palm III running PalmOS version 3.0.
- Approximately 50KB of memory for the application itself.
- Twenty-four bytes of memory for each history record. Two hundred and forty history records (the default) would consume 5.8KB.

## Update Notices

To get an announcement when a new version of GlucoTools is released, subscribe to <https://lists.sourceforge.net/lists/listinfo/glucotools-announce>. When a new version is released, a notice will be posted there shortly afterwards.

# Version Differences

Version 1.0 released November 15, 2002

1. Initial release

Version 2.0 released June 3, 2003

1. Added support for mmol/L blood glucose readings and computations
2. Added min/max blood glucose reading with date and time on Miscellaneous Information screen
3. Allow one to specify the number of days all averages and estimated HbA1c are computed for
4. Added date and time modification fields to Correction Carbohydrates entry screen
5. Check for negative bolus in Meal Bolus computations - this could happen if the computed correction bolus was negative and the absolute value of the correction bolus was greater than the meal bolus itself
6. Increased maximum number of user configurable history records from 2040 to 9999
7. Changed minimum and maximum blood glucose range to 18 and 720 mg/dL, or 1 and 40 mmol/L, respectively

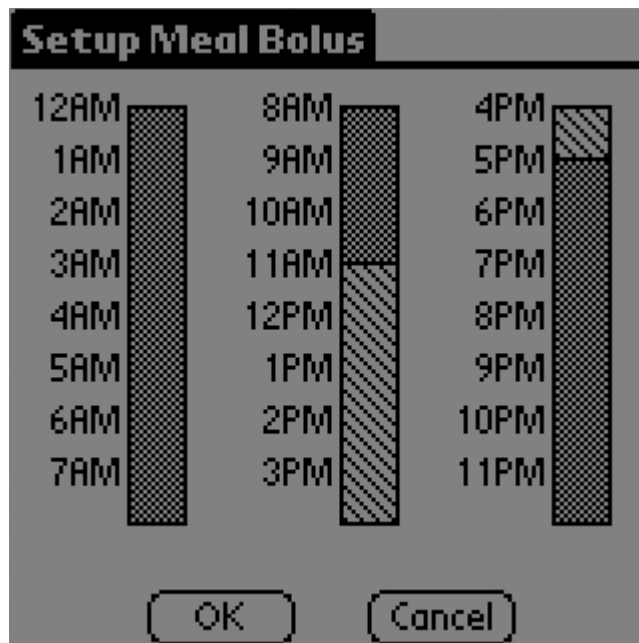
# Configuration and Customization

---

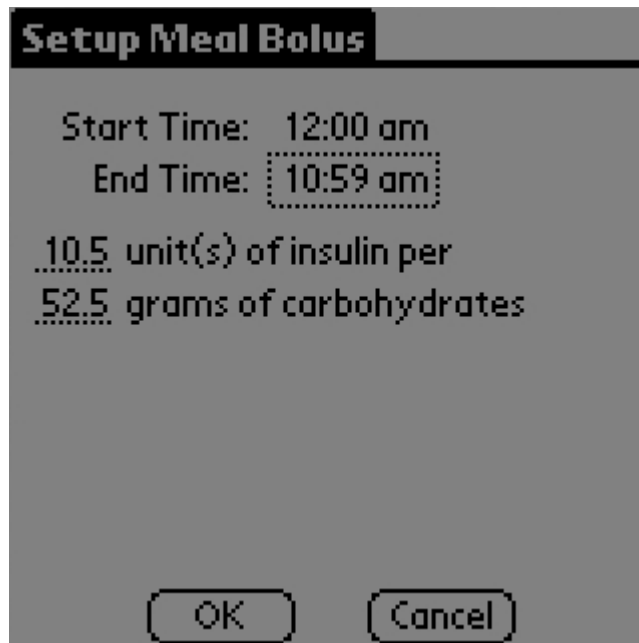


This section describes the configuration required before using GlucoTools. Each item can be found under the *Setup* menu item, shown above, accessible from the main screen. The configuration is presented here, before the main screen, because this configuration is required before using GlucoTools. There are three items under the *Setup* menu, each is explained below.

## Setup Meal Bolus

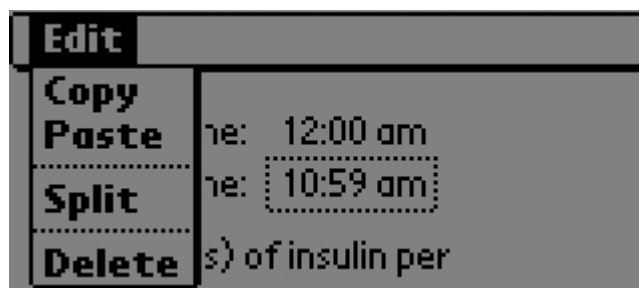


Select the *Meal Bolus* menu item to configure the insulin units to grams of carbohydrates for different times of the day to be used in meal and snack bolus computations. Initially there is a single meal bolus entry that spans from 12AM to 11:59PM. The screen below has three records as can be seen from the shading - the first record is from 12AM to 10:59AM, the second from 11AM to 4:59PM, and the third from 5PM to 11:59PM. Tapping anywhere in the 12AM to 10:59AM shaded area displays the screen below.



If the end time is changed from 10:59AM to 11:59AM, the next records start time will automatically change to 12:00PM. The start time cannot be changed directly, it can only be changed indirectly by adjusting the previous records end time.

The units of insulin to grams of carbohydrates is also specified in the above screen for the specific time interval. The 10.5 units per 52.5 grams of carbohydrates is convenient for me since I eat the same thing for breakfast during the week. This could be any combination you choose, 1 unit per 13 grams of carbohydrates or 0.5 units per 10 grams of carbohydrates - the insulin to carb ratio is simply computed from the two values. On the weekend I may eat more or less than 52.5 carbohydrates for breakfast, the same insulin to carb ratio is used to calculate the insulin dose for for the number of carbohydrates.



When editing a meal bolus record, there are four menu items available. Each item is discussed below.

The *copy* menu item copies the contents of the record for later pasting.

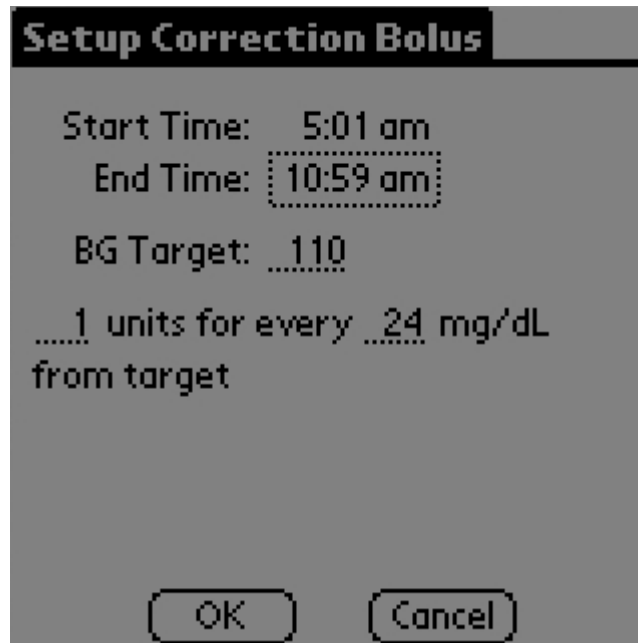
The *paste* menu item copies the contents of a previously copied record into the current record, the end time is not copied.

The *split* menu item splits the current record into two separate records dividing the time interval evenly between the records. When a record is *split*, the second half must be viewed and the OK button tapped to confirm the contents of the split record. If an attempt is made to compute a meal or snack bolus when a record has not been viewed and the OK button tapped, an error message will appear indicating which record has not been confirmed. The only way to cancel a split record is to cancel back to GlucoTools' main screen or delete the additional record created from the split.

The *delete* menu item simply deletes the record. When a record is deleted, the previous records' end time is extended to that of the deleted record. An exception to this is when the first record is deleted, in which case the second record's start time becomes 12AM making it the first record.

Tapping *Cancel* cancels the changes made in the current screen. The exception to this, discussed above, is when a record has been split.

## Setup Correction Bolus



The *Correction Bolus* setup is similar to the meal bolus, except when a time segment is tapped the screen above is displayed.

The *BG Target* value is simply the blood glucose target for the specified time interval. In the screen above we see 1 unit of insulin lowers this persons blood glucose by 20 points. Decimal numbers such as 0.5, 0.75, and 1.3 can be entered for the units of insulin.

How is the correction bolus computed? Using the values in the screen above, we show the formula GlucoTools uses to compute the correction bolus for a current BG reading of 145.

$$(145-110)*(1/20) = 1.75$$

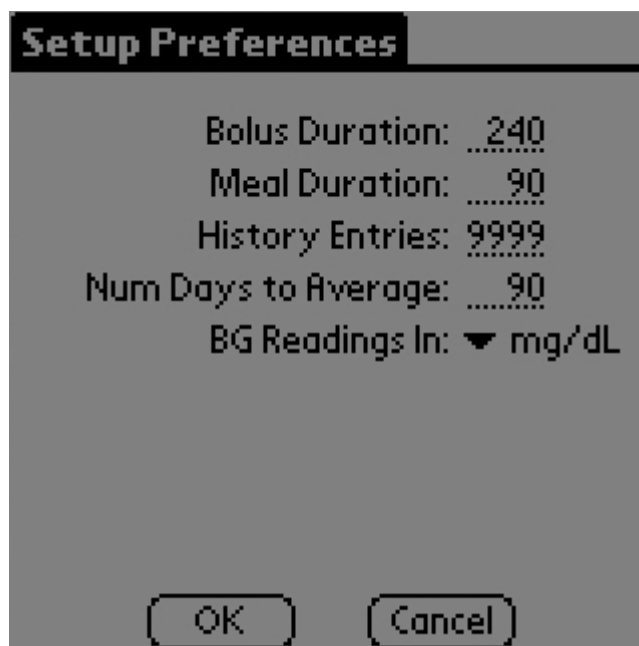
This is simply the number of points away from the target multiplied by the correction factor. This also works for BG readings less than the target. Below is the correction bolus for a BG reading of 85.

$$(85-110)*(1/20) = -1.25$$

The negative correction bolus would be added to the meal bolus decreasing the total bolus by 1.25 units. More on this in the *Meal Bolus* section.

If using mmol/L, the minimum mmol/L that may be specified is 1. If you need, for example, to specify 1 unit of insulin for every 0.75 mmol/L you could achieve this by specifying 2 units for every 1.5 mmol/L.

## Setup Preferences



There are currently three items under the *Preferences* menu item. Each item is discussed below.

The *Bolus Duration* indicates how long a bolus is active in minutes. If another bolus is computed while a previous bolus is still working/active, a bolus warning will be displayed along with the computed bolus. Setting the bolus duration to 0 prevents any bolus warning messages. An example of this can be seen later in this document when the Snack Bolus is discussed.

The *Meal Duration* specifies how long a meal lasts in minutes. If a meal bolus is computed and a snack bolus is later computed within the number of meal duration minutes from the previous meal bolus, that snack bolus will be added to the previous meal bolus. This allows one to start a meal not knowing exactly how many carbohydrates will be consumed over the course of the meal. For example, A meal bolus is computed at 5:00pm for a meal and at 5:45pm a snack bolus is computed to cover a salad and crackers. Since the snack bolus was 45 minutes away from the meal bolus and the meal bolus duration is 90 minutes in this case, the carbohydrates and units of insulin are added to the meal bolus that was computed at 5:00pm. Setting the meal bolus duration to 0 prevents any snack boluses from being added to previous meal boluses.

The *Bolus History Entries* simply specifies how many history records to retain. When the number of specified entries is reached, the oldest entry is deleted. Once a history record is deleted there is no way to get it back. For example, if there were currently 200 history records and the bolus history history entries was changed from 240 to 50, the next time a record is saved, history records 51 through 201 (200 + the one just added = 201) will be deleted leaving 50 records.

The *Num Days to Average* specifies how many days to use in computing the averages and estimated HbA1c - essentially all the computations under the Tools pull-down menu. This is useful for reviewing the averages for the past thirty and ninety days. **A value of 0 computes the averages over all the history records.**

The *BG Readings In* specifies blood glucose fields are in mg/dL or mmol/L. This can be changed at anytime without loss of information, you can change back and forth as often as

you like. However you need to be aware of some anomalies - **this can be skipped if you don't plan to switch back and forth between mg/dL and mmol/L**. Here is an example:

Say the BG Target is 110 mg/dL and we take 1.1 units for every 23 points from the target.

Now we convert the above mg/dL to mmol/L ( $110/18 = 6.1$  and  $23/18 = 1.3$ )

So for mmol/L the BG target is 6.1 mmol/L and we take 1.1 units for every 1.3 points from the target.

Finally we compute a correction bolus for a 499 mg/dL ( $499/18 = 27.7$  mmol/L) and get:

$$\begin{aligned} &\text{mg/dL} \\ &499-110=389 \\ &389/23=16.91 \\ &16.91*1.1=18.6 \text{ units} \end{aligned}$$

$$\begin{aligned} &\text{mmol/L} \\ &27.7-6.1=21.6 \\ &21.6/1.3=16.6 \\ &16.6*1.1=18.2 \text{ units} \end{aligned}$$

We now see the anomaly, 18.6 units was computed for mg/dL while 18.2 units was computed for mmol/dL even though the blood glucose values were the same. The above units are correct for each, even though they're not equal when one might expect them to be.

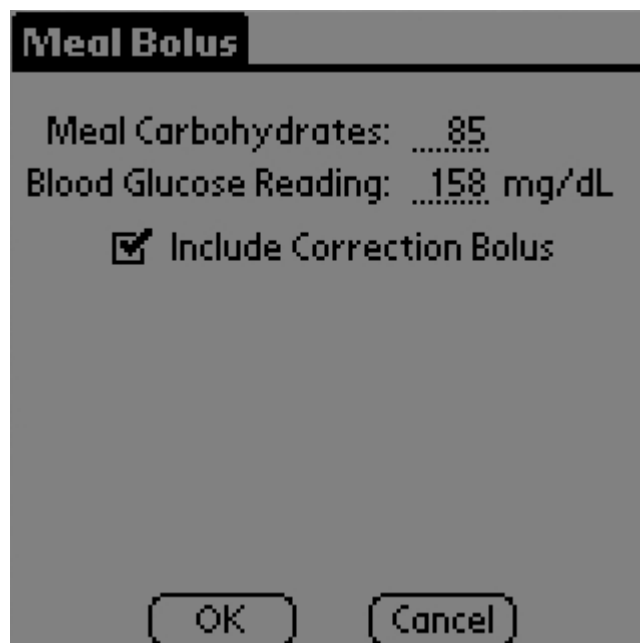
# The Main Screen

---



The above screen is displayed when GlucoTools is launched. Each of the buttons will be discussed below.

## Meal Bolus



The above screen is displayed when *Meal Bolus* is tapped. Enter the number of carbohydrates in the meal and your current blood glucose reading and tap OK; three values are displayed in a pop-up window as shown below. After entering the carbohydrates, you can enter “i” in the alphabet graffiti area (not the numeric area) to advance to the blood glucose reading field.


**Meal Bolus**

Meal Carbohydrates:   
Blood Glucose Reading:  mg/dL

Include Correction Bolus

---

**BOLUS AMOUNT**

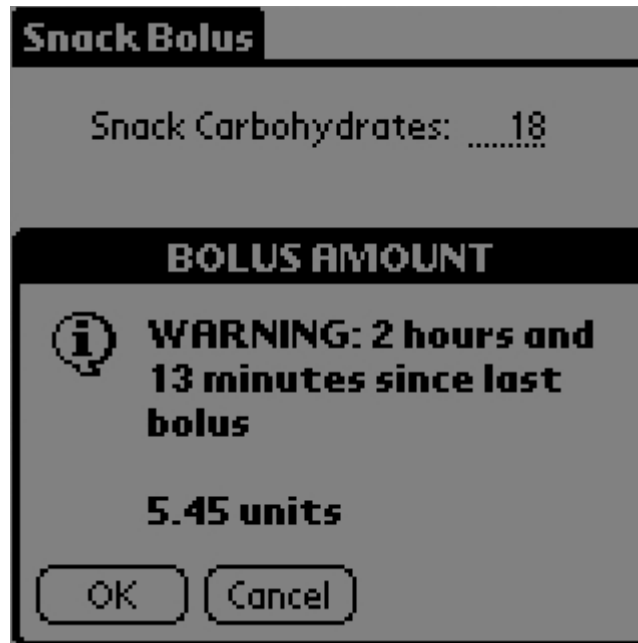
 **Meal bolus: 17.00**  
**Correction bolus: 1.60**  
**Total bolus: 18.60**

The three computed values are the meal bolus, correction bolus, and the total bolus. If the blood glucose reading were less than the blood glucose target, the correction bolus would be negative. Boluses are computed for the current time - there is no way to change the time the bolus is computed for - this is for real-time computations.

If the *Include Correction Bolus* check box is unchecked, the correction bolus will not be computed. This is handy when you know your previous bolus is still working and you don't want a correction bolus for your blood glucose reading. For example, I ate breakfast at 8:30AM and am now setting down for lunch at 12:00PM (three and a half hours later), my current blood glucose reading is 130 and I know my bolus usually drops my glucose level by 20 points in the last half hour, I would not want to compute a correction bolus for my current blood glucose reading. When the box is unchecked, "\*\*\*\*\*" is displayed in the correction bolus field indicating it was not computed.

Tapping Cancel prevents the values from being saved in a history record - they are only saved if OK is tapped.

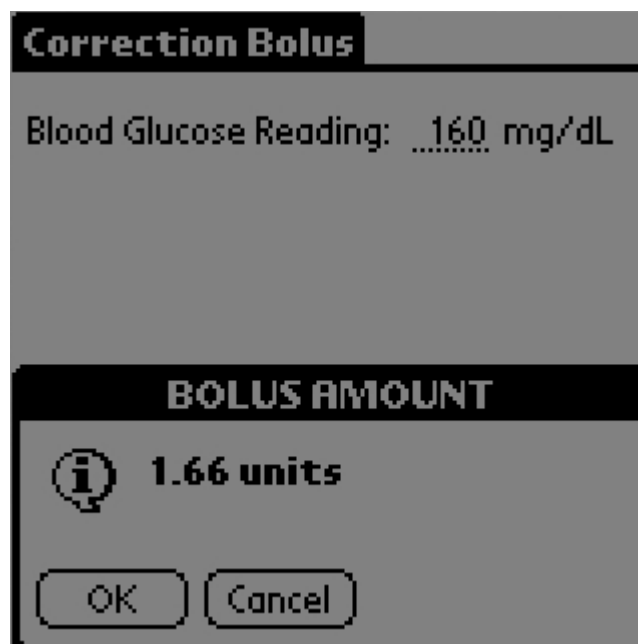
## Snack Bolus



A *Snack Bolus* is to cover carbohydrates and does not include any correction bolus. Above we see a total of 5.45 units are required to cover 18 carbohydrates at the present time. The screen above also shows the message when the *Bolus Duration* is in effect - warning that the last bolus was administered 2 hours and 13 minutes ago.

Tapping Cancel prevents the values from being saved in a history record - they are only saved if OK is tapped.

## Correction Bolus

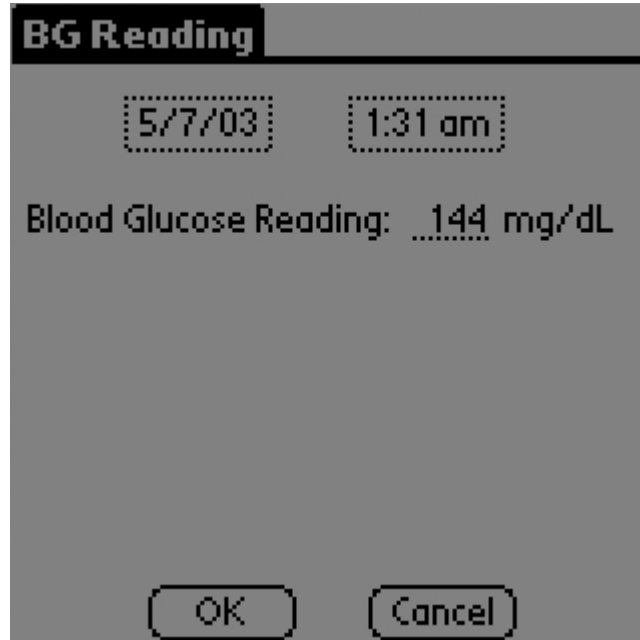


A *Correction Bolus* is computed to cover a high blood glucose reading. No carbohydrates are specified in this calculation. The calculation is based on the values specified in the *Setup*

*Correction Bolus* screen. Gain a thorough understanding of correction boluses from your physician, **going hypoglycemic when you are sleeping can kill you!**

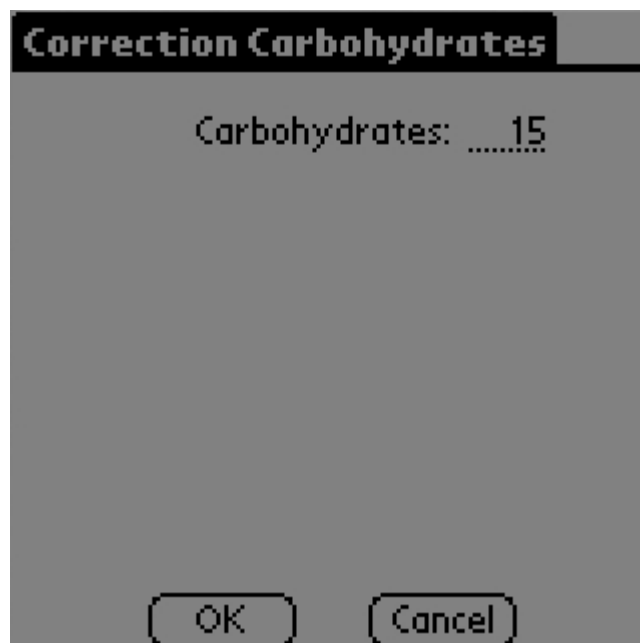
Tapping Cancel prevents the values from being saved in a history record - they are only saved if OK is tapped.

## BG Reading



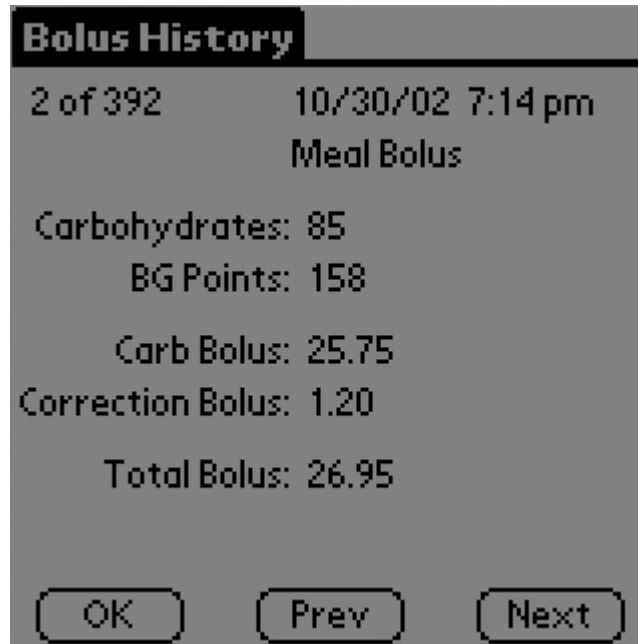
The time and date of a blood glucose reading can be changed when it is entered. Tapping Cancel prevents the values from being saved in a history record - they are only saved if OK is tapped.

## Correction Carbs



The *Correction Carbohydrates* screen is where one enters carbohydrates for which no insulin is taken, such as for correcting hypoglycemia (low blood glucose).

## Bolus History



Not too much explanation needed here, other than the *Bolus History* records are stored in sorted order from newest to oldest. The hardware scroll up/down buttons will also advance to the previous/next history record.

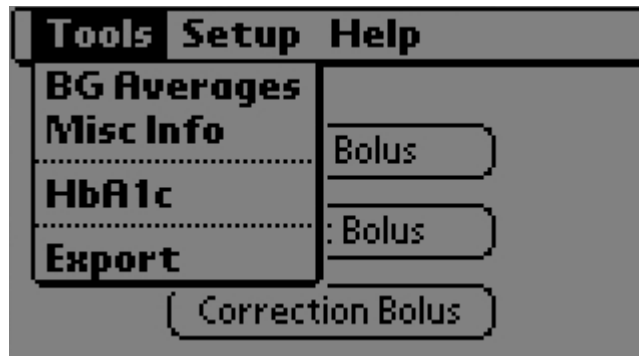


When viewing history records, the date and time can be changed from the drop down menu. Changing the time does not cause the bolus amounts to be recalculated to reflect the new time. One may want to change the time to 6:30PM, for example, if a meal starts at 6:00PM and ends at 7:00PM with snack boluses computed over the course of the meal for additional carbohydrates - averaging the start and end time.

History records can be deleted in this screen. Once a record is deleted there is no way to get it back.

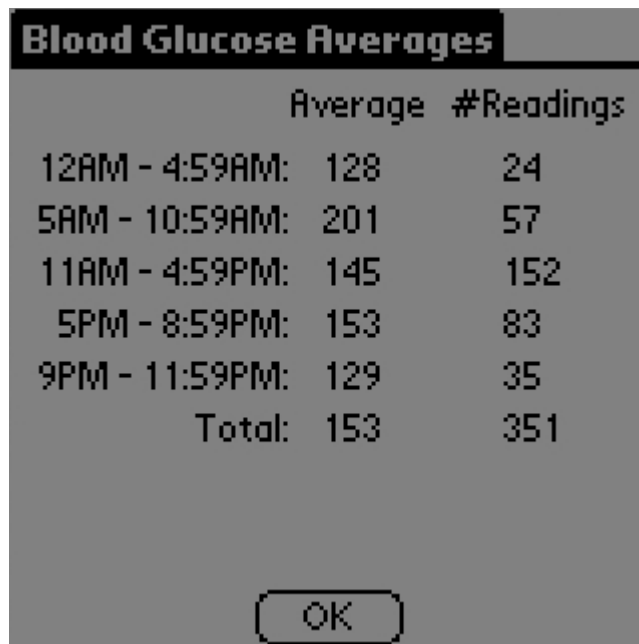
# More Tools

---



This section describes other features of GlucoTools. These additional features are found under the *Tools* menu.

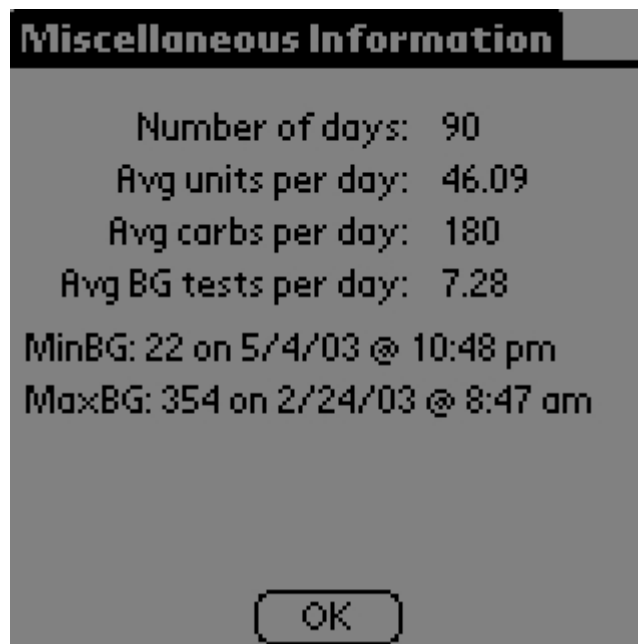
## Blood Glucose Averages



	Average	#Readings
12AM - 4:59AM:	128	24
5AM - 10:59AM:	201	57
11AM - 4:59PM:	145	152
5PM - 8:59PM:	153	83
9PM - 11:59PM:	129	35
Total:	153	351

This is simply the averages for various time periods. The total average and total number of readings is in the last row. The averages are computed from the meal bolus, correction bolus, and blood glucose reading records. Of course, the snack bolus and correction carbohydrate records are not included in the computation since they don't contain blood glucose readings. The number of days over which the averages are computed is limited to the number of days specified in the *Num Days to Average* field on the Setup Preferences screen.

## Misc Info



This is the catch-all screen. Of course the average units per day is only for the bolus units and does not include basal units. Since the basal rates normally don't change for an average day, GlucoTools does not provide a mechanism to record them. Most pumps show the total daily basal rate, adding that value to the average units per day gives the actual average total units per day. The number of days for which the information is computed is limited to the number of days specified in the *Num Days to Average* field on the Setup Preferences screen.

## Estimated HbA1c



This estimated HbA1c is probably one of the more interesting features. The estimated HbA1c is computed using Nathan's formula which is

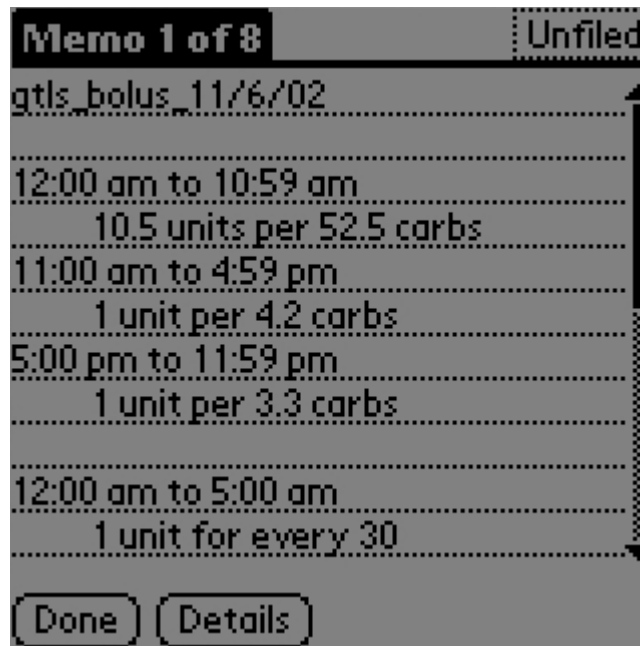
$$\text{HbA1c \%} \times 33.3 - 86 = \text{MBG}$$

Where MBG is the Mean Blood Glucose (arithmetic mean = average)

$$\text{HbA1c} = (\text{MBG} + 86) / 33.3$$

More readings result in a more accurate estimate, some have suggested that seven tests a day provides a fairly good estimate. The estimated HbA1c computed over the number of days specified in the *Num Days to Average* field on the Setup Preferences screen.

## Export



The export feature dumps the meal bolus and correction bolus information to a MemoPad memo named `gtls_bolus_mm/dd/yy` (i.e. `gtls_bolus_11/13/02`). This is convenient for physicians to review, or to print and store in a safe place or carry with you in the event your PDA becomes damaged.

### **Exporting the bolus information should be done before upgrading to a newer version of GlucoTools.**

The contents of a complete exported bolus information memo is shown below.

```
gtls_bolus_10/26/02

12:00 am to 10:59 am
      10.5 units per 52.5 carbs
11:00 am to 4:59 pm
      1 unit per 4.2 carbs
5:00 pm to 11:59 pm
      1 unit per 3.3 carbs

12:00 am to 5:00 am
      1 unit for every 30
      points from 110
5:01 am to 10:59 am
      1 unit for every 20
      points from 110
11:00 am to 4:59 pm
      1 unit for every 25
      points from 110
5:00 pm to 11:59 pm
      1 unit for every 40
      points from 110
```

# Frequently Asked Questions

---

## What tools did you use to develop GlucoTools?

All development was done on the Linux Operating System (information available from <http://www.linux.org>) using the following tools:

Palm OS SDK, available from <http://www.palmos.com/dev/tools> (I used an older version, v3.5)

PilRC - the PILot Resource Compiler, available from here <http://sourceforge.net/projects/pilrc/>

POSE - the Palm OS Emulator, available from <http://www.palmos.com/dev/tech/tools/emulator>, binary package available from <http://sourceforge.net/projects/pose>.

Some instructions putting the above tools together can be found at <http://www.linuxdoc.org/REF/palmdevqs> - several of the links are broken, but the instructions are none-the-less helpful. You can find the things I did to get the SDK up and running in the palm\_sdk\_build\_notes.txt file found in the GlucoTools source tree.

You may want to check out <http://aquapalm.sourceforge.net> for another example of developing Palm applications on Linux.

## I see you run Linux, what did you use to write this document?

This document was created with OpenOffice.org, a complete feature-rich office suite freely available from <http://www.openoffice.org> that runs on many popular operating systems.

## I enter 15.5 carbohydrates for a snack, but it changes the number of carbohydrates to 15 after I press OK. Why is this happening?

If a fraction was allowed it would significantly increase the memory to store a history record. A fraction of a carbohydrate is negligible and limiting the carbohydrate to an integer value was considered worth the space savings.

## I get up in the morning and take a blood glucose reading before eating breakfast. I would like to take a correction bolus for high readings immediately since I'm not sure exactly how long it will be before I eat breakfast or how many carbohydrates will be in my breakfast. Is there any way to do this?

This can be done by computing a meal bolus with 1 carbohydrate, and bolusing only the correction bolus amount. When the number of carbohydrates for breakfast is determined, subtract one carbohydrate from the total and enter that carbohydrate count as a snack bolus. Then go to the bolus history and view the bolus for the number of carbohydrates, you wouldn't include the correction bolus since it was previously taken (and is part of the total bolus in the history screen).

## I can set the time and date in the BG Entry field to a future time and date; I can do this in the Bolus History screen too. Is this a bug?

No. The code space required to check this is simply not worth it. People will usually enter the readings in real-time with no need to change the date most of the time.

**I started using GlucoTools today and have entered three meal boluses. I select the Misc Info menu item and N/A is displayed for various items. What is wrong?**

Nothing. Any daily average does not include the current day. This is done so only complete days are included in the daily average computations - a partial day would cause inaccuracies in the averages.

**In the bolus history screen there are sixty records, but when I compute the averages all sixty records aren't used. Why?**

The bolus history includes all records, while the blood glucose averages don't include snack bolus and correction carbohydrate records, and insulin/carbohydrate averages don't include blood glucose reading records.

**I entered a meal bolus, then changed the time in the bolus history screen to a time with a different insulin to carbohydrate ratio and the insulin dosage did not change. Why not?**

Once a meal or snack bolus is computed, the dosage amount does not change. Changing the time/date in the bolus history screen only changes the time/date - it does not cause the dosages to be recomputed. All dosages are based on the current time.

**Sometimes computing averages takes a while, how do I know the program is not hung?**

There is a status bar in the lower left portion of the screen that will move back and forth during long computations.

**When I change a meal or correction bolus end time, the screen has a start and end time that are the same. Is this correct?**

Yes. On older versions of PalmOS such as the Palm III runs, this was the only built in screen to select a time from. Both the start and end time will change when a new time is selected. Newer versions of the PalmOS, as found on the Visor Neo, have a built in screen from which a single time can be selected.

**When I compute a bolus, when is it stored in the bolus history database?**

Information is only stored when the OK button is tapped. If the Cancel button is tapped, or the GlucoTools is exited, the information is not saved.

# Getting Support

---

All support is provided through email lists. There are two lists for getting help, the glucotools-users list for user type questions and the glucotools-devel list for development and bug reporting. It is recommended that all GlucoTools users subscribe to the glucotools-announce list, discussed in the *Introduction* section, to be notified of new software releases. Specific details on getting help are provided below.

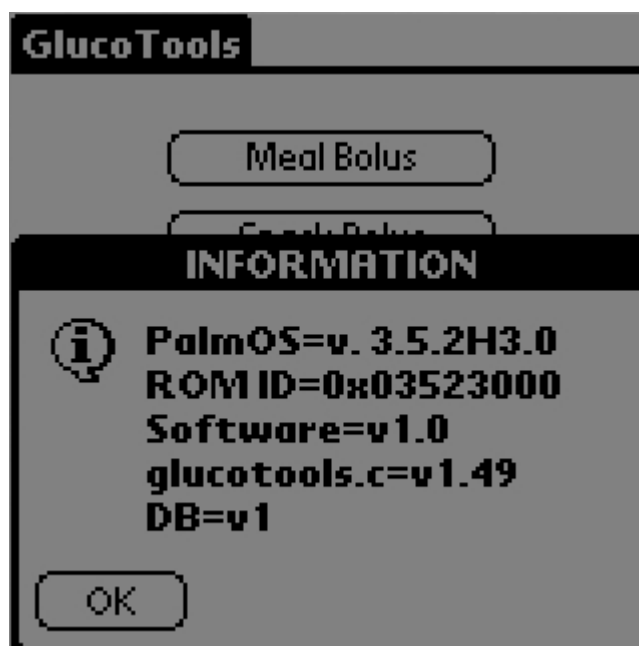
## How Do I Get Help?

First, review the archives of previous user questions at <https://lists.sourceforge.net/lists/listinfo/glucotools-users>. If you don't find the answer, post a message - it is considered proper etiquette to subscribe to the list before posting. If you are not familiar with list/net etiquette, see the classic and very tongue-in-cheek newsgroup and Internet etiquette document at <http://www.templetons.com/brad/emily.html>.

## I Found A Bug, What Should I Do?

Make sure you have the latest GlucoTools version, downloaded from <http://glucotools.sourceforge.net>. Review the open and closed bugs under the “Bugs” link at <http://sourceforge.net/projects/glucotools>. If you don't see the bug you've uncovered, open a new one by clicking on the “Bugs” link and then clicking “Submit New”. Include the information below in the “Detailed Description” of the bug. You can also browse the open bugs and submit a new bug directly from the links provided on <http://glucotools.sourceforge.net>

1. From GlucoTools main screen, press the hardware scroll-up button to get information to be reported with the bug. This information includes the PalmOS version, ROM ID, GlucoTools software version, source file version, and database version.



2. The type of PDA (Visor Neo, Palm III, etc.).

3. Describe the sequence of steps that exposed the bug.
4. Provide the exact text displayed, if any, when the bug is exposed.
5. Any known work-around for this bug.

Normally someone will follow-up within a few days, but it could be longer depending on how many maintainers there are - hey, I like vacations too ☺.

## How I Can Help?

One of the best ways to help after becoming familiar with GlucoTools is to subscribe to the glucotools-users mailing list and review the archives. If someone posts a question and you know the answer, please answer it.

If you are a developer and would like to contribute, review the *Feature Creep* section for things that are in the queue to do. Review the glucotools-devel archive for work that is being done by others. Then subscribe and post a message to glucotools-devel list at <https://lists.sourceforge.net/lists/listinfo/glucotools-devel> indicating what item you plan to start working on and an estimated completion time.

# Feature Creep

---

Feature creep, also called creeping featurism, is best defined at [http://info.astrian.net/jargon/terms/c/creeping\\_featurism.html](http://info.astrian.net/jargon/terms/c/creeping_featurism.html)

Feature creep describes a systematic tendency to load more chrome and features onto systems at the expense of whatever elegance they may have possessed when originally designed. More generally, the tendency for anything complicated to become even more complicated because people keep saying "Gee, it would be even better if it had this feature too". The result is usually a patchwork because it grew one ad-hoc step at a time, rather than being planned. Planning is a lot of work, but it's easy to add just one extra little feature to help someone ... and then another ... and another...

To prevent feature creep, I created this section to list the "Gee, it would be better..." items that I thought of during the development cycle. I hope the features in this list will appear in a future version. Each has a priority associated with it, with a priority of 1 being the most important.

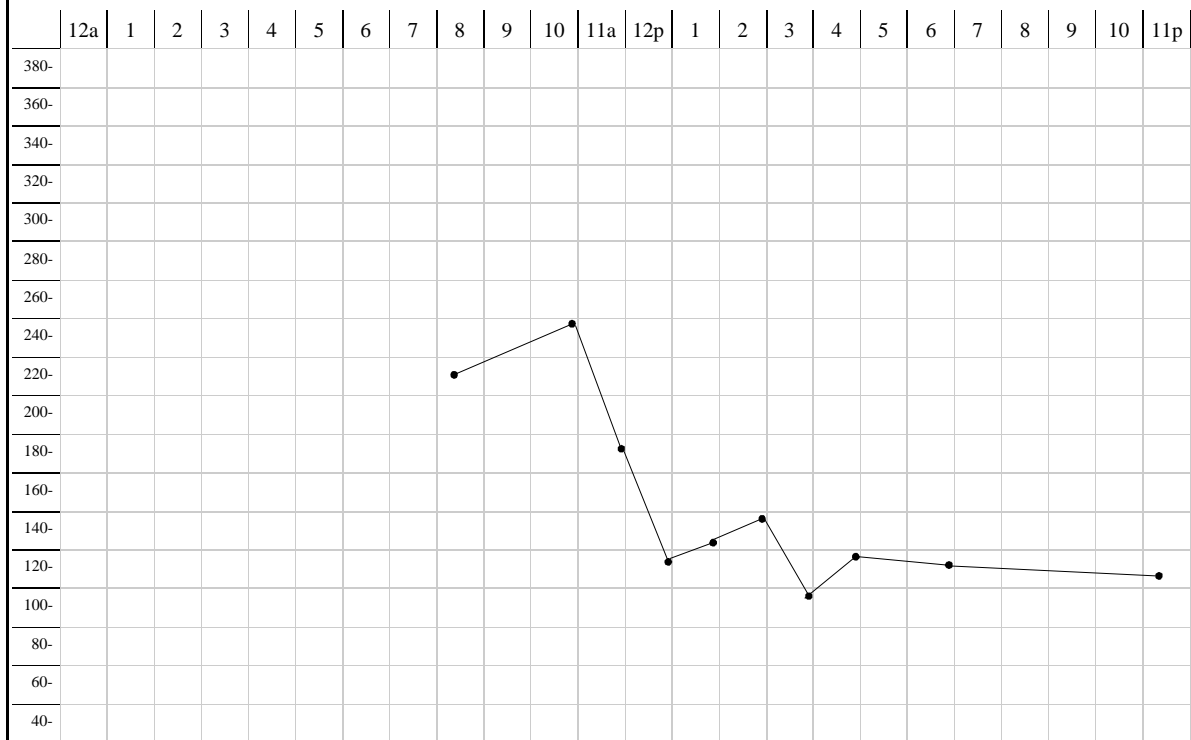
1. An JavaScript program to read the various GlucoTools databases (GTLS-BH.pdb - bolus history records, GTLS-MB.pdb - meal bolus records, GTLS-CB.pdb - correction bolus records), and display a chart similar to the one below. The program should allow the basal rates to be entered and stored in some fashion. Below is a sample of what a page might look like - notice the graph is to scale.

The BG value on the right of the table puts the point centered in the box for the associated value. That is, the BG reading at 8:30AM of 220 puts the point centered in the square. Notice the graph reflects the actual time, 8:30AM while the chart lists only the hours and has no way of listing the minutes after the hour. A way to integrate the boluses and basal rates into the graph needs to be investigated.

In the table below:

BG	= Blood Glucose
Carb	= Carbohydrates
MB	= Meal Boluses
CB	= Correction Bolus
BR	= Basal Rate

[Priority 1, I'm working on this now...]



	12a	1	2	3	4	5	6	7	8	9	10	11a	12p	1	2	3	4	5	6	7	8	9	10	11p	
BG									220			248	182	127	131	145	108	128							116
Carb									52					22											86
MB									10.4					5.2											26
CB									5.5																0.3
BR	0.4			1.0			1.1		1.6		1.5	0.9		0.8			0.7			0.9				0.5	

2. In the setup->preferences window, allow the number of days to be defined for the all average computations. A value of zero would cause all history records to be used in the computations as is currently done. [Priority 4, Completed in version 2.0]
3. Add the capability to interface to an existing food carbohydrate database - nothing fancy, maybe A-to-Z alphabetical grouping buttons with a search capability. This database wouldn't be required for GlucoTools to run, if it exists it would be used. [Priority 3]
4. Allow correction bolus to be scaled by a percent over four even intervals of the correction bolus duration. This scaling could allow an adjusted correction bolus to be computed. For example, if 1 unit lowers an individual's blood glucose by 20 points, the bolus duration is 4 hours, the the correction bolus is divided evenly over the bolus duration (25% per hour), the blood glucose target is 110, the last bolus was administered 3 hours ago for a reading 80 points above the target, and the individual's current blood glucose is 150, the following would be the scaled glucose value for which the bolus would be computed:

$(190-110) = 80$  points above target four hours ago

$80 / 20 = 4$  units correction bolus given four hours ago

$4 * 0.25 = 1$  unit remaining that will lower blood glucose by 20 points over the next hour

$150 - 20 = 130$  points to base the current correction bolus on

So, instead of calculating the current bolus for a blood glucose reading of 150, the correction bolus would be calculated on an adjusted value of 130. This accounts for the 1 unit of insulin that will reduce the blood glucose by an additional 20 points over the next hour.

OK, that is the simple case... this algorithm would also need to take into consideration multiple correction boluses. For example, consider a bolus duration of 5 hours, a correction bolus was administered 4 hours ago and another just 2 hours ago - this substantially complicates the algorithm. [Priority 9]

5. Provide an option to beep if a correction bolus warning is displayed. [Priority 8]
6. Add a basal test feature. After one entered the initial BG reading, an alarm would automatically be scheduled 2 hours (configurable) later to test the BG again. A suggested basal rate increase/decrease could be computed based on the insulin to BG ratio for the time the initial reading was taken to be discussed with ones physician. If the second BG reading is in range, the user would be given the opportunity to scheduled a third BG reading 2 hours later - if the reading was out of range, this third reading would not be available. The user would be given the option to schedule another reading as long as the current reading was in range. If the test started on 8am one day, the could be started at 9am on another day to level out the hours in between if needed.

Actually, this would probably be more trouble than it's worth. It would increase the program size and may not be of much use... I leave it here just for pondering. [Priority none]

7. Add min/max BG reading to one of the tools screens, also include the date and time of those readings. [Priority 2, Completed in version 2.0]
8. Add check box in setup preferences screen that causes Correction Carbohydrates to be added to a previous BG reading if the reading was less than five minutes ago. This would simply merge the carbohydrates with the previous BG record if it was less than five minutes old. [Priority 2]
9. Comment function headers using format for Doxygen (<http://www.doxygen.org/index.html>) - the code is well documented except for the actual functions themselves. [Priority 9]