Overall Kiosk and POS Program Functionality

Introduction

CAT produces software for many types of applications and industries. This document will try to give a somewhat brief overview of the capabilities we currently have for use in point-of-sale terminals, kiosks, and ATM's. The descriptions will all refer to use in kiosks, but the software modules can be used in many similar applications.

Windows

The kiosk powers up, runs, is serviced, and is powered down without any direct interaction with Windows. We do not run the standard windows user shell, the kiosks run our Looper program. It has one job – to make sure the kiosk program is running, and restart it if it is not.

All functions needed for operation and maintenance of the kiosk are provided *within* the program with consistent, easy to use screens.

Highly Refined User Paths and Cash Management

We have produced and refined the exact sequences we send users along depending on the outcome of other actions. When things go well, kiosk software is easy. But if someone enters cash, the transaction fails to post, and the cash dispenser runs out when refunding their payment and the printer is out of paper to print out an explanatory receipt – things get complicated. What if you don't get a sever reply one way or the other – did the payment post? Should the kiosk try again?

We have logic and functionality that has been developed over many years that can handle any situation that might arise. For example, if the receipt printer is out of paper, after getting customer approval we will process their transaction and email the receipt to a special customer service account. If the email is down, the message will be put into our internal queue and email will be re-tried every five minutes till the queue can be sent.

We also have a cash-flow technique that assures no matter what path a customer might take through the screens, cash in/out/posted will be tracked independently to assure no funds are lost or given away.

Security

There is no direct access allowed to the kiosk, all incoming ports are blocked. All requests must be one of the standard (limited) requests and is passed to the kiosk via a secure FTP server with 1024 bit encryption and signing.

Service Keys

A Service Key is inserted into the kiosk to place the kiosk into a service mode. The key identifies the owner by name, and also by job function. If you are a Service Technician, you will be shown the Control Panel. If a Cash Key is inserted by a store employee, they will be shown the Mini-Control Panel. If an Armored Car Company employee inserts an ACS Key, they will get an automated processing of a cash pickup and delivery.

Automated Functions

Many functions are completely automated in the software. A few examples: The kiosks upload status reports every five minutes, they upload log files every night, archive the local log file. They manage any OEM driver generated log files. They can be set to automatically reboot every certain night of the week.

Customer Graphical Interface

The customer GUI is the tiny tip of the iceberg for the SW. It must be written well, but more software is involved in properly processing the customer inputs. The majority of the software is involved in the maintenance, repair, and automated functions of the kiosk

The screens the customer interacts with are not "cookie cutter" or HTML, but Intelligent Data Entry Screens written in VB. This allows us to do just about anything we wish for any screen.

All screens feature multi-lingual support with the ability to support an almost unlimited number of languages. All text seen by a customer on the screen or receipts is table driven. This also allows the exact words used to be altered at any time without recompiling the program – the text is all loaded from a spreadsheet file and only that file need be altered.

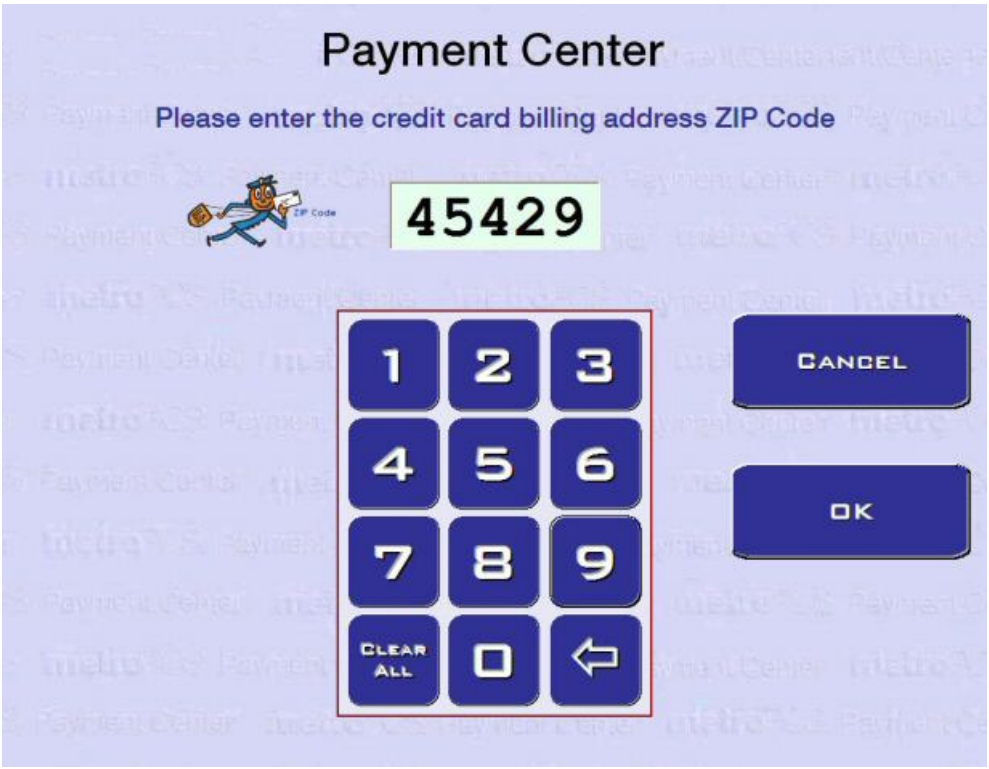
Rather than overwhelm you with every screen we have ever created, I will show just a few to show typical screens. The examples are from a kiosk that was clean yet a bit plain – any "look" needed can be produced.

The first example is a log-in screen for phone customers.

All the buttons are fully reactive – they change in color and appear to depress.

This (and all) data entry screens can be fully reactive, such as setting the background of the data entry box green when a proper number is entered, and only then is the *Push To Access Your Account* button displayed.

The "We Accept:" section is live, and will not display payment methods that are not available, such as when the cash acceptor is allowed to become full.



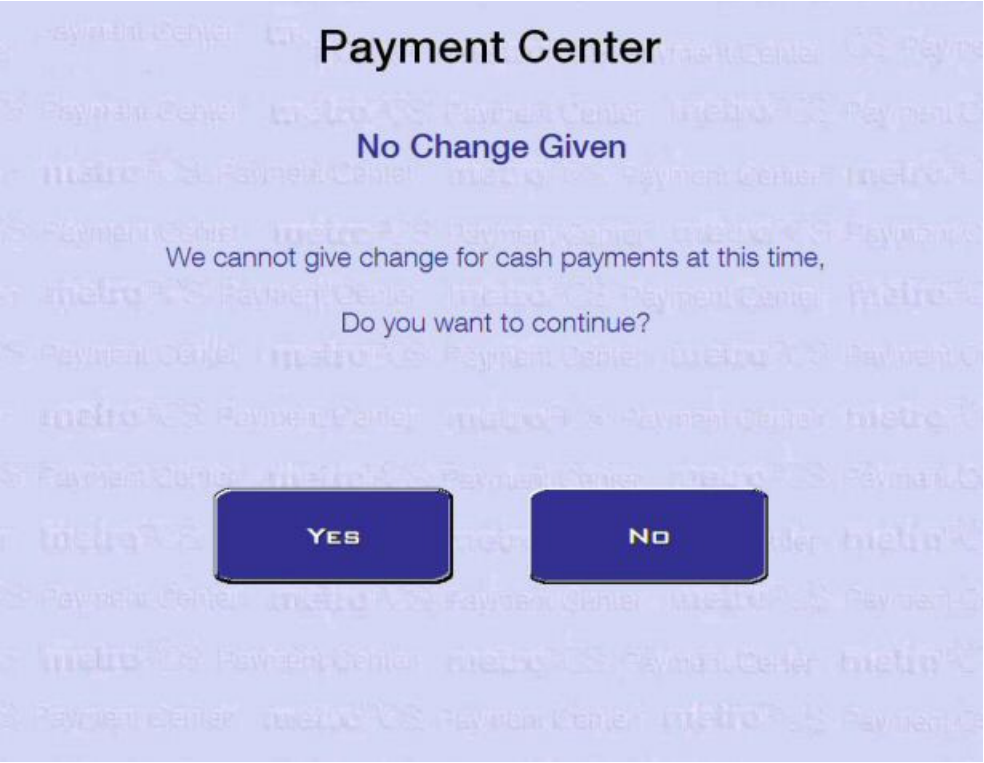
Our standard data input screen can be customized for a specific function. Again, the input box changed from white to green when the fifth digit was entered.

Some user input screens use graphical icon buttons when they help, not hinder. As all buttons do, these all react and change color when depressed.

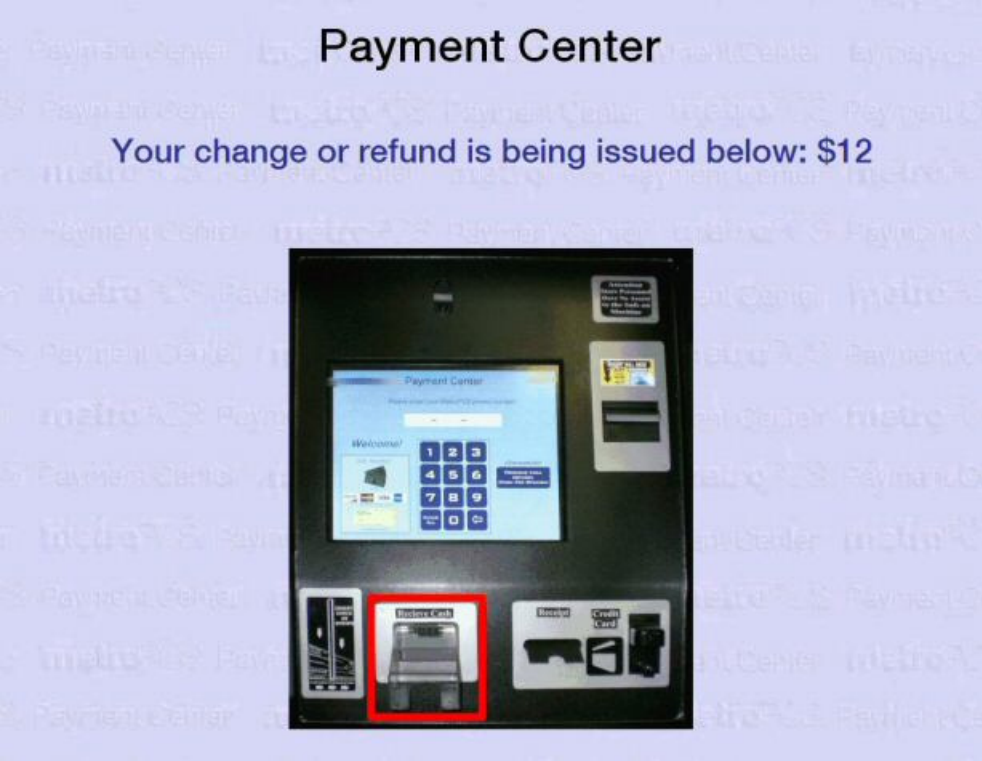
Buttons of any size and shape can be provided, including buttons that react different when different defined regions of them are touched.

We can be completely flexible on entry screens to get exactly what the client desires for a screen. This screen is multilingual, as are all user screens.

We can provide animations, video with or without sound, audio prompts, help buttons that run a help video, etc.

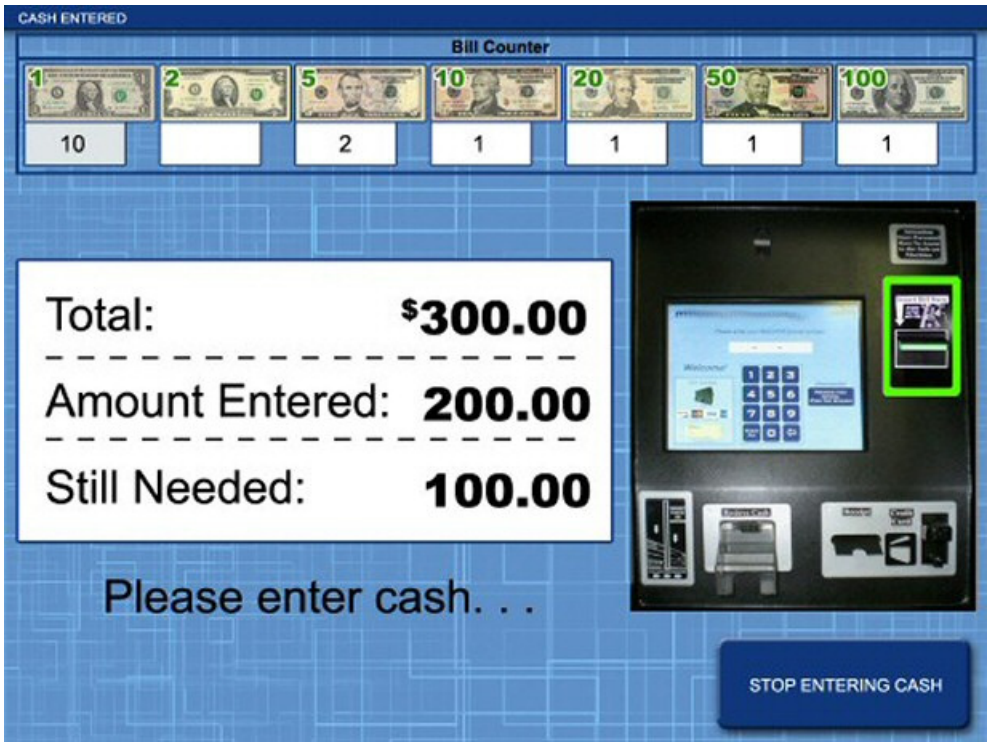


Most of the time this simple screen is called to ask the customer questions. It has a title, explanory text, and two buttons**. We like to stay with only two selections at a time to try and not overwhelm the customer.**



We have the usual screens to support each of the devices the user will interact with. This screen would be for the Cash Dispenser. The red box is animated, and is flashing. We can support flashing LED's physically next to the device, but so far we have found this on-screen method saves costs and allows greater flexibility.

Needless to say, the software running behind this screen to dispense cash is very complex. Many errors can occur with any dispenser, and they all must be dealt with properly.



The Cash Acceptor Module screen actually keeps track of how many bills of each type are entered. The prompt will change briefly to "Please wait…" as each bill is stacked. If a jam occurs, the automatic un-jamming process will begin and the prompt will turn to a red "Clearing Jam – Please Wait…" until the jam is clear.

This is an example of a module that performs very detailed logging to the Kiosk Log File when an option is enabled. Every change of state is logged, plus the actual RS-232 traffic is logged.

These are just a few sample user screens to look at. There are obviously many more, plus all screens for when things go wrong.

Kiosk Control Panel

The Kiosk Control Panel is one of our most important and powerful features, with development time and code volume many times that of the user interface screens.

As I said, we never let our Technicians go outside of our program; everything needed is in the program. This includes all the hardware support needed for testing and problem resolution.

The control panel has two screens, each of which requires a physical service key and one or both of two passwords.

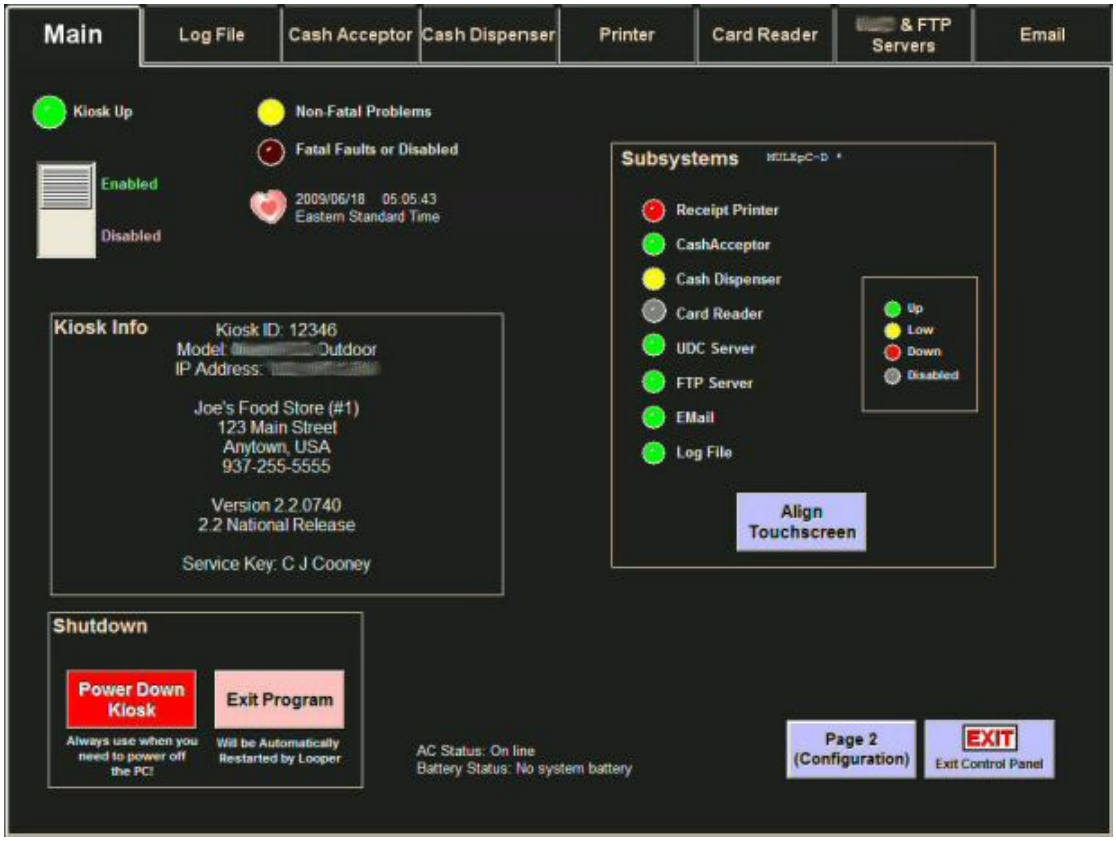
Page One is set up for normal day to day maintenance and testing. It allows a technician to monitor, test, reset, etc. each hardware device and each server that the kiosk accesses.

Page Two has all the configuration and setup functions for the kiosk. It requires a second password and is used rarely once a kiosk is up and running. Most configuration changes can also be made remotely via our secure request system.

It is important to understand the necessity of and the value of the control panel. If a test receipt is desired to be printed, just push a button – you don't have to log on and run a fake transaction to get a receipt to print.

The two control panel screens are each tabbed across the top, with a tab present for each device or service used on that kiosk.

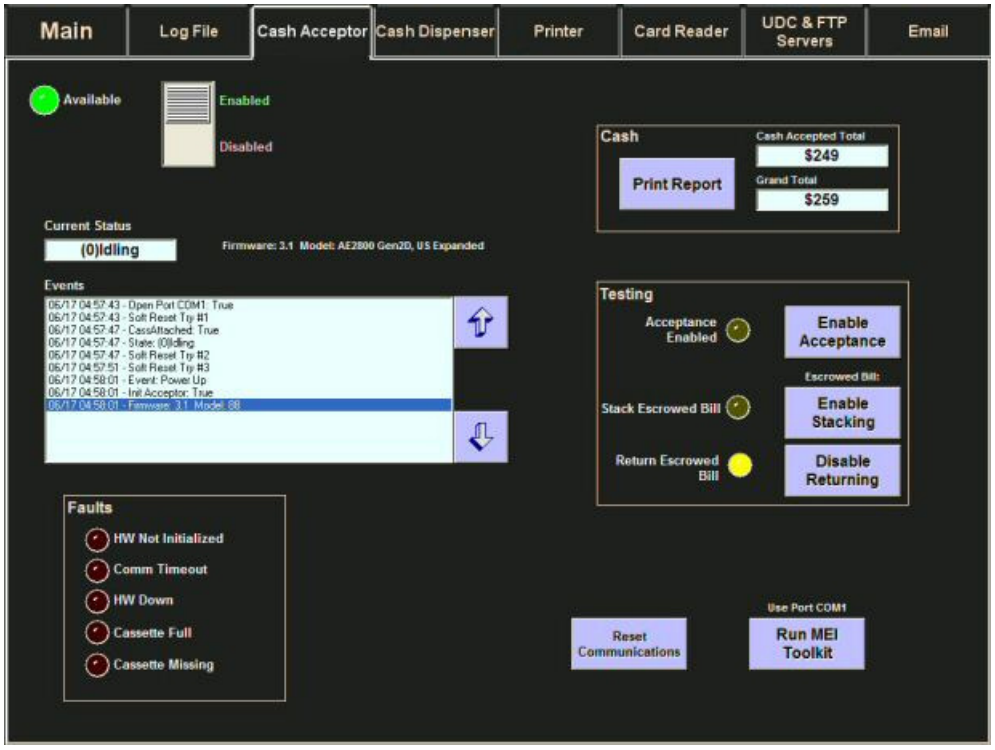
Page One



Page One opens with the Main tab, which gives an overall view of kiosk status. (I am not going to try and describe every function of this and other screens, just point out a few interesting details.) This shows the status of all the subsystems.

The heart shown actually "beats" every scan of the hardware, typically selected to be every five seconds. Our software doesn't wait till a customer is present and the hardware or a server link is needed to see if it is up – all subsystems are continually scanned to make sure they are communication and show a good status.

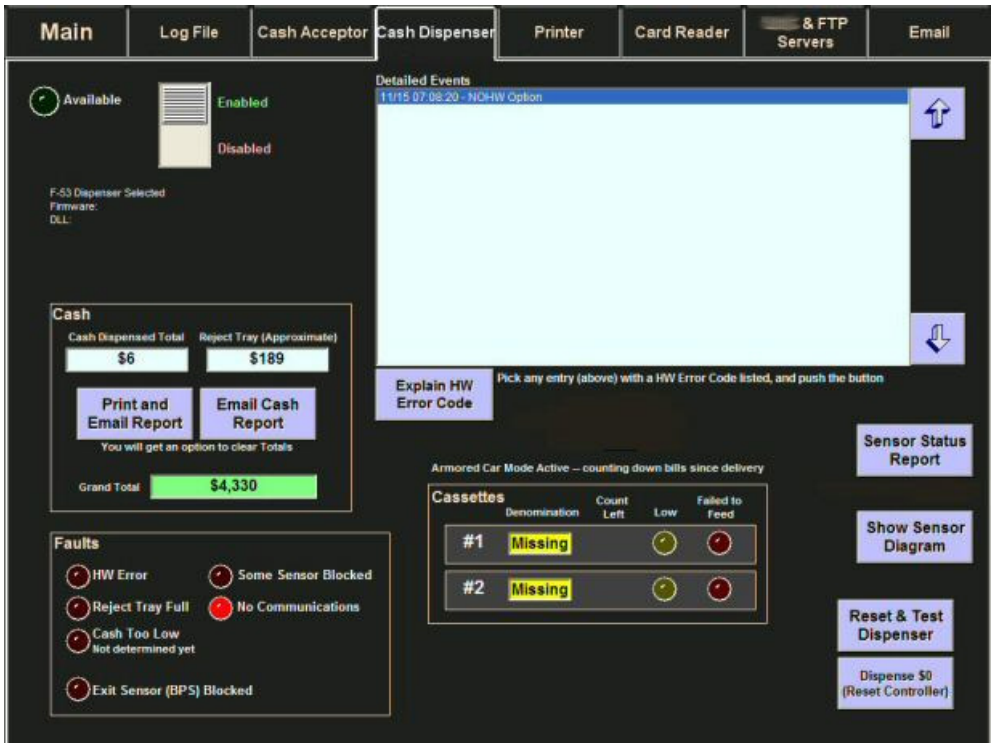
This screen also has the red button that will shut down the software and PC in an orderly sequence vs. just yanking PC power.



This is the Cash Acceptor tab on Page One. A detailed log of even the smallest events is kept here (only major events go into the Kiosk Log File). The faults are fully diagnosed and shown as red indicator lights. Complete testing of the acceptor functions can be performed.

Like all other tabs, an overall green Available indicator is in the upper left corner, and the acceptor can be disabled for any reason by sliding the switch to Disabled. (This can also be set remotely)

If it does become necessary to run the OEM supplied Toolkit, a push of the button will run the toolkit and turn the acceptor over to it.

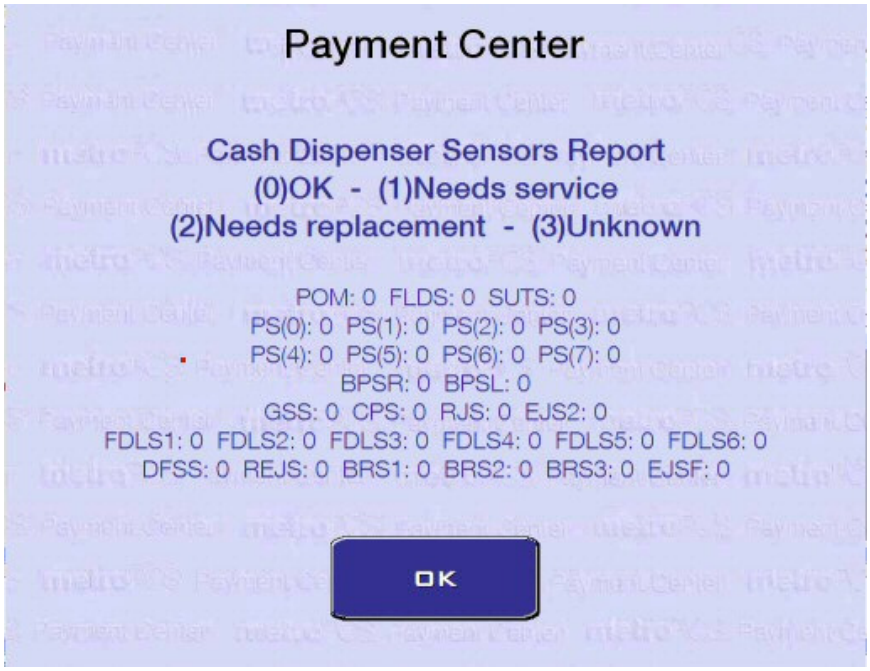


This is the tab for the cash acceptor. It is much like the previous screen in most functions. It automatically reconfigures itself for one through four cassette dispensers.

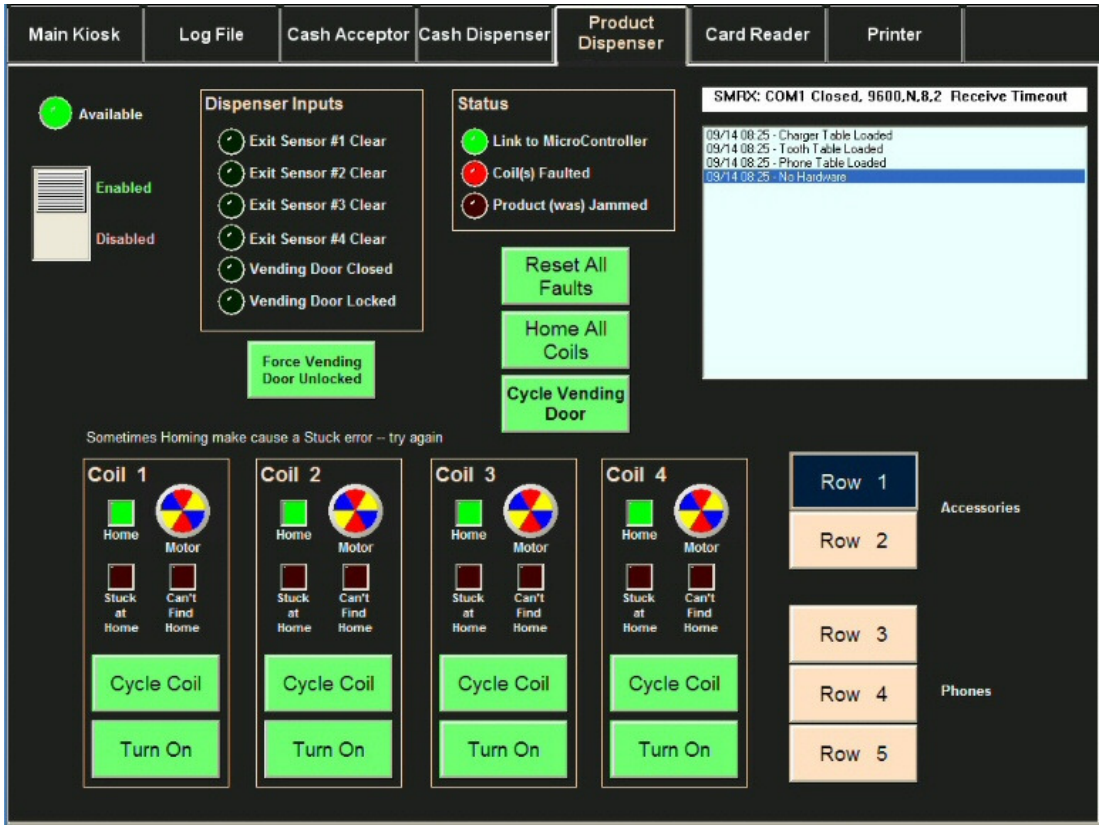
As with all other systems, the status of the dispenser is checked every "heartbeat" including the ability to make change of every possible amount required.

This tab gives direct access to many functions supplied by the OEM's software drivers. For example, any line that has a HW error code can have a screen like this displayed that explains the error code meaning and the recommended recover steps.

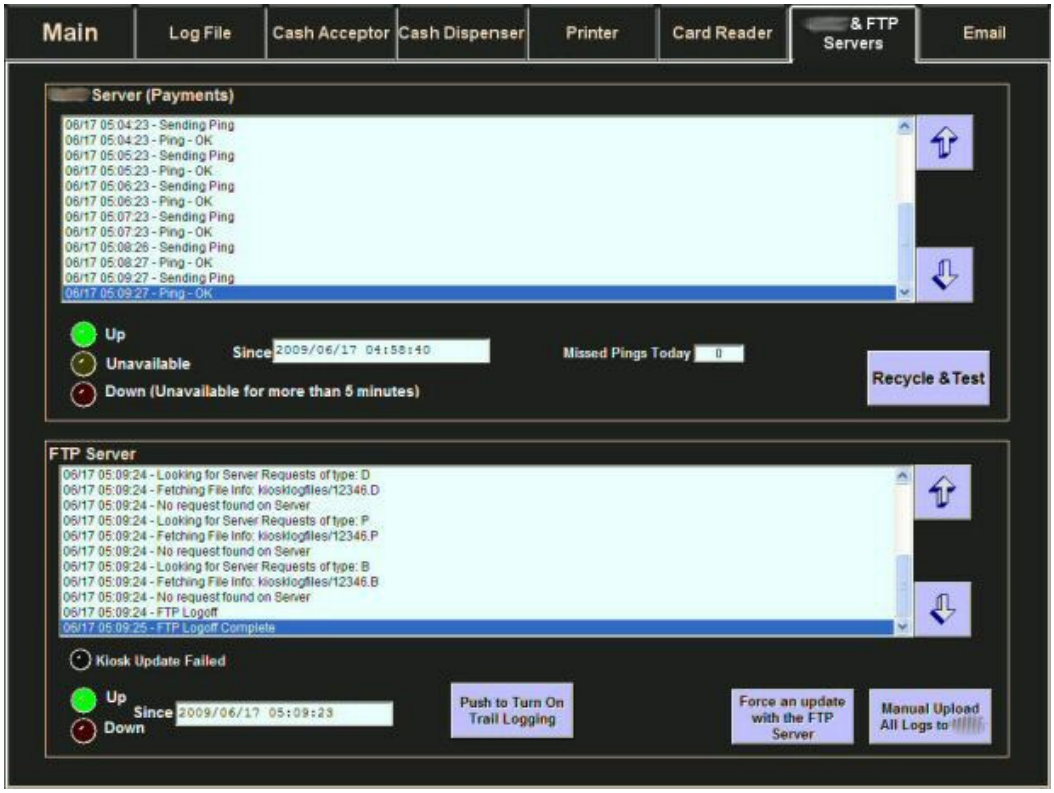
A sensor location diagram can also be popped up.



OEM driver supported diagnostics can be run from the control panel.



Here is a tab used to work with the Product Dispenser function we have on some kiosks.

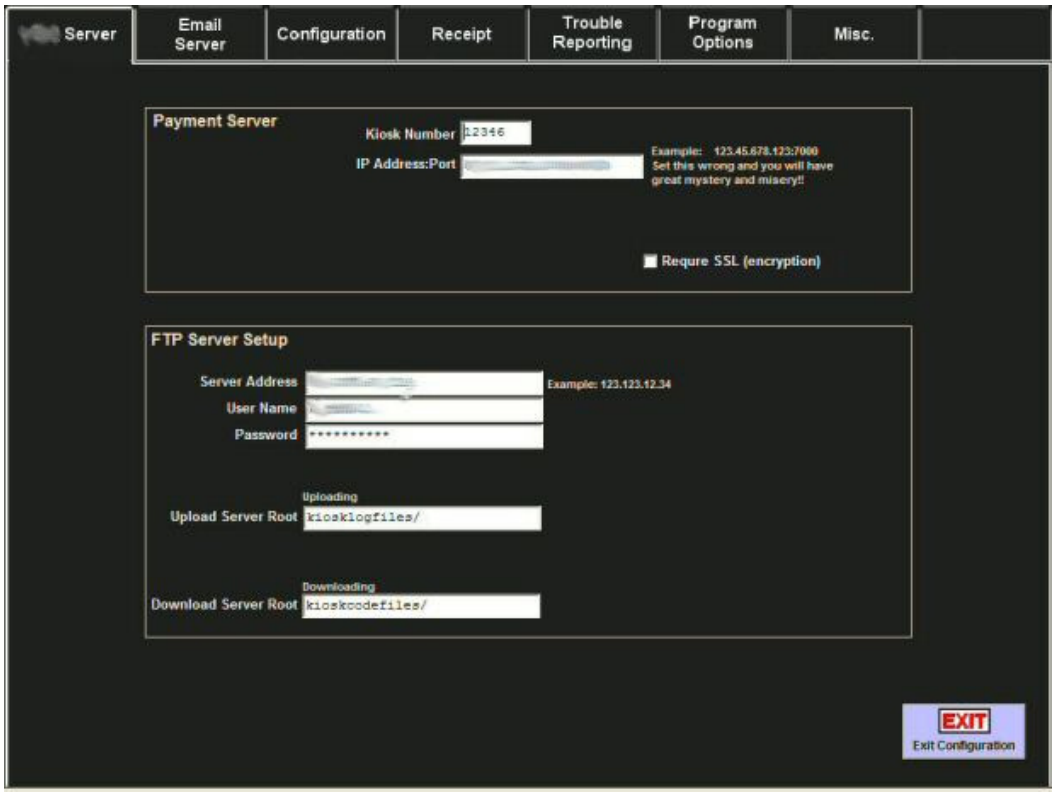


Links to servers on the Internet also need detailed monitoring and support for debugging. This tab displays the fine detail of what has been going on with servers, and also allows each server to be manually tested.

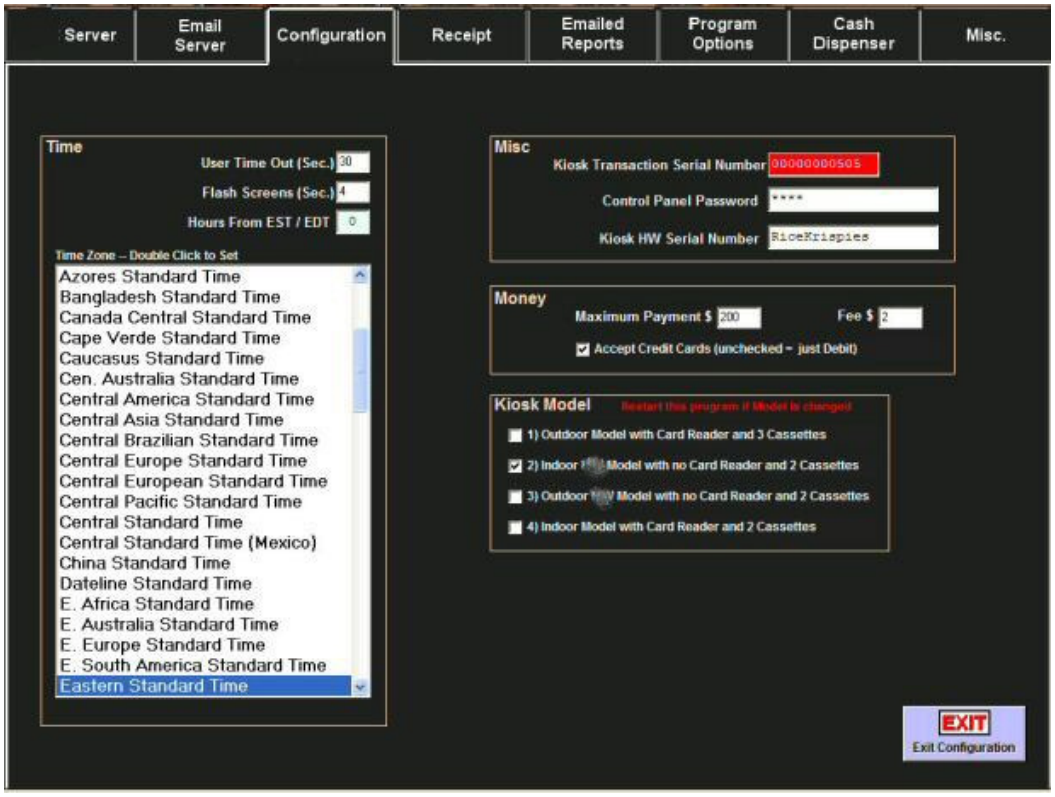
Kiosk Control Panel Page Two

Page Two is protected by a secondary password and contains the complete configuration for the kiosk. This can be reviewed or changed on these screens. Most changes can also be made remotely.

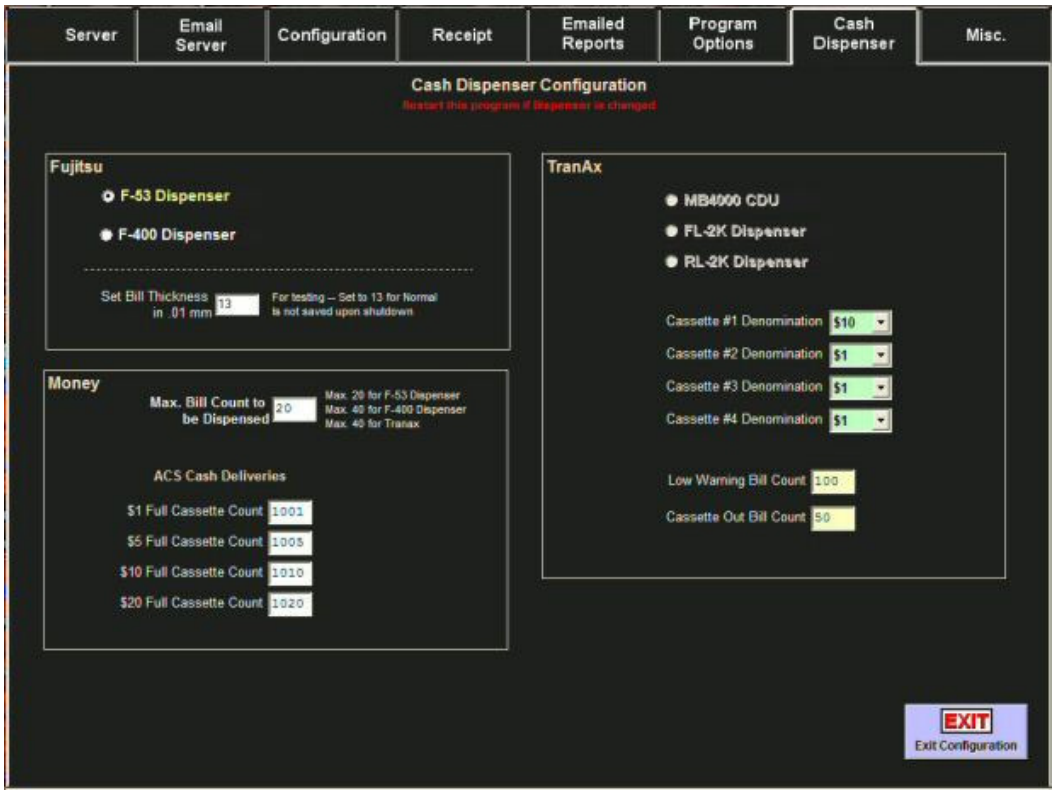
Most of these screens are self-explanatory. I'll show just a few as examples – tabs are added or removed depending on the kiosk's requirements.



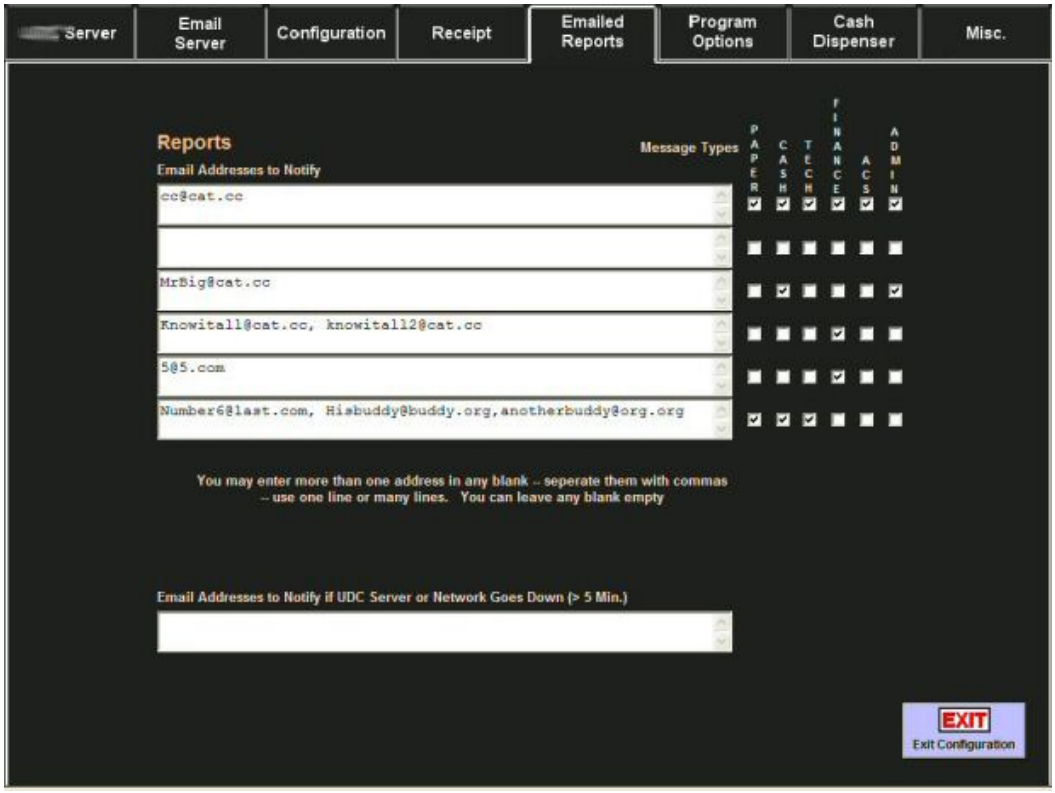
Server configuration.



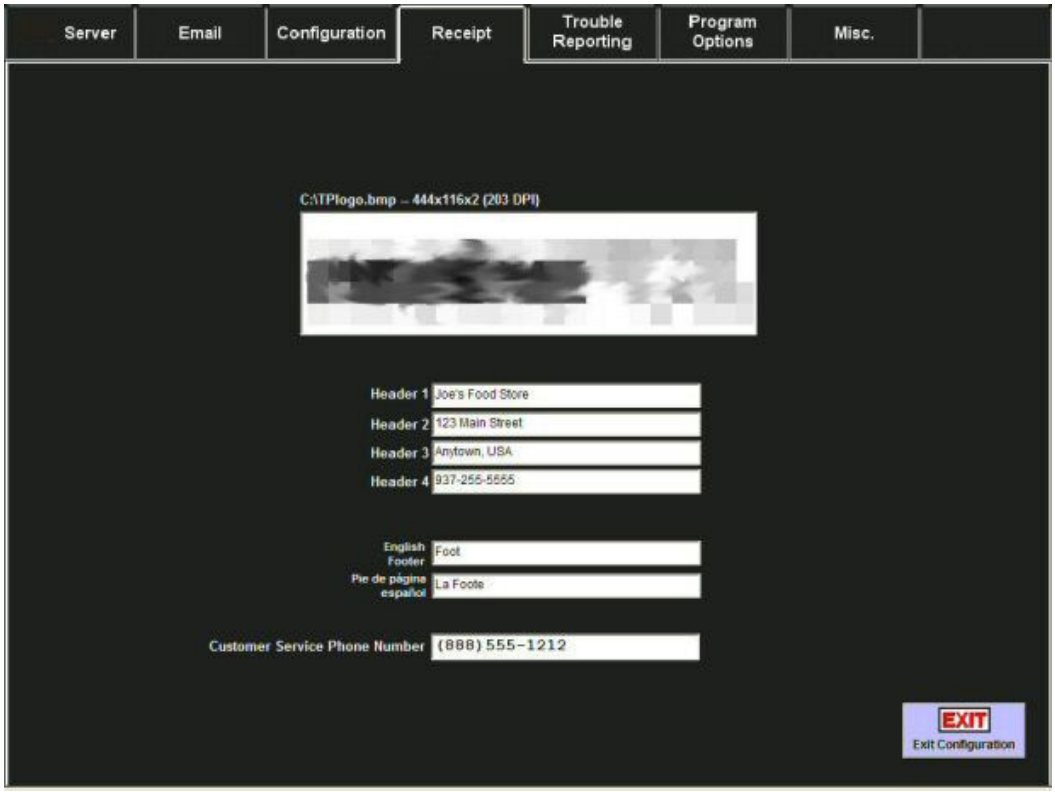
Misc. configuration.



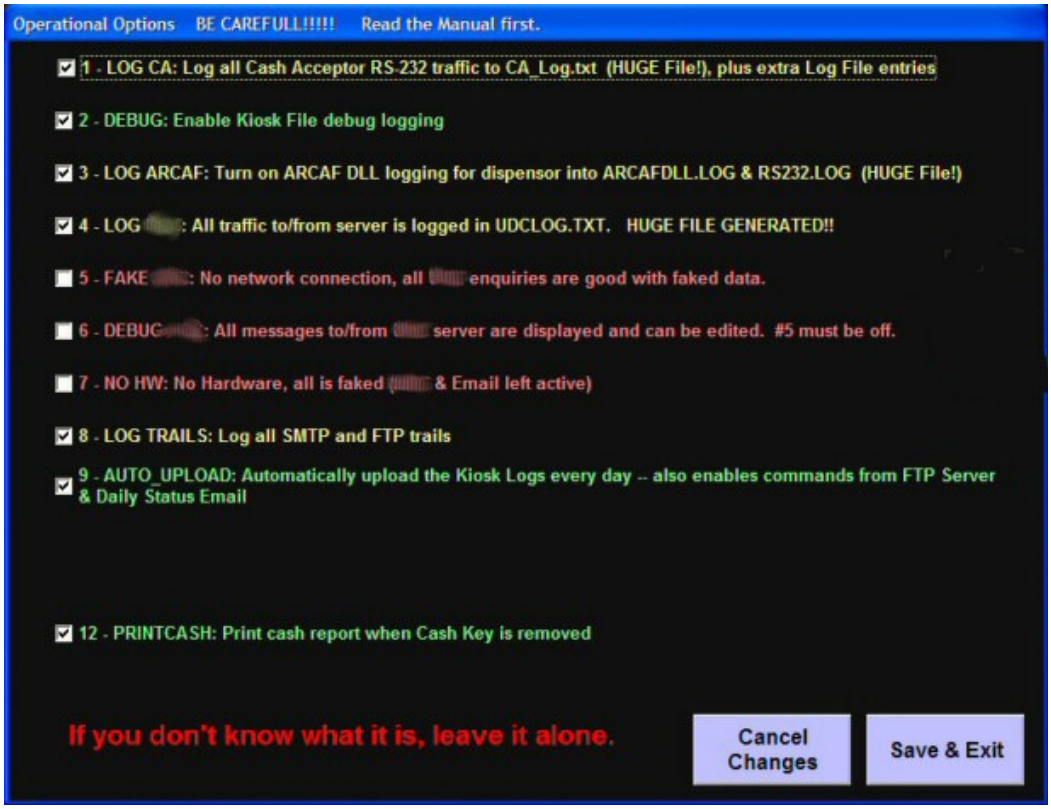
The same program can (currently) support five different Cash Dispenser models, and they can be changed in the field with the rest of the program reconfiguring for the changes. This also is where our Armored Car Service (ACS) Delivery System amounts are set. (These are typically changed remotely).



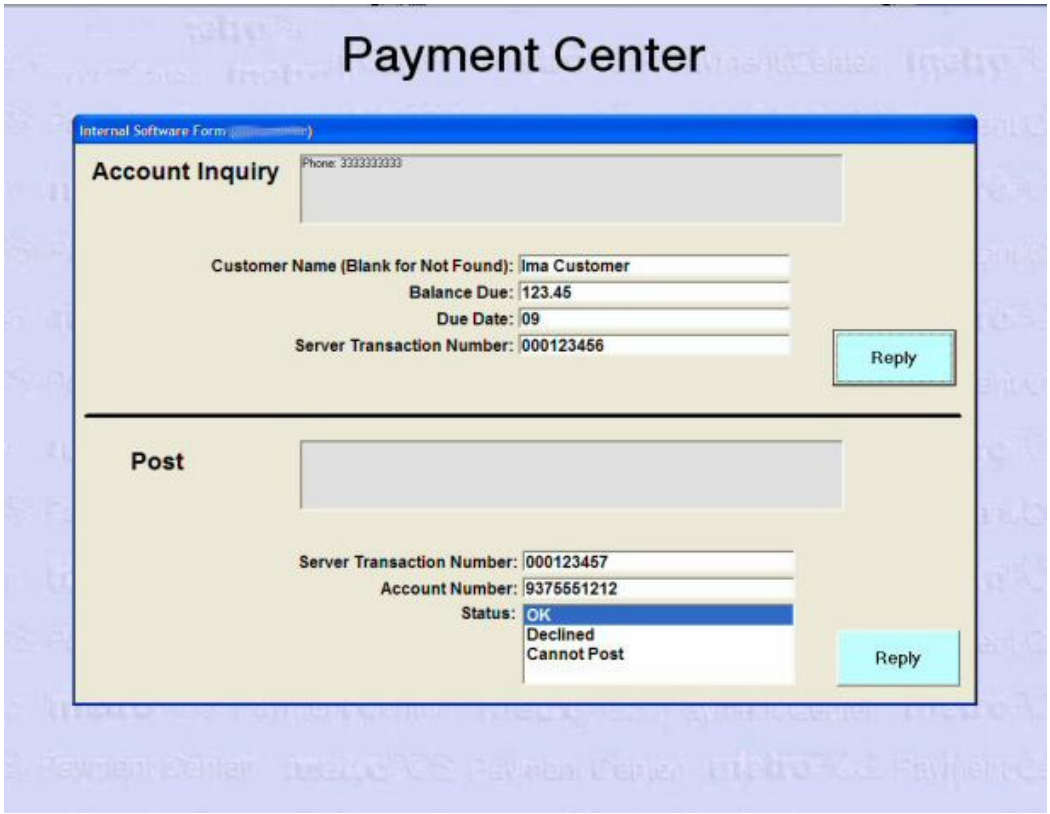
All email messages are tagged as being one or more of the following types: PAPER, CASH, TECH, FINANCIAL, ACS, and ADMIN. Using this screen, emails can be added to one of six groups and the types of messages each group desires can be set.



This is where the graphical logo and the text desired on the printed receipts are set.



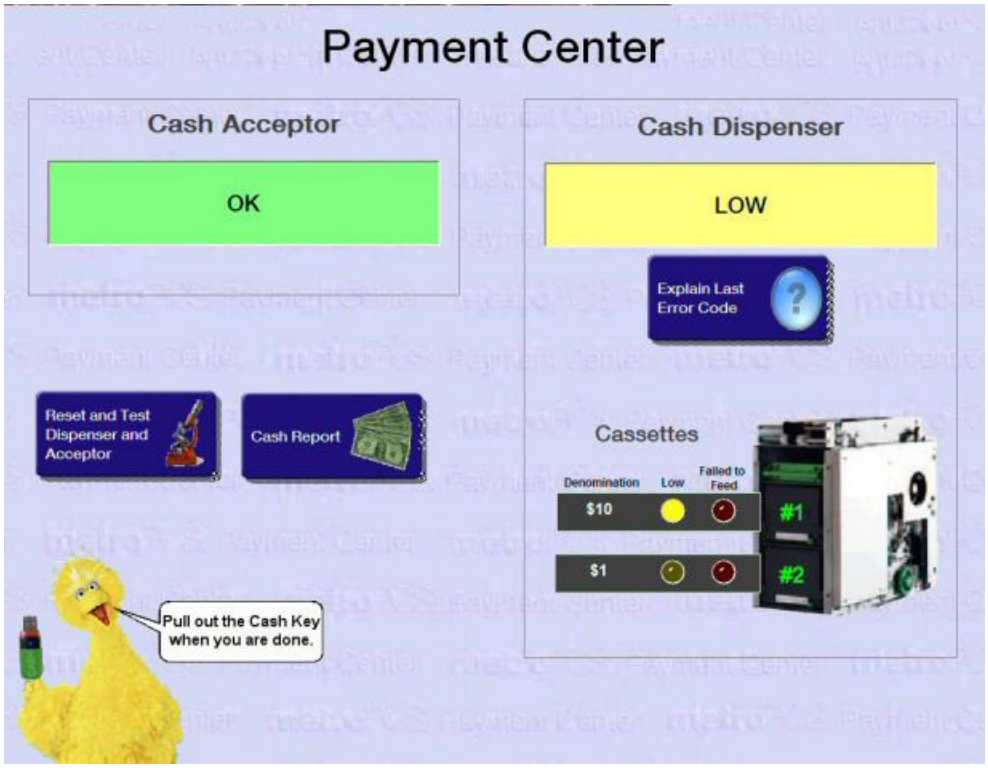
This is a very powerful screen that is used to set various program options. Extra log files written by OEM HW drivers can be turned on and off, the detail of logging for the Kiosk Log File can be set.



There are also important software development and debug functions. As an example, if Option #6 is turned on then this screen will be displayed every time the kiosk talks to the payment server.

The messages being sent to a server are intercepted before being sent to look at, and they can be hand edited before they are sent. The same with replies from a server – they are displayed before being sent to the program, and can be edited first if desired. (This is one of many software development and debug tools incorporated in the software.)

Mini Control Panel



When a customer or store staff will be servicing the cash devices, a mini control panel is supplied for their use. It is an extremely simplified version of the screens used by technicians, with single error messages and only a few buttons to push.

Emailed Reports and Messages

The kiosk can use emails for many purposes. The reports are in straight ASCII with the important information at the top so that they can also be used with text messaging. Some are used in conjunction with automated functions – managers can receive an email from each kiosk that is down at 4 AM after the daily log files upload.

The email below is typical of one sent out when a CASH problem occurs.

```
Cash Acceptor Status Report
Kiosk: Unavailable

  Acceptor Cassette FULL

=== mULEPc-dF ===
4/9/2010 5:20:13 PM (ET+0)

XXXXXX Kiosk  2.2.0857
#84633 (00000007692)
IP Address: 145.168.1.12
(2) Indoor XX Model

XXXXXXX
278 Easystreet Overpass
Bronx, NY, 10455
800-555-1212
```

Receipts

Joe's Cell Phones
222 Easystreet Overpass
Isles Of Langerhans, OH 45401
(800) 555-1212 (941) 228-1612

(666)555-1212
Joseph Ayre

Your account has been credited:

Monthly Bill:	$25.49
[illegible]:	$10.51
Fee:	$2.00
Total:	**$38.00**

Payment was made by cash.

Cash Entered:	$100.00
Change Returned:	$20.00

Thank you for your payment of

Because of a problem we could not dispense $42 cash owed you.

Please call us at the number on this receipt or visit any Company Store.

Your bill date each month is the 0th

Free Toaster with new phone!

78333-00000000484--0042
3/23/2008 3:47:27 AM

The receipts are fully graphical including a customer logo at the top. A receipt of almost any format can be produced, using graphics, lines, and different fonts (including high-security fonts).

The receipt generator is very smart and generates the proper receipt for any transaction, problem-free or not.

Kiosk Log File

The kiosk runs with one master log file. It is stored in the kiosk on the hard drive plus a permanent EAROM “thumb” USB drive. It is compressed and uploaded automatically every day along with all other log files being generated by OEM drivers.

It is quite wide, with a final field that can be of any length – all of these lines below are truncated to make them fit.

The file has fixed-width fields, and they are also delimited by “|” for easy import by either method.

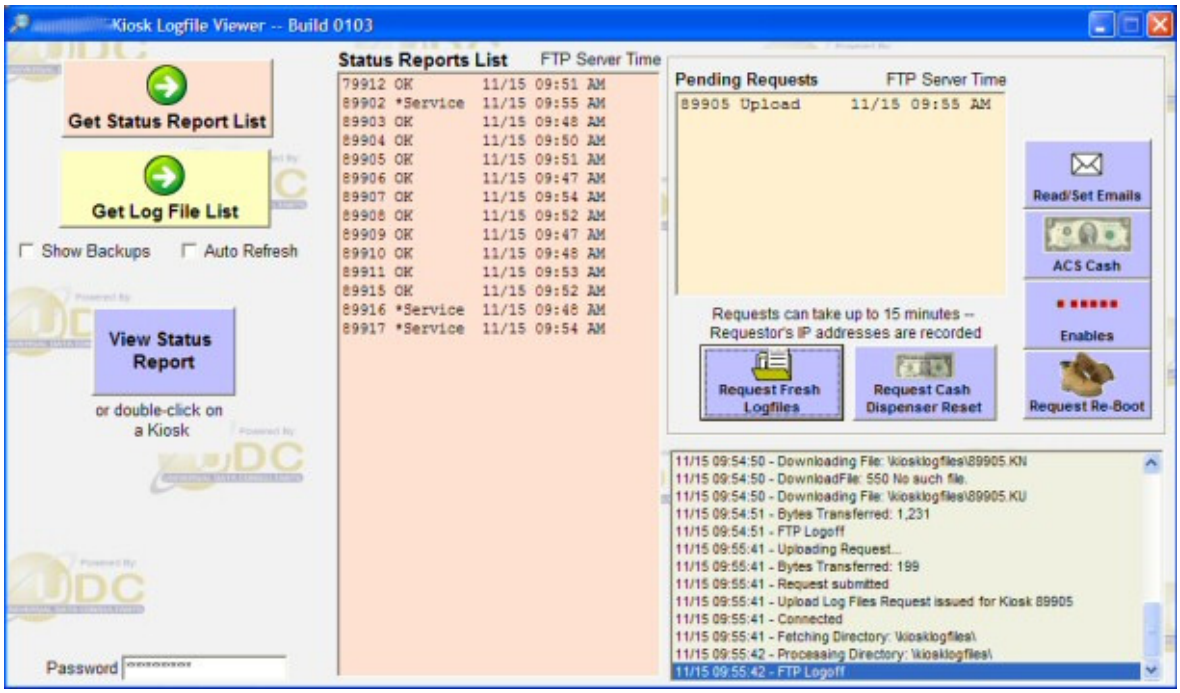
Each line is identified by the routine that generated it and the function being logged.

```
2010-07-13 11:51:43.56|00000001678|[NONE]    |muLEpcrdF|Logging         |Open            |===================|
2010-07-13 11:51:43.56|00000001678|[NONE]    |muLEpcrdF|Logging         |OpenLogFile     |LogLength          |0.1 MB
2010-07-13 11:51:43.56|00000001678|[NONE]    |muLEpcrdF|Logging         |Name            |metroPCS           |89917
2010-07-13 11:51:43.56|00000001678|[NONE]    |muLEpcrdF|XXXserver       |OpenXXXLogFiles |OK                 |
2010-07-13 11:51:43.59|00000001678|[NONE]    |muLEpcrdF|MainModule      |ReadKeys        |Service Key        |Present
2010-07-13 11:51:44.97|00000001678|[NONE]    |muLEpcrdF|ControlPanel    |Init1           |Complete           |
2010-07-13 11:51:44.97|00000001678|[NONE]    |muLEpcrdF|ControlPanel2   |Init            |Complete           |
2010-07-13 11:51:44.97|00000001678|[NONE]    |muLEpcrdF|ControlPanel    |Init            |Complete           |
2010-07-13 11:51:44.98|00000001678|[NONE]    |muLEpcrdF|Emailer         |ConfigureEmail  |OK
2010-07-13 11:51:44.98|00000001678|[NONE]    |muLEpcrdF|Emailer         |RestoreEmailQueu|OK                 |0
2010-07-13 11:51:49.38|00000001678|[NONE]    |muLEPcrdF|Sequences       |InitializePrinte|Printer
2010-07-13 11:51:49.38|00000001678|[NONE]    |muLEPcrdF|LPTdriver       |Open_Driver     |Failed
2010-07-13 11:51:49.55|00000001678|[NONE]    |muLEPcrdF|CashAcceptor    |OpenCAport      |COM1               |True
2010-07-13 11:51:49.56|00000001678|[NONE]    |muLEPcrdF|CashAcceptor    |CAsoftReset     |Soft Reset         |Try #1
2010-07-13 11:51:53.23|00000001678|[NONE]    |muLEPCrdF|CAsupport       |DecodeOMINreply |CassAttachedBit    |Present
2010-07-13 11:51:53.23|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |ProcessCAstatus |CashBoxAttached    |True
2010-07-13 11:51:53.23|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |ProcessCAstatus |State: (0)Idling
2010-07-13 11:51:53.23|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |CAsoftReset     |Soft Reset         |Try #2
2010-07-13 11:51:57.23|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |CAsoftReset     |Soft Reset         |Try #3
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |ProcessCAstatus |PowerUp Event
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CAsupport       |EnableCashAccept|False              |PowerUp
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CAsupport       |EnableCashAccept|False              |PowerUp
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |CAsoftReset     |Complete
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |InitAcceptor    |True
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CashAcceptor    |InitAcceptor    |Firmware:          |2.5
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CAsupport       |EnableCashAccept|False
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CAsupport       |EnableCashAccept|False            |
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|Sequences       |InitHW          |Cash Acceptor      |Init
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CDsupport       |LogCDcassettesSt|CD Cass#1          |UP
2010-07-13 11:52:05.69|00000001678|[NONE]    |muLEPCrdF|CDsupport       |LogCDcassettesSt|CD Cass#2          |UP
2010-07-13 11:52:06.00|00000001678|[NONE]    |muLEPCrdF|CDFujitsuSupport|FujiOpenDLL     |ARFDllInit         |OK
2010-07-13 11:52:06.00|00000001678|[NONE]    |muLEPCrdF|CDFujitsuSupport|FujiOpenDLL     |ARFDllInfo
2010-07-13 11:52:16.06|00000001678|[NONE]    |muLEPCrdF|CDFujitsuSupport|ProcDispDLLretur|Time Out           |0
2010-07-13 11:52:16.06|00000001678|[NONE]    |muLEPCrdF|CDFujitsuSupport|FujiUpdateDS    |Comm Timeout
2010-07-13 11:52:27.13|00000001678|[NONE]    |muLEPCrDF|CDFujitsuSupport|TestDispense    |ARFcancel          |Failed
2010-07-13 11:52:28.86|00000001678|[NONE]    |muLEPCrdF|Sequences       |InitHW          |Cash Dispenser     |Not
2010-07-13 11:52:28.89|00000001678|[NONE]    |muLEPCRdF|Sequences       |InitHW          |Card Reader
2010-07-13 11:52:28.89|00000001678|[NONE]    |muLEPCRdF|Sequences       |InitHW          |Complete
2010-07-13 11:52:29.91|00000001678|[NONE]    |muLEPCRdF|XXXserver       |Connect         |FAILED
2010-07-13 11:52:31.00|00000001678|[NONE]    |muLEPCRdF|XXXserver       |Connect         |FAILED
2010-07-13 11:52:35.20|00000001678|[NONE]    |muLEPCRdF|Sequences       |MrStatus        |Unavailable        |muLEPC
2010-07-13 11:52:35.50|00000001678|[NONE]    |muLEPCRdF|MainModule      |Main            |Startup Complete   |2.2.0862
2010-07-13 11:52:35.50|00000001678|[NONE]    |muLEPCRdF|Emailer         |SendEmail       |Called             |TECH SW
2010-07-13 11:52:35.50|00000001678|[NONE]    |muLEPCRdF|Emailer         |SendIt          |Queued             |TECH SW
2010-07-13 11:52:40.51|00000001678|[NONE]    |muLEPCRdF|Emailer         |TransmitEmailQue|Send Queue         |1
2010-07-13 11:52:40.51|00000001678|[NONE]    |muLEPCRdF|Emailer         |TransmitQueueSMT|Disconnected       |Safety
2010-07-13 11:52:40.51|00000001678|[NONE]    |muLEPCRdF|Emailer         |TransmitQueueSMT|Connecting
2010-07-13 11:52:40.51|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Info
2010-07-13 11:52:40.63|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Info
2010-07-13 11:52:41.01|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.01|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Client
2010-07-13 11:52:41.13|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.45|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Client
2010-07-13 11:52:41.66|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.66|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Client
2010-07-13 11:52:41.78|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
2010-07-13 11:52:41.78|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Client
2010-07-13 11:52:41.89|00000001678|[NONE]    |muLEPCRdF|EmailerForm     |SMTPctl Trail   |Server
```

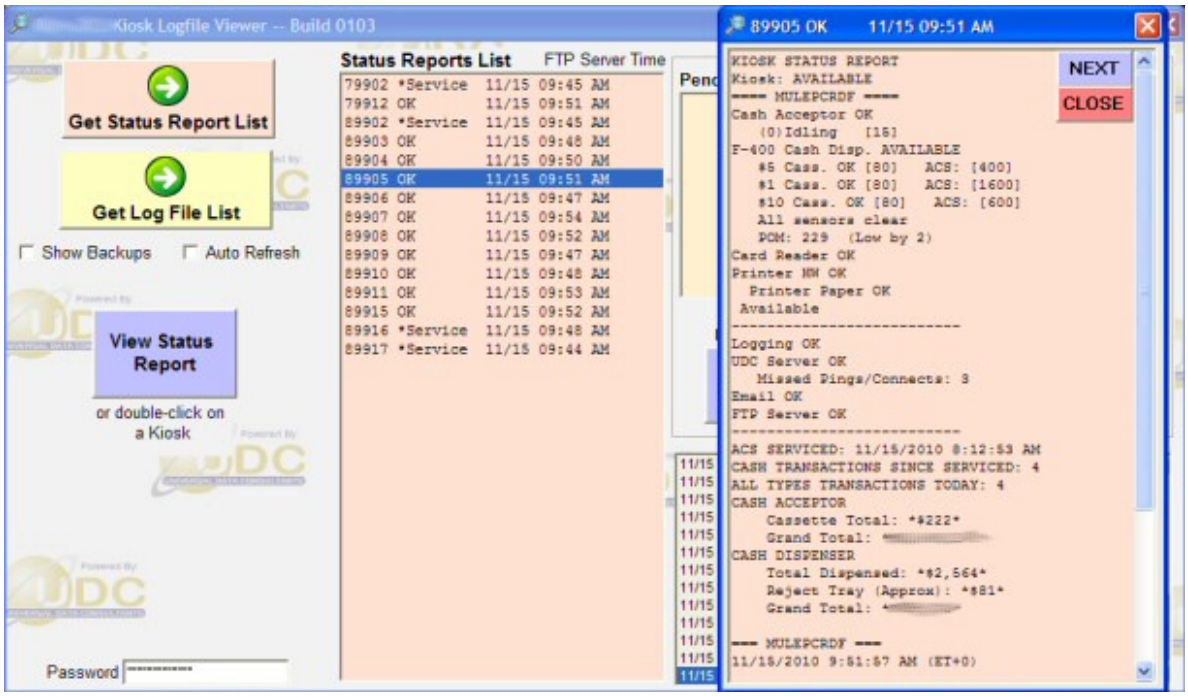
Log File Viewer

The LogFileViewer is s a separate application that runs on PC's used by Service Technicians, Financial Staff, Managers, etc. Different passwords allow different numbers of functions to be accessed.

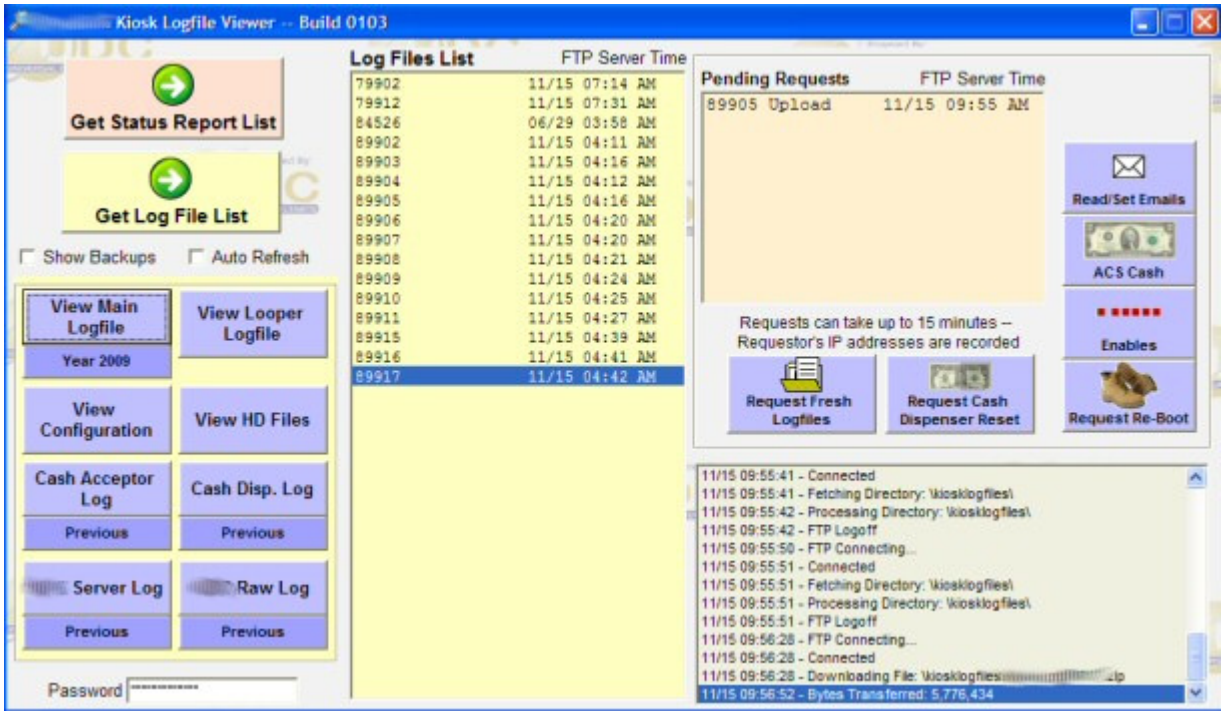
As previously explained in the Security section there is no direct communication to the kiosks. This program uses the Status Reports that are uploaded to the FTP server every five minutes, and Log Files that are uploaded every night.



The main Status Report screen shows which kiosks have reported in and when.



The last Status Report sent by any kiosk can be viewed. If run minimized and the status of any kiosk changes, the program will beep.

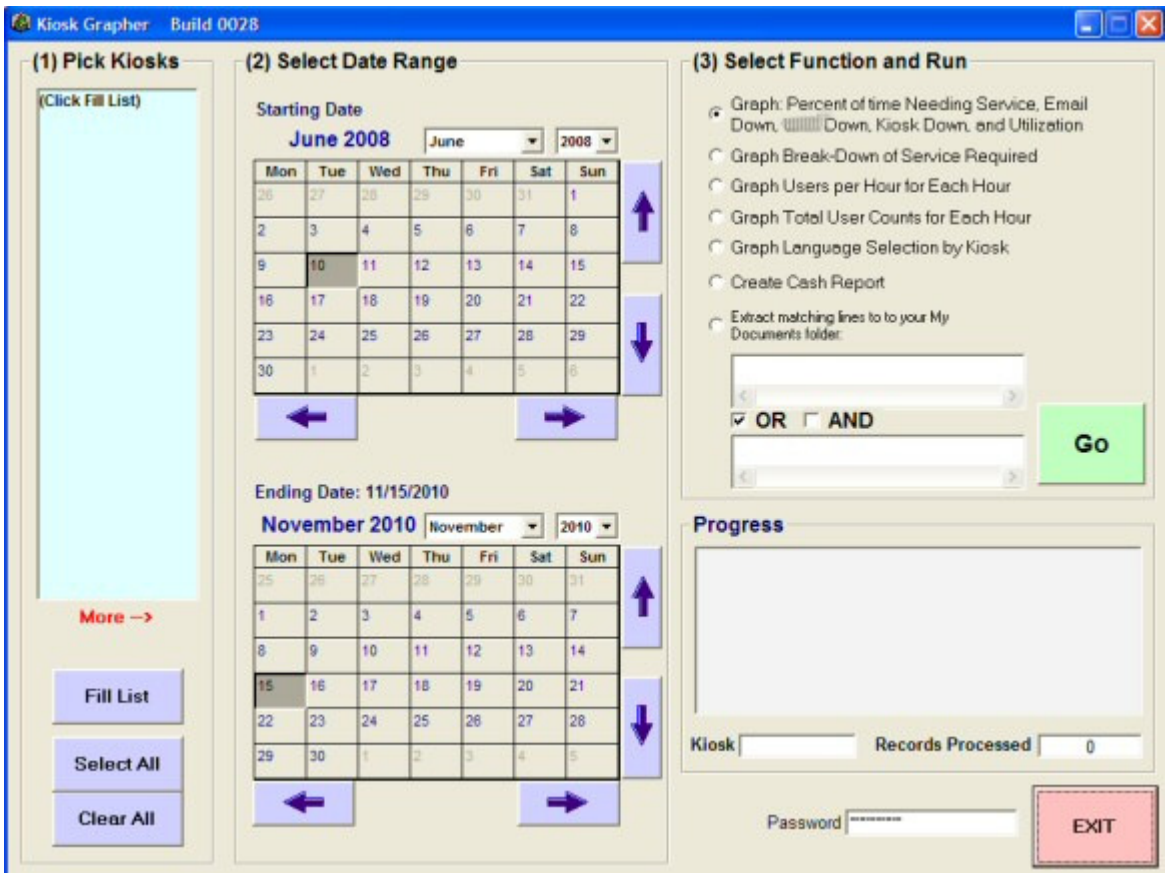


The Log File List screen shows which log files can be viewed.

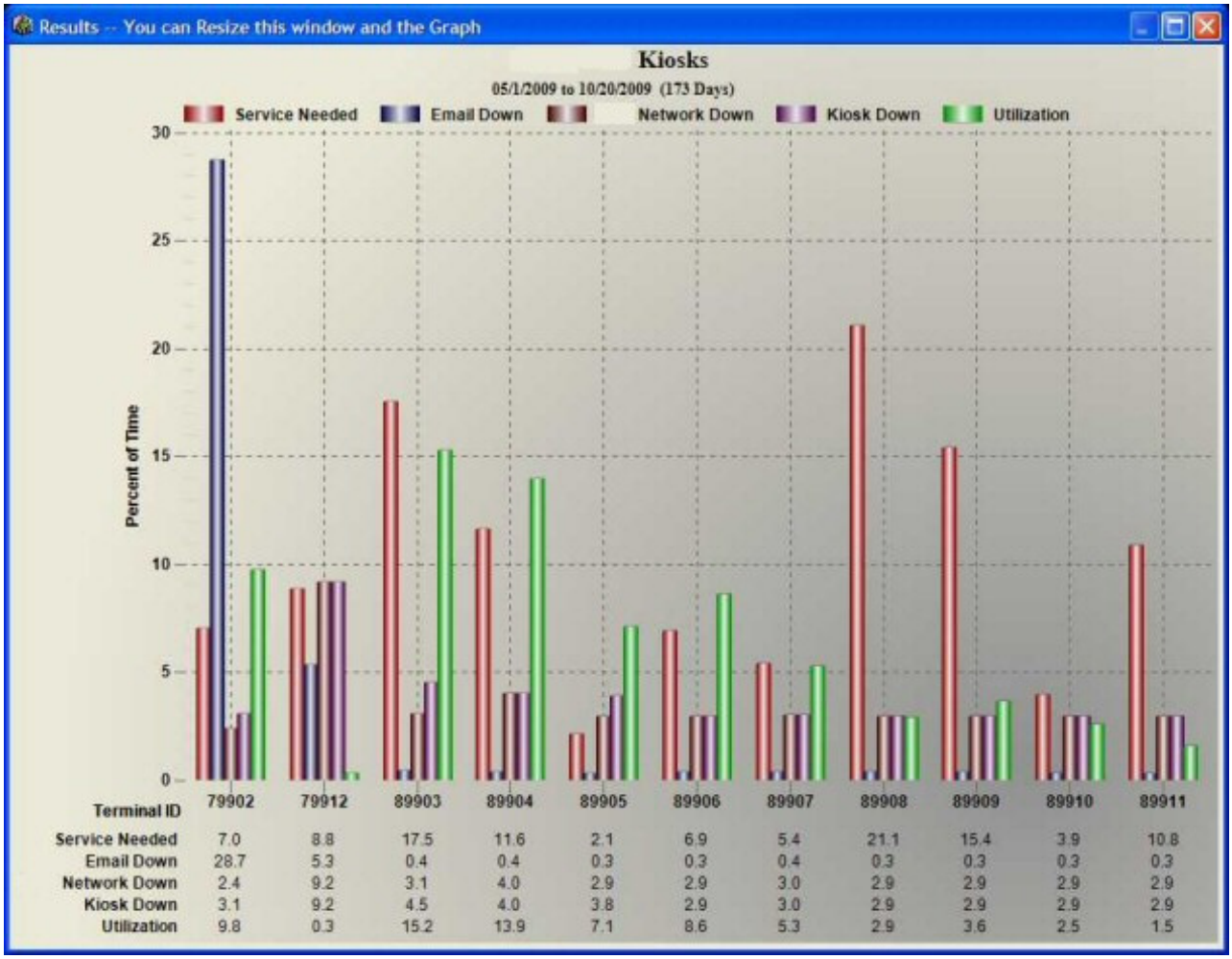
Other functions such as requesting fresh log files, changing email setup, etc. can be accessed by authorized users.

Kiosk Grapher

The Kiosk Log Files can be directly imported into programs such as Excel and Access. We have a program that allows the processing of common functions across multiple Kiosk Log Files, including file retrieval from the FTP server.

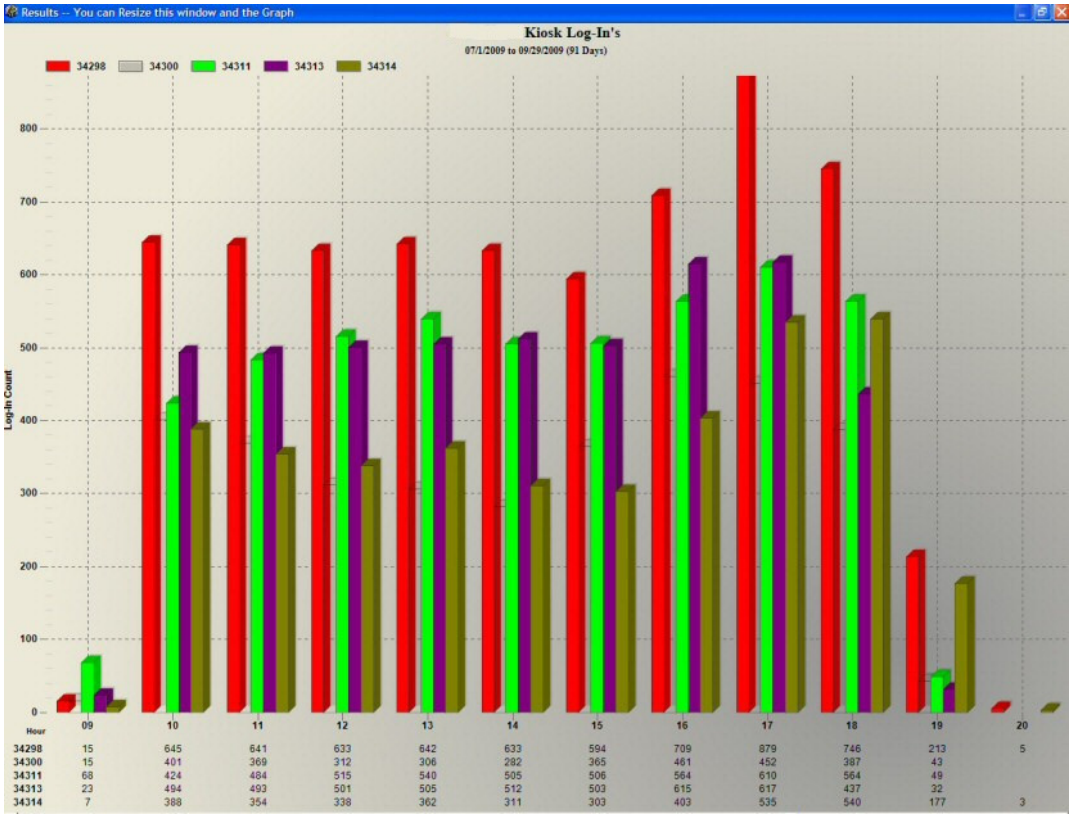


The kiosks are selected, the function selected, and either a file or graph is generated.

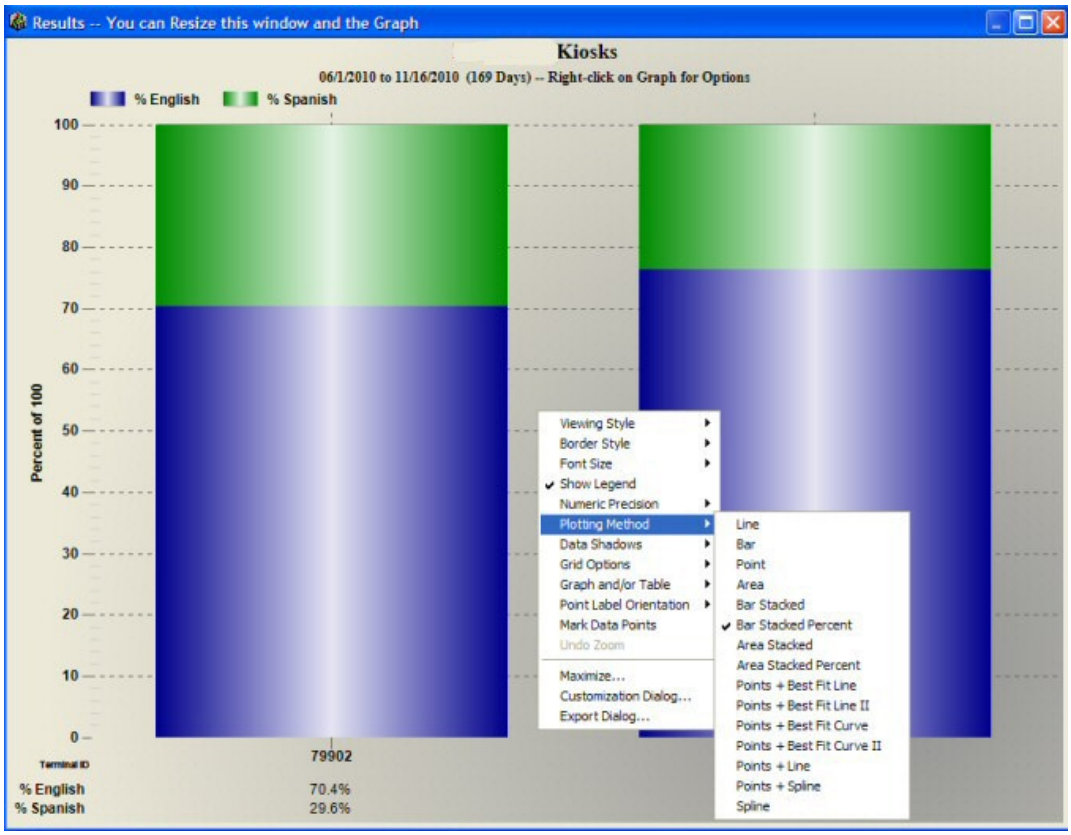


Terminal ID	79902	79912	89903	89904	89905	89906	89907	89908	89909	89910	89911
Service Needed	7.0	8.8	17.5	11.6	2.1	6.9	5.4	21.1	15.4	3.9	10.8
Email Down	28.7	5.3	0.4	0.4	0.3	0.3	0.4	0.3	0.3	0.3	0.3
Network Down	2.4	9.2	3.1	4.0	2.9	2.9	3.0	2.9	2.9	2.9	2.9
Kiosk Down	3.1	9.2	4.5	4.0	3.8	2.9	3.0	2.9	2.9	2.9	2.9
Utilization	9.8	0.3	15.2	13.9	7.1	8.6	5.3	2.9	3.6	2.5	1.5

This is the amount of time each kiosk spent in which state over the selected interval.



This graph shows utilization vs. the hour.



Each graph is dynamic; everything will resize with resizing the graph window.

Colors, plotting methods, titles, grids, etc. can be changed.

The graph image can easily be exported.

CAT Software

All source code is heavily documented and commented (for self-serving purposes). More lines have comments than not, all routines have full header comments detailing function, inputs, outputs, routines called. Routines interfacing with servers or OEM drivers all have the server message or device message format documentation incorporated into the comments.

We have been producing applications for 37 years and continue to support everything still in use. We have had high praise from customer who did acquire our source code.

I hope I managed to pass on the width and depth of the software we have at this time without supplying too much information. I'd be happy to discuss the details of anything we do or could do.

Chris J. Cooney

Engineering Manager
CA Tech, LLC.
P.O. Box 192039
Kettering, OH 45429-0039
(937) 294-0653
www.CAT.cc
cc@CAT.cc