Kitchen Screen

Installation & User Guide
V1.50

www.chromis.co.uk

Introduction

Welcome to the first of a number of POS based applications. The kitchen screen application can be used with any system which can present a table in a database, to which it has the ability to connect and contains the correct details.

This documentation is written with Chromis POS in mind. To use the application it is recommended that you are using the latest version of Chromis. The lastest version can be downloaded from

https://sourceforge.net/projects/chromispos/files/

The main order screen is updated at 5 second intervals, with data from the database. The main screen consists of 8 main order panels and a 'Order waiting' panel. The screen is populated in the sequence that the orders are passed to the table.

The user can have any number of screens attached to the database. These can all show the same orders or there can be up to 9 independent screens.

An example of this is.

Imagine a restaurant that has 4 preparation areas

- Bar
- Starter
- Mains
- Desserts

The system can be configured to allow the correct items to only be sent to the appropriate preparation area.

You have ability to control the display using either a touch screen, keyboard or bump bar. Using any of these options you will select completed orders and remove them from the display. As part of that you are now able to recall a previous order back into the system, the history of orders that can be recalled is configured in the system setup.

The POS applications must use a multi connection database, like MySQL or PostgreSQL.

System Requirements

In order to run the application, there are a few requirements.

- Any OS that supports JavaFx (Windows or Linux) or Raspberry Pi 2
- Java 1.8(u45) JRE
- Keyboard (only required for configuration)
- Mouse
- POS software
- Time in sync with the database server clock
- PDF reader for instructions (except Raspberry Pi)

The application will only work with a database engine that supports multi connections, it will not work with the default Unicenta database.

It is not intended to release a dedicated version for the Mac OS, unless there is a huge demand. The reason behind this logic is cost. The idea is to allow the user to implement a low cost order screen and Mac's do not come into the cheap category.

Table details

Table name = orders

Each item sent to the system contains the following

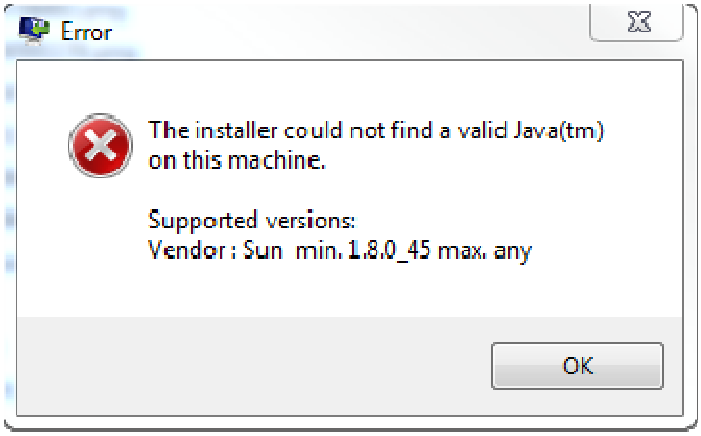
Column name = Id, string	**unique identity for each line, Primary Key**
Column name = ordered, string	**unique identity for each order sent to the system**
Column name =qty, integer	**number of items on line**
Column name = details, string	**Main description**
Column name = attributes, string	**Any attributes passed**
Column name = notes, string	**Any manual notes added to the line**
Column name = ticketed, string	**Name for ticket id can be customer name, table number**
Column name = ordertime, timestamp	**Time row added, auto calculated**
Column name = displayed, integer	**Display number**

Installation (Windows)

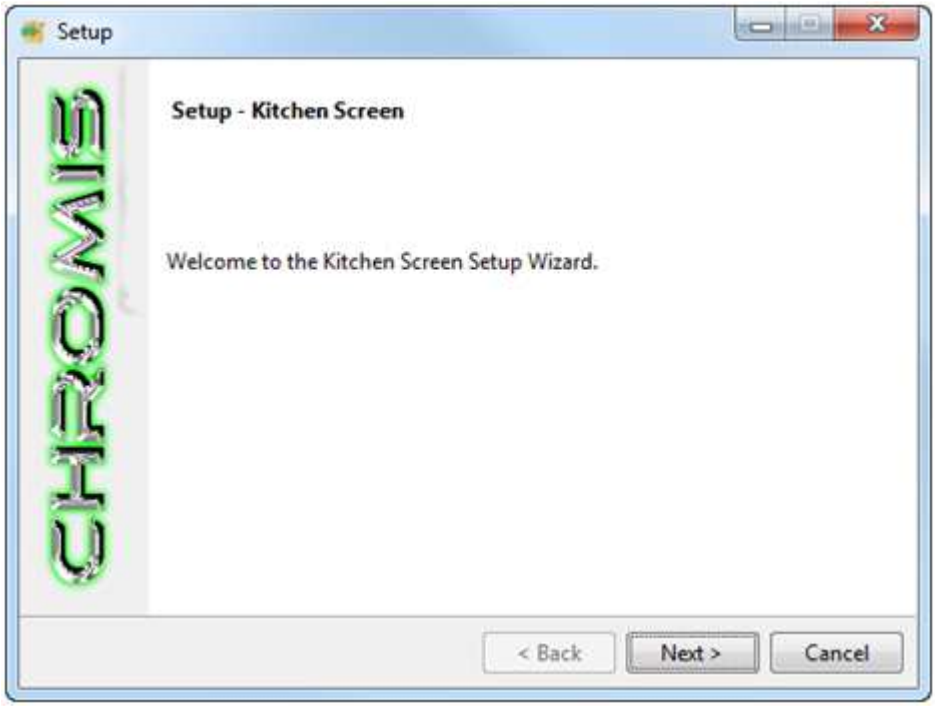
If you do not already have the install files, download then from

https://sourceforge.net/projects/chromis/files/

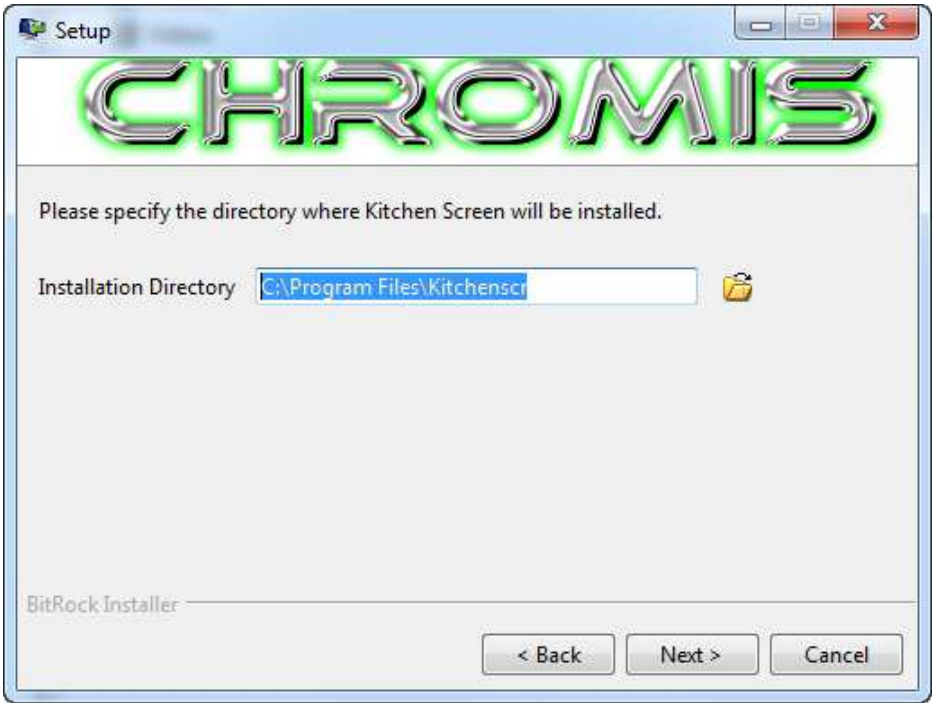
Launch the installer



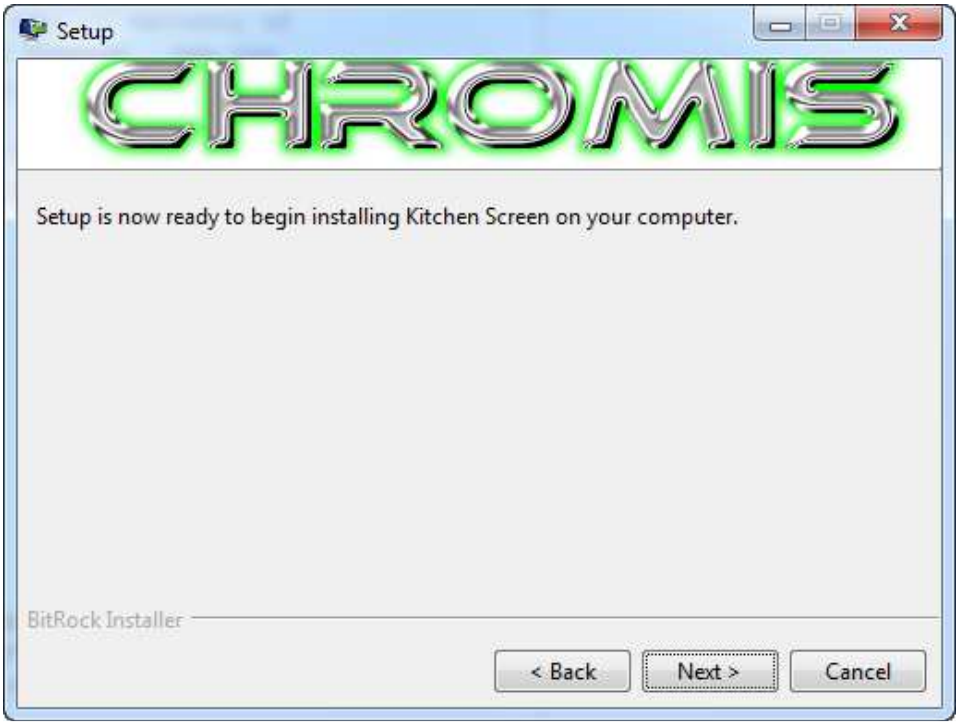
You will get a warning if the correct version of Java is not installed, if that is the case download and install the correct version of java an restart the installer.



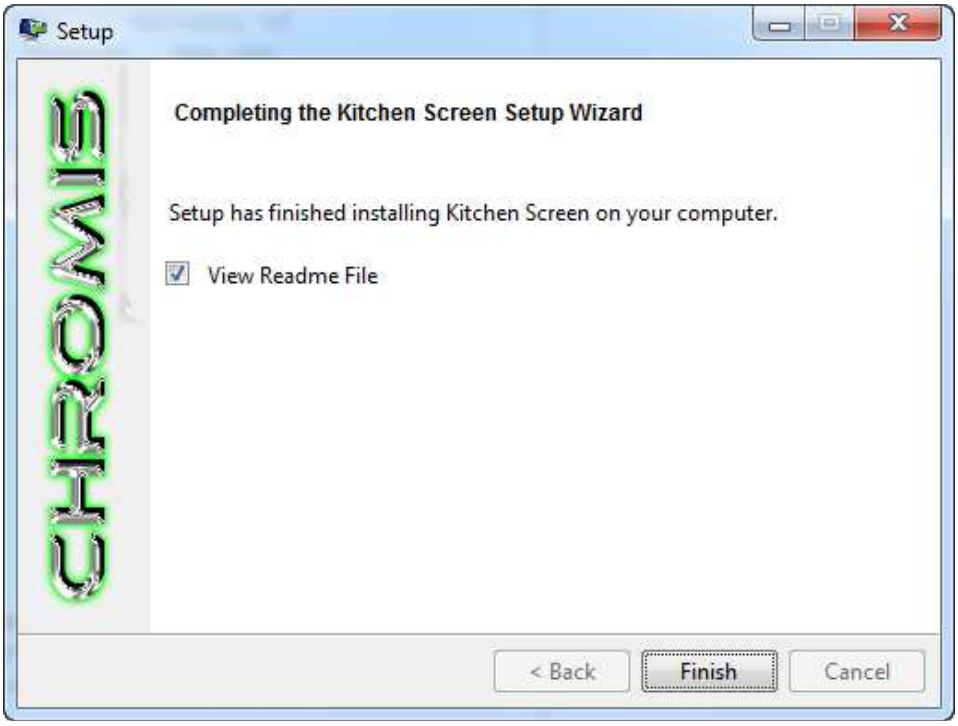
Select ‘Next’



Click ‘Next’



Click ‘Next’



Click ‘Finish’

Installation is now complete.

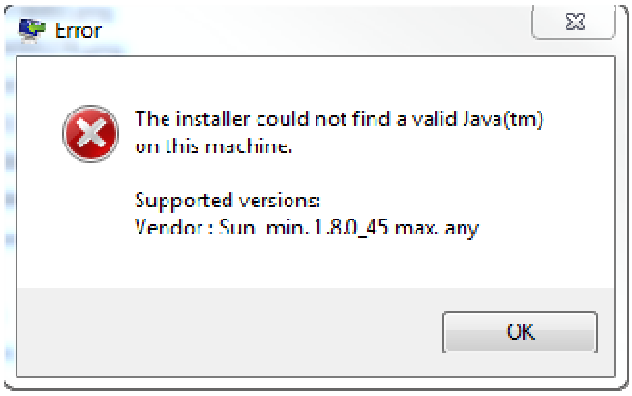
Installation (Linux)

These installation instructions were tested and a version of Linux Mint.

open the Terminal window.
Execute the followining commands

cd Downloads
wget https://sourceforge.net/projects/chromis/files/linux/ kitchenscr-1.40-linux-installer.run
sudo chmod +x kitchenscr-1.40-linux-installer.run
sudo chmod +x /opt/kitchenscr/*.sh
sudo ./kitchenscr-1.40-linux-installer.run

The installer will now run



You will get a warning if the correct version of Java is not installed, if that is the case download and install the correct version of java an restart the installer.

Click 'Forward'

Click ‘Forward’

Click ‘Forward’

Click ‘Finish’

Installation is now complete.

Installation raspberry Pi

This application has only been tested on the Raspberry Pi 2. Raspbian - NOOBS 1.4.1

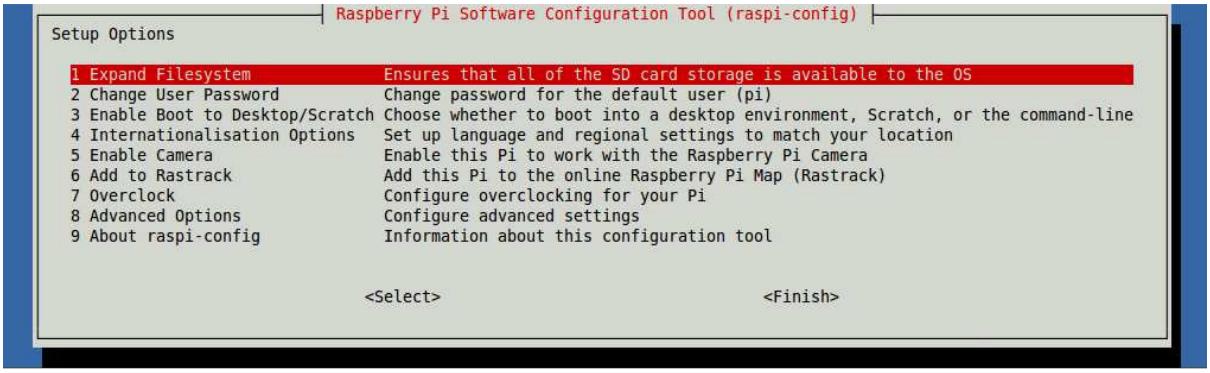
At this moment in time I have be unable to locate an installer for applications on the Raspberry Pi. With this in mind the instruction will use a SSH and SFTP application to install. These instruction will cover installation using a Windows based PC.

When you run a JavaFX application on the Pi it takes over the complete screen. The application is configured to poweroff once you exit the application, if you wish to stop this then you need to remove ‘sudo poweroff’ from the .sh files.

Preparing the Raspberry Pi.

This is not intended as a step by step guide for setting up the Pi, only as a guide to allow the software to run from a base install of Raspbian and not changed the passwords or default user.

Install the raspbian OS for the Pi using NOOBS 1.4.1, this will install version 1.8.0 of Java, that includes JavaFX.



Configure your time zone using option 4 from the menu.

Configure boot to desktop using option 3

Configure the Overscan using option 8

Select A1 Overscan and disable (if this is not completed, the screen will not appear correctly when running the application)

Configure the memory spli using option 8

Select A3 memory Split, the set the GPU to 128

Reboot the Pi

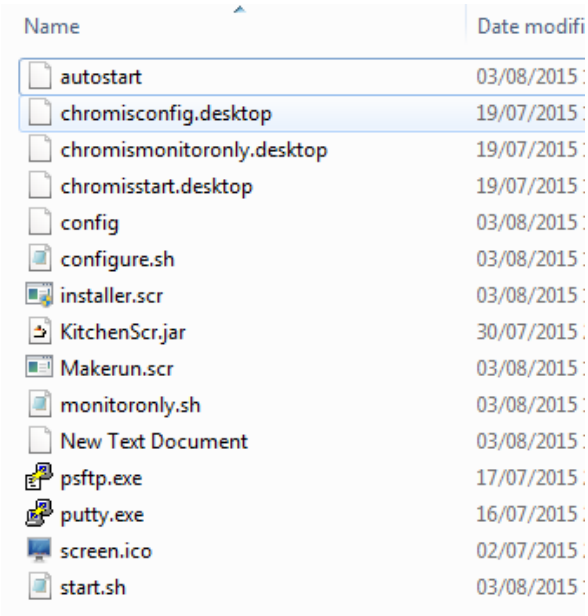
You will need the Pi connected to the network and also know its IP address.

If you do not already have the install files, download then from

https://sourceforge.net/projects/chromis/files/

1. Unzip the file to a folder on your PC.
2. Open a cmd window
3. Change the directory to the folder with the unzipped files

The folder should contain the following files



You can now complete the tasks manually or use the scripts supplied.

Using supplied scripts

In the command window run the following commands

Psftp 'ip address of the pi' –l pi –pw raspberry –b installer.scr

If you get a message about cached keys respond 'y'.

putty -ssh pi@'ip address of the pi' 22 -pw raspberry -m makerun.scr

The above scripts will create the required directory, copy files to it, create the desktop shortcuts, set the application to autostart and set the security on the files.

Manually

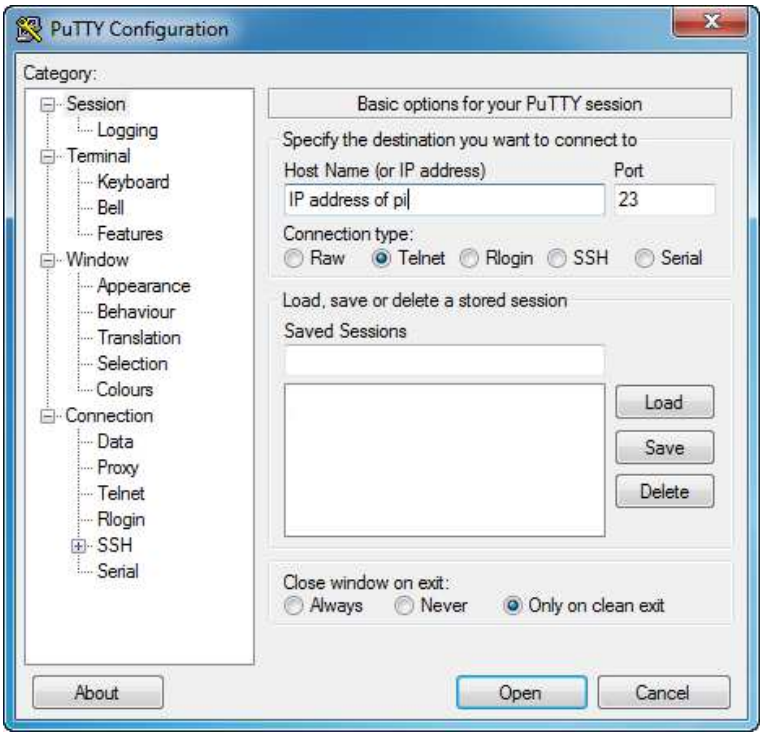
Open psftp.
Connect to the pi using the command open 'ip address of the pi'
Enter username and password when prompted.
Execute the following commands one at a time

mkdir kitchenscr
cd kitchenscr
put KitchenScr.jar
put start.sh
put configure.sh
put monitoronly.sh
put screen.ico
put autostart
put config
cd ..
cd Desktop

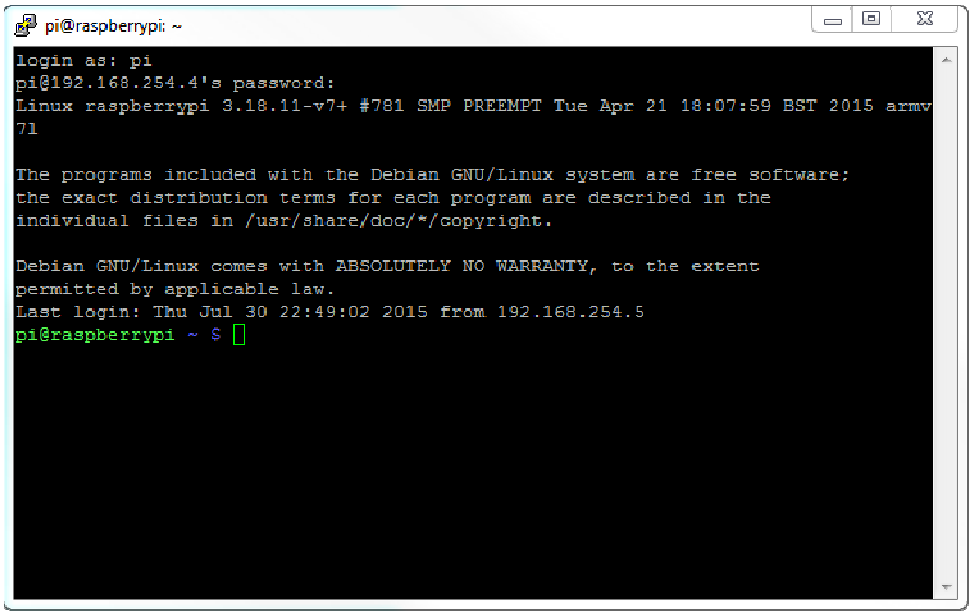
put chromisconfig.desktop
put chromismonitoronly.desktop
put chromisstart.desktop
cd ..

close psftp.

Open putty and connect to the pi using SSH



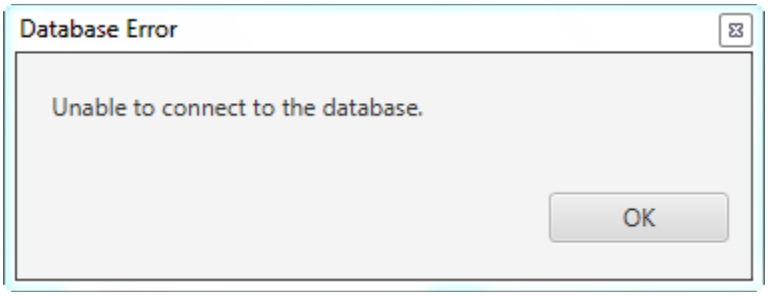
Enter username and password when prompted



Execute the following commands in order

cd kitchenscr
sudo chmod 777 start.sh
sudo chmod 777 configure.sh
sudo chmod 777 monitoronly.sh
sudo mv autostart /etc/xdg/lxsession/LXDE-pi/
sudo mv config /etc/kbd/
sudo chmod 777 /etc/xdg/lxsession/LXDE-pi/autostart

Restart the Pi and the application should autorun, as it will be the first time it should report that it cannot connect to the database

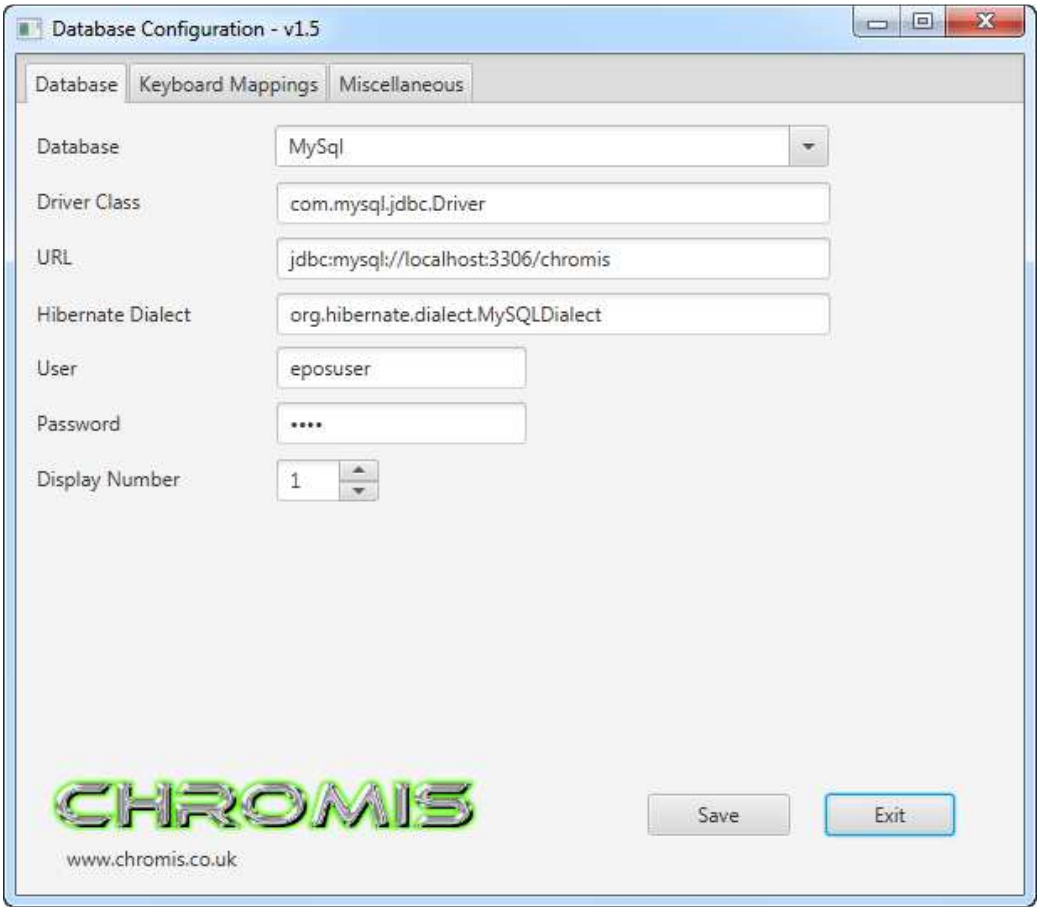


Click Ok, and you should now be taken into the configuration screen.

Configuration

Launch the configuration panel. This is the only time you actually require a keyboard, if you are planning on using a touch screen or bump bar

Database



There are 4 database options currently available within the application.

These are selectable from the combobox

- MySQL
- Apache Derby Client/Server
- Oracle 11g Express
- PostgreSQL

If the database you wish to use is not in the list, the details can be entered manually, all the details will need to be entered manually.

As the system using Hibernate for the database backend, you will need to know the Hibernate dialect for the database you will be using.

If you are using a database not in the list or the version of JDBC driver is not correct, copy the required JDBC file into the 'lib' folder with the Unicenta folder.

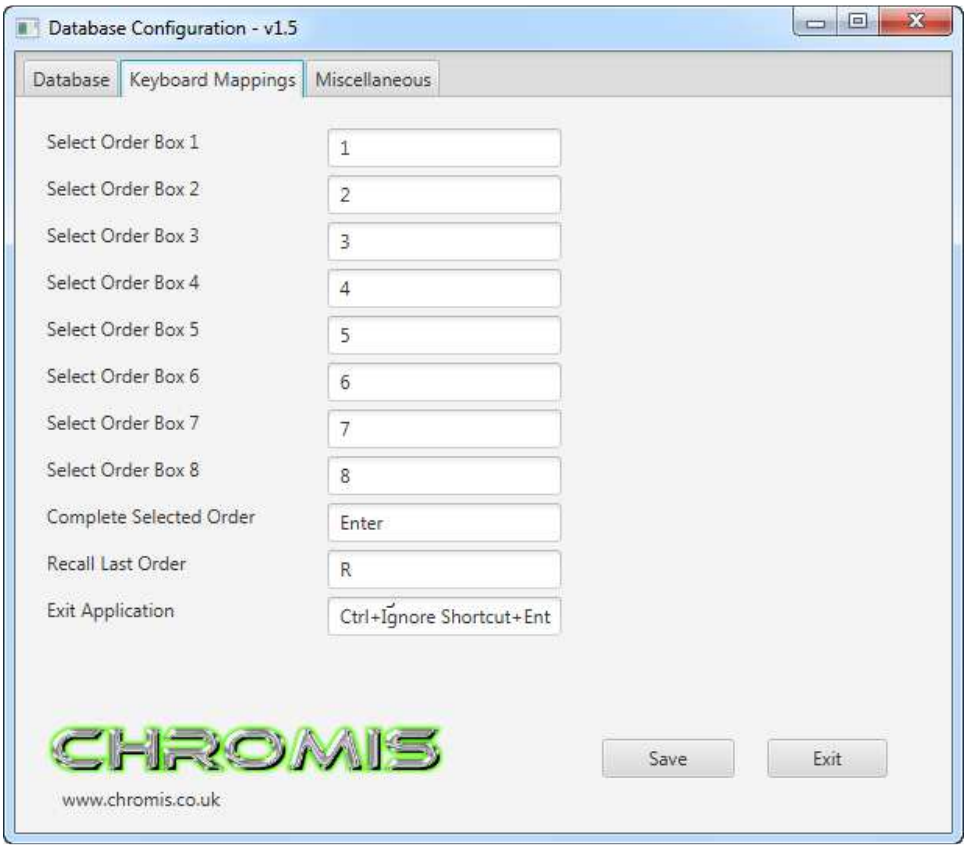
Complete all the other fields relating to the database.

The final field is the 'Display Number', for now set this to '1'. If the POS software can support it, you can have up to 9 screens all displaying different information.

For example

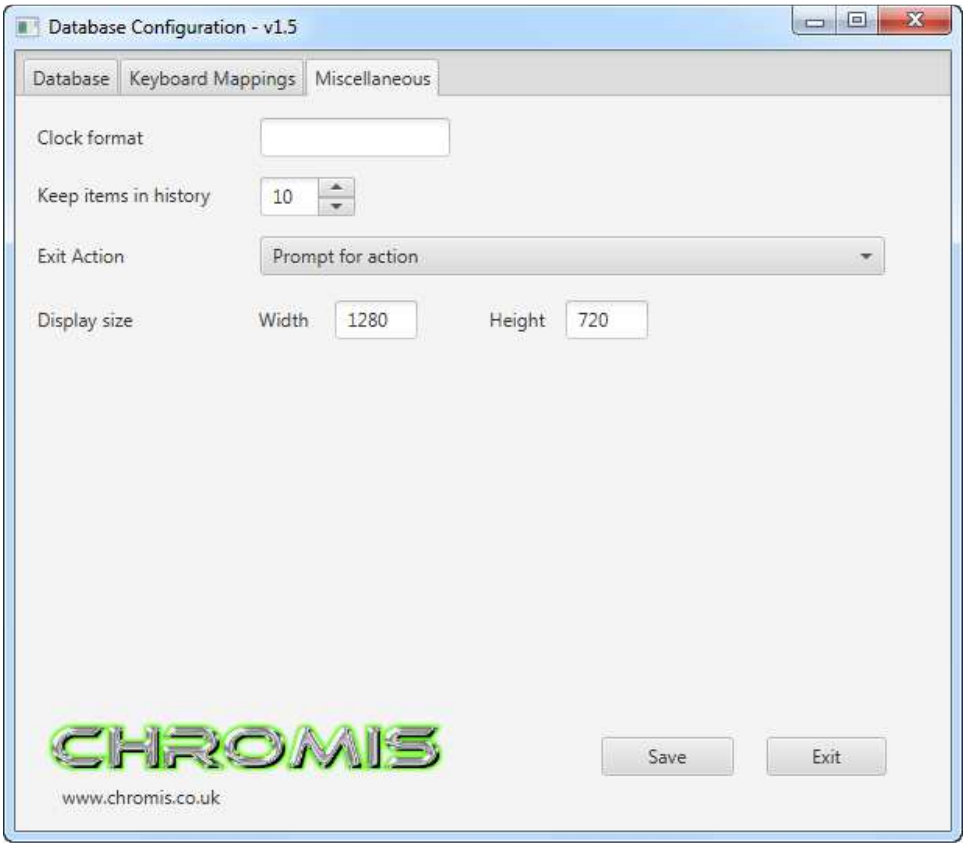
- Starter screen (1)
- Mains screen (2)
- Dessert screen (3)
- Bar screen (4)

Keyboard Mappings



If you are planning to use a keyboard or bump bar using this screen you are able to set up the keys that should be mapped to a function. Once you have set up the keys save your save settings.

Miscellaneous



By default the display is set to use a size of 1024 x 768. If you wish to change these enter the new and the display will adjust to that when run. If you specify a size that is greater than the screen resolution, the display will be reset to use the screen resolution.

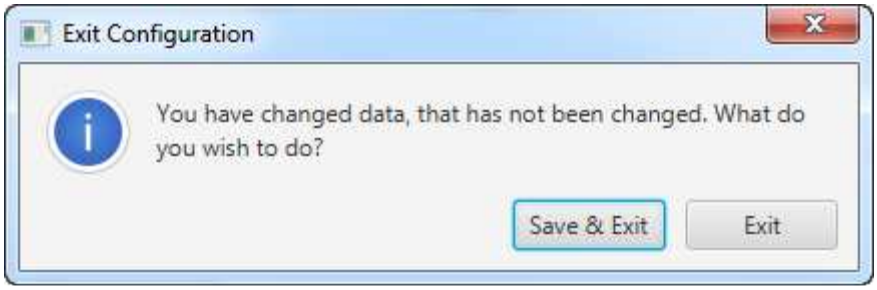
Leaving the clock format blank will use the system default configuration, if you wish to change this then enter the format you require. The application check the format you enter when it runs and will run the default if you enter the format incorrectly.

eg H:mm:ss a display AM or PM, with no leading zero on the time.

Keep items in history is used by the recall option, set this to number of orders you wish to keep in the history.

Once you have configured your settings, selecting ‘Save &Exit’ will save your configuration to ‘chromis.properties’.

If you select ‘Exit’ and you have unsaved data, you will be given the option to save your configuration.



After a save the application will attempt to create the table required in the database. This process does not perform any changes to the database if the table exists.

Main kitchen Screen

Launch the main screen.

When you launch the main screen if the application is unable to connect to the database, you will be taken into the configuration panel again to check your settings.

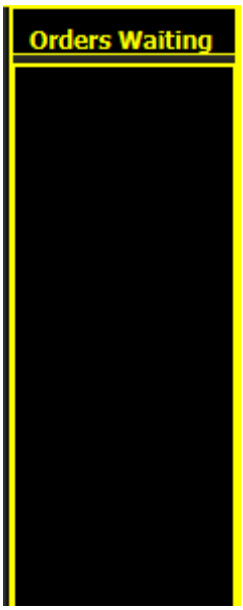
The screen is composed of 3 main areas.

- Main order panels
- Orders waiting
- Button and time bar

There are 8 main order panels; which are populated in the order that they are sent to the database by the users.

The panel shows the order reference (ticketed column from the database), the elapsed time since the order was placed. And finally the order details itself, if required scroll bars will be displayed in the panel.

Down the right hand side is the order waiting panel. Iif all the main panels are populated then any further orders are added to the list in this panel.



In the bottom right side is the real time clock

The main button in the centre of the bar, is used when an order is completed.

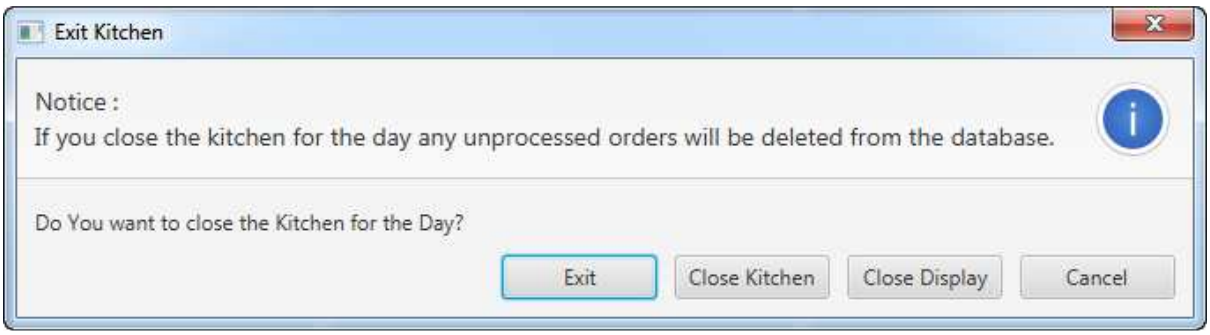
To complete an order, click on the main panel that contains the order, this will then update the text on the button, to indicate which order you are about the complete.

Click on the button and the order will then be removed from the list. If there are orders waiting, the order at the top of the list will be moved to the main panels.

To exit the screen, click on exit.

When you complete an order, the recall button will be displayed as long as there is a order in the history file.

You will then be presented with an exit dialog box.



Exit – exits the application.
Close kitchen – will remove any remaining orders from the database and exit the application.
Close Display- just closes all order for the current display.
Cancel – returns back to the order screen.

Configuring Chromis POS to work with the screen

By default Chromis contains the core code to control the kitchen screens, but will need to tell the software to send the details to the screen.

This change is in the script.sendorder.

To activate the process when the user selects sent for kitchen the script will need to updated as below.

```
boolean kitchen = false;

for(int i= 0; i < ticket.getLinesCount(); i++){
        line = ticket.getLine(i);

        if (line.isProductKitchen() && (line.getProperty("sendstatus") == null)){
                line.setProperty("sendstatus", "No");
        }
        if((line.isProductKitchen()) && (line.getProperty("sendstatus").equals("No"))){
                kitchen = true;
        }
}

if (kitchen) {
        sales.kitchenOrderScreen(); // can be passed with other arguments
        sales.printTicket("Printer.TicketKitchen");
}
if (kitchen){
        javax.swing.JOptionPane.showMessageDialog(null, "Order sent to Kitchen");
} else {
        javax.swing.JOptionPane.showMessageDialog(null, "Nothing to Send", "Warning",
JOptionPane.WARNING_MESSAGE);
}

for(int i = ticket.getLinesCount()-1; i>= 0 ; i--){
        line = ticket.getLine(i);
        String a = line.getProperty("sendstatus");


        if((line.isProductKitchen()) && (line.getProperty("sendstatus").equals("No"))){
                line.setProperty("sendstatus", "OK");
        }
}
```

The kitchenOrcerScreen() method is capable of receiving a number of parameters.

- kitchenOrderScreen()
- kitchenOrderScreen(String "ticketId")
- kitchenOrderScreen(Integer display, String “ticketed”)
- kitchenOrderScreen(Integer display)

kitchenOrderScreen() - will create the id for the screen automatically based on the following in order

- Customer name
- Table Name
- PickupID

It means that the pickupId is now allocated as soon as the order is sent to the kitchen.

kitchenOrderScreen(String "ticketId") - allow you to create your own id for the screen, simply replace ‘ticketId’ with the details you wish to use.

kitchenOrderScreen(Integer display, String “ticketid”) – allows you to pass your own display number and ticketed.

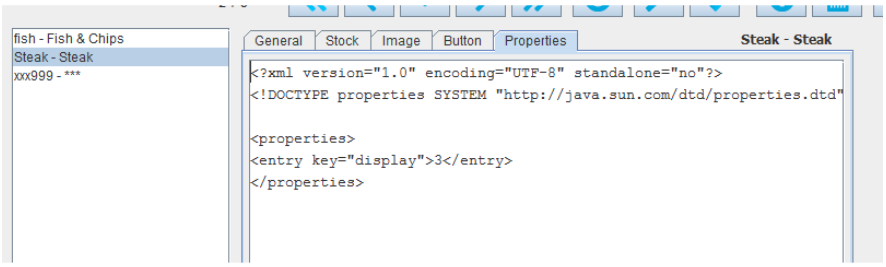
kitchenOrderScreen(Integer display) – Allows you to pass your own display number but will create the id for the screen automatically based on the following in order

- Customer name
- Table Name
- PickupID

Customizing the products

If you wish to send to the display based upon the product

Open products in Unicenta. Select the product and open its properties tab.



By adding the correct code in the properties box, you can control what display the product will be sent to, this should only be used in multiple location displays situations.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"

<properties>
<entry key="display">3</entry>
</properties>
```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http//java.sun.com/dtd/properties.dtd">

<properties>
<entry key="display">2</entry>
</properties>

So in the above example, any orders for Steak will be sent to display 3.

Without any setting in the properties files, display ‘1’ or the display passed with the method will be used.

Finally, Please supply any comments, issues or feedback you may have.

Thanks go to Nick Deppe for the latest changes to the application.

John Lewis (john@chromis.co.uk)
Chromis.co.uk