iLab Point of Sale

iLabCentral

v1.1 Technical Documentation and Setup Procedure
12/06/2015
Updated 13/09/2016

iLabMalta - L.Tonna

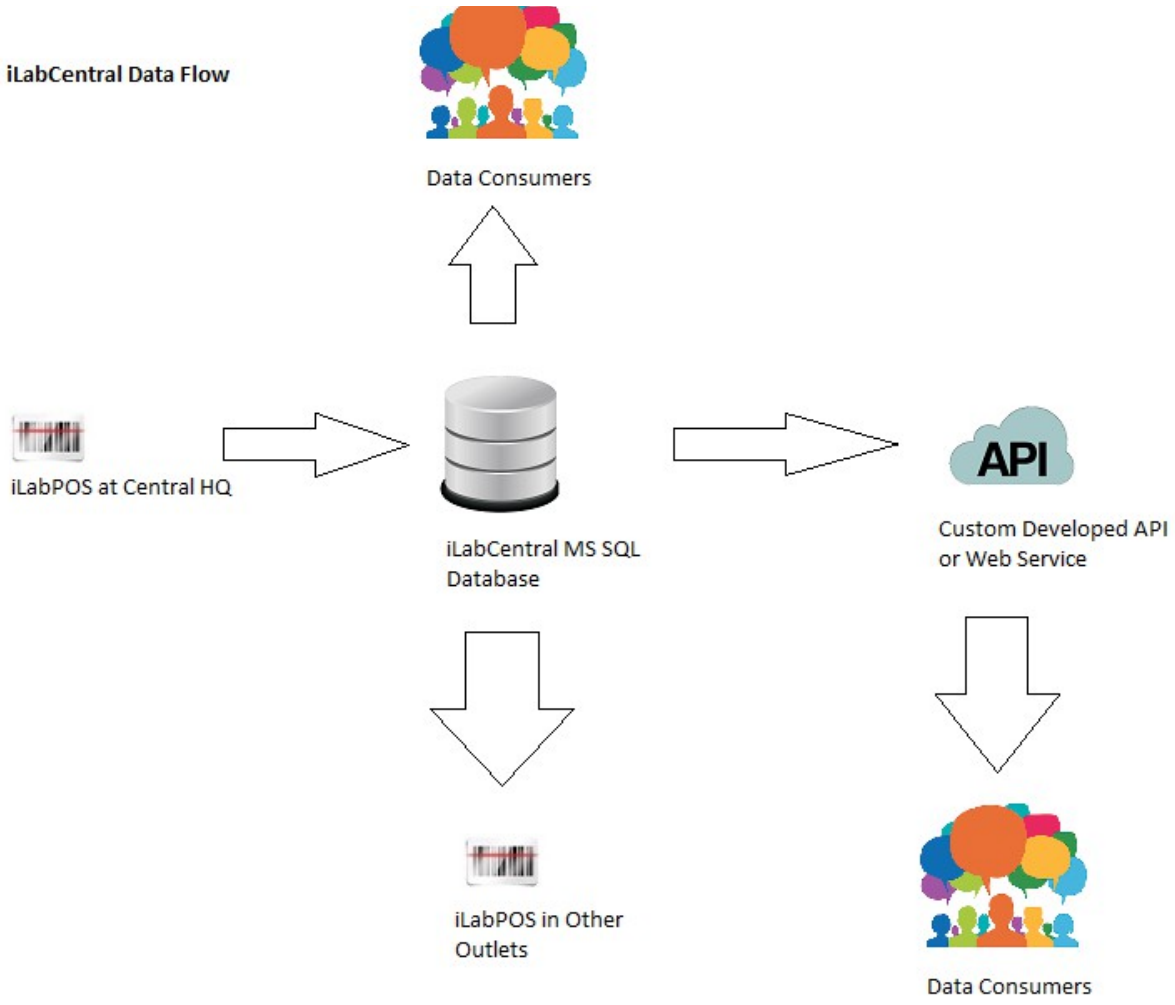
1. Introduction

The primary objective of iLabCentral is to interconnect several retail outlets for a business. Products/Pricing/Categories etc... are defined in one main outlet, and an unlimited amount of other "sub-outlets" are then automatically syncronised with the main outlet. The main outlet can also be a store, or company head quarters. Each outlet/store will be able to view what stock is available in other outlets, and also transfer stock to other outlets and stores.

The secondary objective of iLabCentral is to share iLabPOS data containing all Product information with various consumers, such as online shops, accounting systems, sales catalogues, and any custom software you wish to develop and have integrated with your stock/pos system. iLabCentral will provide you with an online data repository to retrieve product information, pricing, stock quantities, pictures and more, in quasi realtime. You may query the MS SQL database directly through your custom software, and you may also develop WEB API's, Web services according to your needs.

iLabCentral runs on Microsoft SQL Server 2008 and above. It is also fully compatible with MS SQL Express. The iLabCentral MS SQL database can be setup on your Local Network, or on a server accessible over the Internet. It can also be setup on Hosting servers such as GoDaddy.

The client will choose an iLabPOS instance, usually at the main shop or head quarters to act as the Master POS. The iLabCentral Sync Tool (POSSYNC.EXE) is pre-configured and scheduled on the main HQ iLabPOS instance to run every 5 minutes (less often or more often as required). The process will upload any new products, product updates, product pictures, pricing etc... to iLabCentral, hence having an updated MS SQL Database of all your iLabPOS data accessible from various locations. In turn, Slave Shops or Online shops will access the iLabCentral Database to check for updates, new products and pictures.



Creation of products/price changes/picture updates are only possible on iLabPOS at HQ. So are categories, category groups, suppliers, supplier types, tax departments, system data, expense types. Once an article is created at HQ, it is uploaded to the Central Database, and then downloaded by the various shops or online shops.

Normally changes are refreshed to slave shops every 5 to 30 minutes, depending on Internet Connection Speeds, size of the data, and your needs.

Invoices and stock quantities are also uploaded from the HQ and various shops, hence having one central database with all invoices and stock qtys.

Any shop to shop/store to shop/shop to store stock transfers are also routed via iLabCentral.

Retail Companies with Several Outlets

Through iLabCentral you can have multiple retail outlets centrally controlled. Products will be controlled centrally and feeder (or slave shops) will receive stock / pricing updates / stock transfers accordingly.

Simply put, iLabCentral works like Google Drive / One Drive / Drop Box. In each shop you have iLabPOS installed on the local PC or network. When the internet is available, stock, sales,

clients, etc... are uploaded on the cloud (Or on one of your servers).

If internet goes down in your shop, iLabPOS will still keep running. You will of course loose features such as viewing what stock is availble in other shops, but otherwise everything will function.

One of your shops/warehouse is set as the Master (or HQ). Setting of prices, creation of new stock, new suppliers & categories is all managed on the Master (Only the Master). Once you change the price of an item on the master, the price is reflected in other shops after a few minutes. Same when creating new stock, a few minutes after creating a stock item it is duplicated on all other shops connected to iLabCentral.

From each shop you can view what stock is available in other shops. Further more, shop to shop stock transfers can be done through the system. The sender will create a stock transfer, choosing which shop stock is being sent to. A stock movement sheet is printed and sent with the stock. The receiving end will then need to confirm the stock transfer (by a unique reference code) before it is commited.

From HQ, one can extract global sales and stock data for each individual shop, or for all shops together

Each shop will require an internet connection. HQ will require a fixed IP Address.

Online Shops and Sales Catalogues

If you are a software developer and your client (using iLabPOS) wants to develop an online shop, then you are looking in the right place. iLabCentral is your data repository from where you can collect all product information. Your client will be using iLabPOS to create products, attach product pictures, specify HTML encoded article descriptions, attach meta keywords, etc.... All this information together with stock availability is then uploaded and synced on an MS SQL Database every few minutes, and which you can then access, read only, freely. iLabCentral is mostly one way, ie, only providing you with product information. It is also possible to pass on Sales Orders to the maister POS Instance.

Kindly note that iLabCentral needs to be purchased seperately from iLabPOS.

2. Data Definition

Through this manual we shall explain all data elements in the iLabCentral MS SQL Database. This part of the manual is intended for those developers who wish to integrate software with iLabPOS/iLabCentral. Integration can be either direct with SQL Server, or through a WEB API/Web Service.

2.1 Generic

As a general rule, most tables will have 2 columns included.

_ilc_versionno will contain the version of the row. Through this field you can check if there were any updates carried out on the product. The Version Number is always incremented by 1, starting from 1.

_ilc_lastupdated will contain the timestamp a row was updated.

Hence you would normally take stock of these 2 fields in your system so that in the next synconisation you can determine whether any changes were carried out or not. Once would suggest using the version_no field to compare.

You will also notice 2 primary keys in each table. One is the iLabCentral Primary key which is automatically generated by SQL Server. The second primary key is the iLabPOS primary key. Both primary keys are static and automatically generated in series, however they might not always contain the same numeric value.

2.2 Product Catalog

The product catalog is spread across several tables, as follows:

tb_article - Product listing including most one to one relational data such as barcode, description, meta key words, pricing.

tb_article_image - Containing several images per article

tb_article_location_stock - A many to many relationship table between an article and a location, specifying the stock quantity available.

tb_artkit - Congregates several articles together to form a kit

tb_category - Article Categories

tb_catgroup - Category Groups

tb_department - VAT Rates

tb_supplier - Supplier Table

tb_brand - List of Brands

Articles (Products) - tb_article

This is the main product definition. Pay particular attention to artdiscont (Discontinued article) and artwpublish (determines whether article is published on online shop or not)	
art_pk	iLab Central Article Primary Key (Never Changes)
artid	iLabPOS Article Primary Key
artbarcode	Article Barcode
artrefcode	Article Reference Code
artdesc	Article Description
artlongdesc	Article Long Description
artnote1	Article Comment 1
artnote2	Article Comment 2
artnote3	Article Comment 3
artsuppid	Supplier Foreign Key - Join with tb_supplier.suppid
artdptid	Tax (VAT) Department Foreign Key - Join with tb_department.dptid
artctgid	Category Foreign Key - Join with tb_category.ctgid
artshelf	Article Shelf Location
artsellprice	Selling Price 1 Excluding VAT
artcostprice	Cost Price Excluding VAT
artsellprice2	Selling Price 2 Excluding VAT
artsellprice3	Selling Price 3 Excluding VAT
artsellprice4	Selling Price 4 Excluding VAT
artcostdptid	Tax Rate applicable on purchase cost
artwsaleprice	Consumer Price including VAT
artcostadd	Additional Cost excluding VAT
artofferind	Special Offer #1 (1 - Activated, 0 - Not Activated) - Customer buys x qty, pays only for y qty
artofferbuy	Special Offer #1 - X Quantity
artofferfree	Special Offer #1 - Y Quantity
artofferind2	Special Offer #2 (1 - Activated, 0 - Not Activated) - Customer buys X Quantity for a special combined price of Y
artoffer2buy	Special Offer #2 - X Quantity
artofferpay	Special Offer #2 - Y Amount incl. VAT
artsaleprice	Reduced Sale Price excl. VAT
artsalebegin	Date from when Reduced Sale Price applies
artsaleend	Date until when Reduced Sale Price applies
artpoints	Number of points earned per item sold
artstock	Not used - refer to tb_article_location_stock
artreordpnt	Reorder Point
artreordqty	Reorder Quantity
artcolor	Colour
artsize	Size

artunit	Units (Kgs / Meters / eyc...)
artdateadd	Date Article Created
artlastsold	Date Last Sold
artdiscont	Item Discontinued (0 - Still Active, 1 - Discontinued)
artbrand	No Longer used - replaced with artbrandid
artwarr	Warrant Conditions Free Text
artiskit	Is Article a Kit made of other Articles (1 - Yes, 0 - No) - Refer to tb_artkit
artcodepri	Price in Barcode (1 - Yes, 0 - No)
artcodepridec	Number of Decimal Places in barcode
artnoquik	No QuikScan - Used in iLabPOS to de-activate Quick Scan on scanning a product
artlstexpdt	Oldest Expiry Date
art_ilc_lastupdated	Date Last Updated on iLabCentral
art_ilc_versionno	Version No of article - incremented with each update
art_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted
artwpublish	Publish (1 - Yes, 0 - No) - Can be used in online shops to determine whether a product is published in the shop or not.
artwshowhome	Show on Home Page (1 - Yes, 0 - No) - Can be used in online shops to determine whether a product is featured on the Home Page or not.
artwmetakeyw	Meta Keywords - Used for online shops to improve meta searches
artwmetadesc	Meta Description - Used for online shops to improve meta searches
artwmetatitle	Meta Title - Used for online shops to improve meta searches
artwallcustr	Allow Customer Reviews - Can be used in online shops to determine whether you will allow customer reviews or not
artwdispavail	Display Availability (1 - Yes, 0 - No) - Can be used in online shops to determine whether you wish to display if the item is in stock or not.
artwdispqty	Display Quantity Available (1 - Yes, 0 - No) - If artwdispavail is on, then this will determine if you also want to display the quantity in stock still available.
artwbuyenabled	Buy Enabled (1 - Yes, 0 - No) - Can be used in online shops to determine whether a product can be purchased online or is just online for cataloging purposes.
artwebdesc	Web Desciption - An nvarchar(max) field containing an HTML detailed description of the product (Would normally contain HTML elements)
artwwlist	
artwcallpri	Call for Price - Can be used in online shops when you do not want to show the price
art_scm_show	1 - Show on Sales Catalogue, 0 - Do not show
artbrandid	Brand Foreign Key - Join with tb_brands.brandid

Article Images - tb_article_image

Contains several images per article. Joined with tb_article through aim_art_fk

When images are removed, the isactive indicator is set to 0 and the binary field is nullified. In case a new image is uploaded, a new row is inserted with the same picture number. Hence you need to ensure that you check for deleted images, and for new images being uploaded.

aim_pk	iLab Central Article Image Primary Key (Never Changes)
aim_art_fk	tb_article Foreign Key
aim_filename	Image File Name including the extension
aim_picturebinary	Image binary
aim_picture_no	Picture serial number per article, starting from 1, finishing with 3
aim_picture_mime_type	Mime Type, example: image/jpeg
aim_isactive	1 - Active, 0 - Deleted
aim_created_by	Created by Timestamp
aim_created_on	Created on Timestamp
aim_lastupdated_by	Last updated Timestamp
aim_lastupdated_on	Last updated on Timestamp
aim_recordversion	Record Version Number

Stock - tb_article_location_stock

Will return 1 row per article per location. Usually the location with PK 1 is the Master Location. However in certain instances you might need to sum the quantity per article for all locations, this depends on your clients needs.

Stock quantities are continuosly refreshed and do not carry a version number like other tables. In this case always assume a change, or else access atock quantities in real time.

stk_pk	iLab Central Primary Key (Never Changes)
stk_loc_pk	tb_location Foreign Key
stk_artid	tb_article Foreign Key - Joins with the iLabPOS Article Primary Key, and not with the iLabCentral Primary Key.
stk_qty	Quantity in Stock
stk_last_updated	Time Stamp Last Updated

Article Kit - tb_artkit

Articles in tb_article where tb_article.artiskit = 1 will have a definition of the articles in this table. Example, if article in tb_article with artid = 10 has artiskit = 1, then this is not a physical article but a collection of several other articles sold at a special price when purchased all together.

To obtain articles which from the kit in the example above, the query will look something like this:

*SELECT * FROM tb_artkit*
INNER JOIN tb_article ON tb_artkit.atkartid = tb_article.artid
WHERE atkmasterid = 10
AND atk_ilc_status IN (1,2)

*The price of the kit excluding VAT is derived by calculating a sum of (QTY * PRICE) for all child articles.*

atk_pk	iLab Central Primary Key (Never Changes)
atkid	iLabPOS Kit Primary Key
atkmastertid	Master Article - Points to tb_article on artid
atkartid	Kit component - Points to tb_article on artid
atkqty	Quantity used in Kit of this pparticular article
atkpriceind	
atkcreated	Date Created in iLabPOS
atklastuser	User last updated record in iLabPOS
atksellprice	Selling price excluding VAT applicable in KIT
atk_ilc_lastupdated	Last Updated on iLabCentral
atk_ilc_versionno	Row Version Number
atk_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

Categories - tb_category

Each article is joined with a category through tb_article.artctgid = category.ctgid.

Categories are then further grouped into Category Groups. Each Category group can contain several categories, and in turn each category can contain several articles.

ctg_pk	iLab Central Primary Key (Never Changes)
ctgid	iLabPOS Category Primary Key
ctgname	Category Description
ctgcode	Category code or abbreviation
ctggrpid	Category Group (Join with tb_catgroup)
ctgitouch	True (1) or False (0) - Determines whether a Category is visible in iLabPOS iTouch Screen
ctgpublish	True (1) or False (0) - Determines whether the category is to published on the online shop or not.
ctgshowhome	True (1) or False (0) - Determines whether the category is featured on the homepage or not.
ctgshowtopmenu	True (1) or False (0) - Determines whether the category is shown on the top menu or not.
ctgmetakeyw	Category Meta Keywords
ctgmetadesc	Category Meta Description
ctgmetatitle	Category Meta Title

ctg_ilc_lastupdated	Last Updated on iLabCentral
ctg_ilc_versionno	Row Version Number
ctg_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

Category Groups - tb_catgroup	
grp_pk	iLab Central Primary Key (Never Changes)
grpid	iLabPOS Primary Key
grpcode	Category Group Code
grpdesc	Category Group Description
grp_ilc_lastupdated	Last Updated on iLabCentral
grp_ilc_versionno	Row Version Number
grp_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

TAX Departments - tb_department	
dpt_pk	iLab Central Primary Key (Never Changes)
gdptid	iLabPOS Primary Key
dptname	Tax Rate Name
dptcode	Tax Code (Example F)
dpttaxrate	Ignore
dpttaxrate_actual	Tax Rate %
dpt_ilc_lastupdated	Last Updated on iLabCentral
dpt_ilc_versionno	Row Version Number
dpt_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

Brands - tb_brands	
brand_pk	iLab Central Primary Key (Never Changes)
brandid	iLabPOS Primary Key
branddesc	Brand Name
brand_ilc_lastupdated	Last Updated on iLabCentral
brand_ilc_versionno	Row Version Number
brand_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

2.3 Supplier Database

Suppliers - tb_supplier	
supp_pk	iLab Central Primary Key (Never Changes)
suppid	iLabPOS Primary Key
suppname	Supplier Name
supprefno	Supplier Reference Code (Might be empty)
supptype	Joined with tb_supptype and determines the supplier family
suppaddr	Supplier Address (No and street)
suppcity	Supplier Address (Locality)
suppzip	Supplier Address (Post Code)
suppcountry	Supplier Address (Country)
suppcont	Supplier Main Contact (Name and Surname)
suppemail	Supplier Email
supptel	Supplier Phone Number
suppfax	Supplier FAX no
suppmobile	Supplier Mobile No
suppcomm	Supplier Comments
supprecdur	Normal Order Duration in days
supptax	Supplier VAT no.
supp_ilc_lastupdated	Last Updated on iLabCentral
supp_ilc_versionno	Row Version Number
supp_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

Supplier Types (Group/Family of Suppliers) - tb_supptype	
spt_pk	iLab Central Primary Key (Never Changes)
sptid	iLabPOS Primary Key
sptdesc	Type Description
spt_ilc_lastupdated	Last Updated on iLabCentral
spt_ilc_versionno	Row Version Number
spt_ilc_status	1 - Inserted, 2 - Updated, 3 - Deleted

2.4 Sales

Invoice/Cash Sale - tb_invoice Invoices are posted and never manipulated. Hence the ilc_versionno field is not present.	
inv_pk	iLab Central Primary Key (Never Changes)
inv_loc_fk	Joins with tb_location indicating which site carried out the sale
inv_site_id	Indicates the terminal number. Usually always 1, unless location has several terminals.
inv_no	iLabPOS Invoice/Cash Number - Unique and generated in series
inv_date	Date of Invoice
inv_time	Time of Invoice
inv_total	Total amount of invoice
inv_deposit	Amount paid as deposit. Always equal to inv_total for a cash sale, but may be less or zero for an invoice.
inv_clt_id	Joins with tb_clients and indicates the client (Might not always be populated, depending on the setup. You might wish to use inv_clt_name instead of joining)
inv_clt_name	Snapshot of the Client Name
inv_created_on	Date posted on iLabCentral
invusrid	Contains the iLabPOS User ID who carried out the sale

Invoice Detail - tb_invline Invoices are posted and never manipulated. Hence the ilc_versionno field is not present. The line total including VAT is calculated as follows: inl_netamt + inl_vatamt - inlnetdisc - inlvatdisc	
inl_pk	iLab Central Primary Key (Never Changes)
inl_inv_pk	Joins with tb_invoice.inv_pk - The Invoice Header
inl_art_id	Join with tb_article.artid to identify the article
inl_art_desc	Snapshot of the article name on sale
inl_art_barcode	Snapshot of the Article Barcode
inl_art_refcode	Snapshot of the Article Reference Code
inl_id	The detail line number starting from 1 per invoice
inl_qty	Quantity sold on sale (Decimal)
inl_unitprc	Unit price excluding VAT
inl_vatrate	VAT Rate
inl_netamt	Quantity multiplied by the Unit Price
inl_vatamt	Total VAT Applicable on NET

inl_netdisc	Discount amount on inl_netamt
inl_vatdisc	Discount applicable on VAT portion
inl_costprice	The cost price excluding VAT per unit
inl_costvatrate	The VAT Applicable on cost (Usually ignore this field and take inl_vatrate)
inl_desc	Line Comments

2.5 Others

Locations (Shops) - tb_location	
loc_pk	iLab Central Primary Key (Never Changes)
loc_name	Location Name
loc_address	Location Address
loc_status	1 - Active, 0 - Not Active
loc_created_by	User who created the location
loc_created_on	Date location Created
loc_lastupdated_by	User who last updated the Location
loc_lastupdated_on	Date location last updated

3. Sample Queries

Sample query to retrieve product price and reduced price through the barcode:

```
SELECT artbarcode,
artsellprice SellingPriceExcl,
ROUND((artsellprice + (artsellprice * 18 / 100)),2) SellingPriceIncl,
artsaleprice ReducedPriceExcl,
ROUND((artsaleprice + (artsaleprice * 18 / 100)),2) ReducedPriceIncl,
artsalebegin ReducedPriceValidFrom,
artsaleend ReducedPriceValidUntil
FROM tb_article
INNER JOIN tb_department ON artdptid = dptid
WHERE artbarcode = '910984'
```

Sample query to retrieve total stock quantity in all locations by barcode:

```
SELECT SUM(stk_qty)
FROM tb_article_location_stock
INNER JOIN tb_article ON tb_article_location_stock.stk_artid = tb_article.artid
WHERE tb_article.artbarcode = ''
```

4. Setup

This section of the manual is intended for iLabMalta Developers to assist in the setup of iLabCentral.

Create Database

The first job is to run the Database Creation Script on SQL Server. Ideally name the database according to company name as follows:

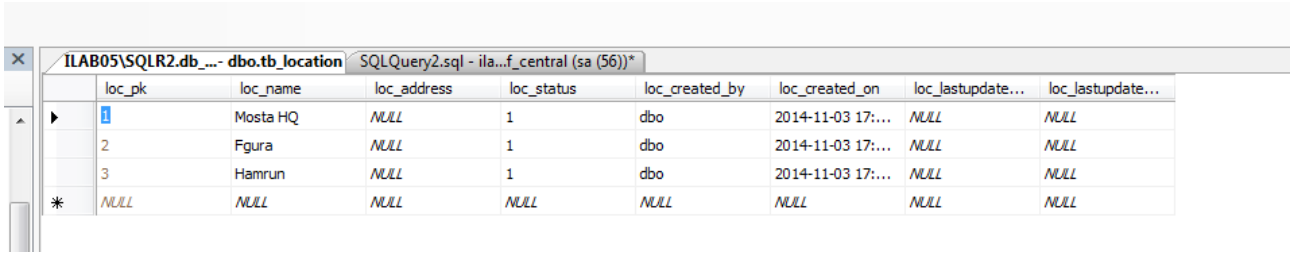
db_XXXX_central

example db_empire_central

Avoid long names for the DataBase. (To rename database in script, you just need to replace the name on top twice. Ignore the drop.)

tb_location

The various locations need to be inserted manually in the tb_location table as follows:



loc_pk	loc_name	loc_address	loc_status	loc_created_by	loc_created_on	loc_lastupdate...	loc_lastupdate...
1	Mosta HQ	NULL	1	dbo	2014-11-03 17:...	NULL	NULL
2	Fgura	NULL	1	dbo	2014-11-03 17:...	NULL	NULL
3	Hamrun	NULL	1	dbo	2014-11-03 17:...	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

You only need to input the loc_name. The rest is inserted by default. Take note of the loc_pk as this needs to be setup for each instance of iLabPOS.

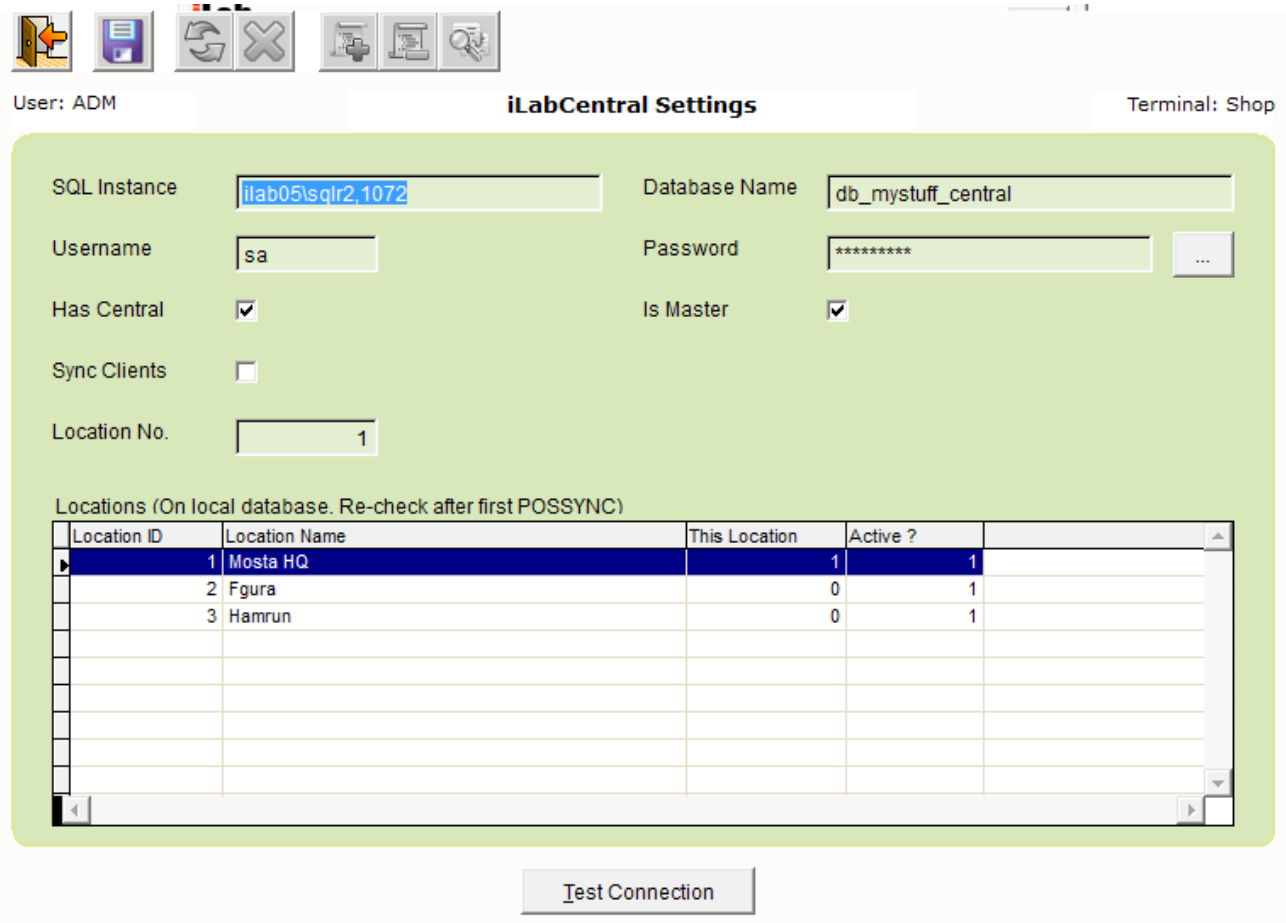
HQ Instance

The HQ Instance is a normal iLabPOS instance, and should have the following settings:

Reginf.IsMaster = 1
Reginf.hasCentral = 1
Reginf.ilc_loc_id = PK for HQ from tb_location
Reginf.ilc_instance = SQL Server Instance Name
Reginf.ilc_database = database name
Reginf.ilc_user = user name with read access to database
reginf.ilc_password = Password
reginf.ilc_syncclient = 0 (Or 1 to Sync Clients and Credit Control)

From Tools you have a configuration Screen to set all above settings

iLabCentral Settings



The above is a typical setup. Syncronisation of Clients is not suggested and speak to Support should you wish this feature.

Make sure you have inserted the locations on the MSSQL Server tb_location first. Then for each Shop/Store, allocate the appropriate the Location No from the loc_pk.

Now copy the POSSYNC.EXE and execute it.

Ensure through POSSYNC.LOG that the process has started and ended.

!!! It is not a problem if the HQ instance has data already inside. !!!

Slave Instances

Slave instances should be empty. It can be a copy of the Master Instance when it was empty.

The slave instance is a normal iLabPOS instance, different from the HQ instance,needs to be setup as follows:

Reginf.IsMaster = 0
Reginf.ilc_loc_id = PK for Slave from tb_location
Reginf.hasCentral = 1
Reginf.ilc_instance = SQL Server Instance Name

Reginf.ilc_database = database name
Reginf.ilc_user = user name with read access to database
reginf.ilc_password = Password (Coded from Tools Menu)

Once setup, execute the possync and ensure that data is exchanged between slave and master.

Task Scheduling

The task for the Server HQ to sync should be set to run on the hour, every hour. Important settings for the task are not to start a new task if the old one is still running, and the startup folder should be set to the iLabPOS Instance folder.

The task for the slave shops should be set to run every hour, on the half hour.

Upgrading From an existing Instance

If a client already has iLabPOS setup and populated with data, follow the setup instructions, including setting up of Location Table and Reginf Settings, and run POSSYNC manually to upload all the initial data onto iLabCentral. Ensure that the Latest version of iLabPOS has also been executed at least once to upgrade the local FoxPro Database.