

## Clarion IP Driver and MS SQL Server

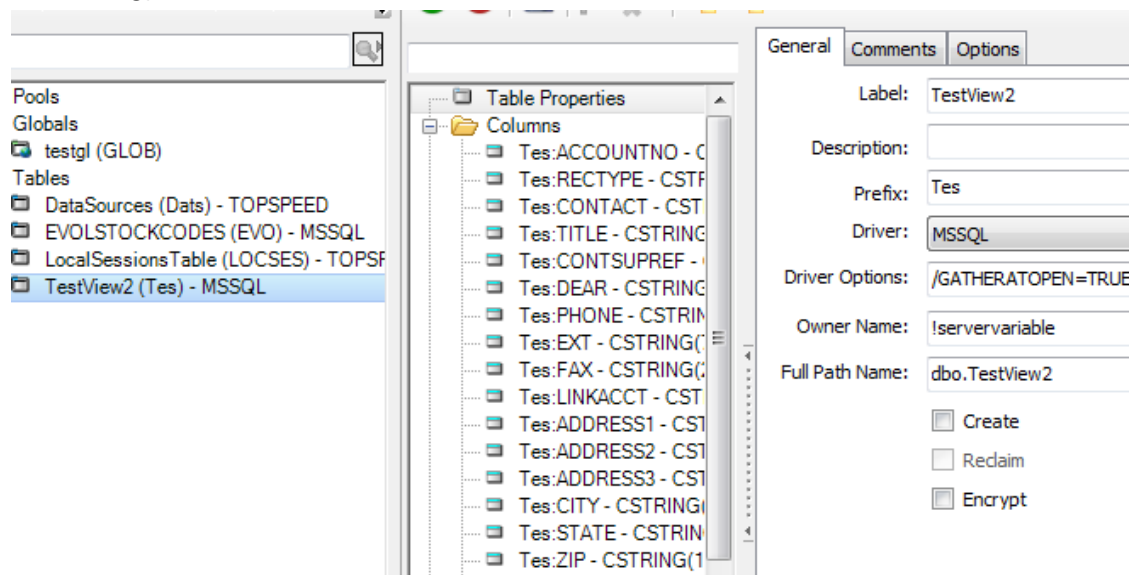
Normally when you create a Clarion programme that accesses an SQL Server you would define the details of the connection string in the owner field of the dictionary properties for the table. This can be hard coded or a variable.

One of the issues that's been a problem with the Clarion IP Driver/ Server has been the inability to change the sql connection string at runtime.

By exhaustively checking various options I have found a solution that appears to work and allows the programmer to change the SQL Connection string at runtime.

**Tip** - Don't define the variable as part of the dictionary's global variables.

In my example I'm using a variable called **servervariable** (this is defined in the data dll global as a string)



### IP Driver Variables

To change variables on the Clarion IP Server you will need to define the variables (global) in your IP Data DLL. In addition you will need to allow them to be accessed/changed by the IP Client Application (to do this you will need to use the extended features of the IP Driver)

You will need five variables in all (of course you can use names that are meaningful to you)

ServerVariable String - use this as the owner attribute for the sql table/s

DBServer - String/Cstring - the details/name of the SQL Server

DBDataBase - String/Cstring - the name of the SQL Database

DBUsername - String/Cstring - the user name to be used to connect - in my case I created a separate user clarionipserver in the SQL Server for the Clarion IP Server to use for connection

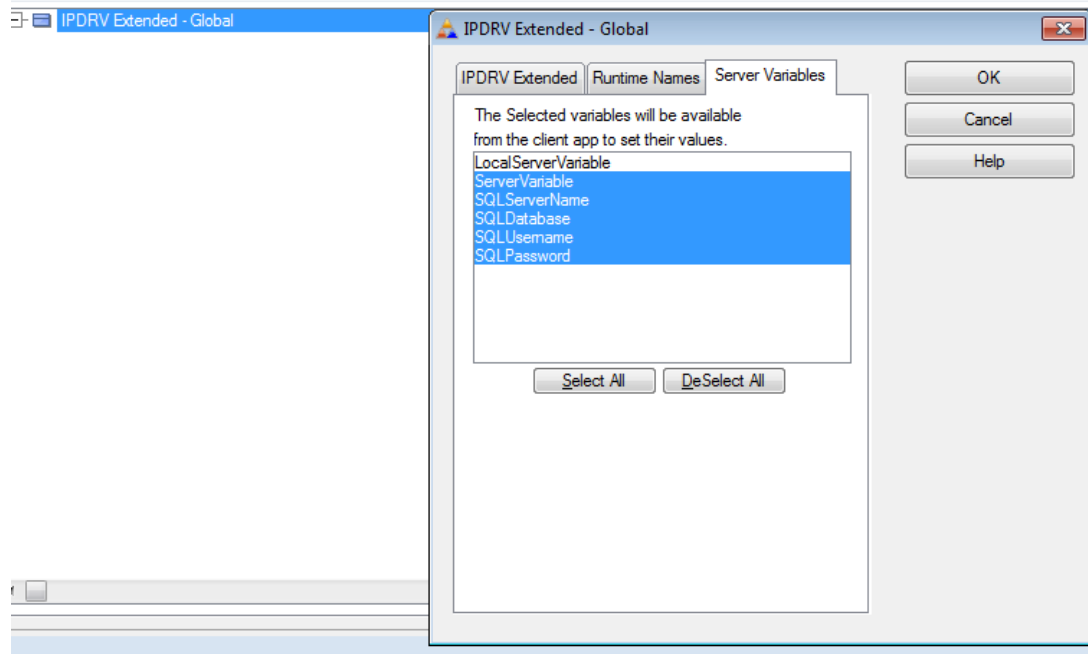
DBPassword - String/Cstring - the password used to access the SQL Server

\* Optional - you'll need a variable for the DBport if you're not using the default SQL TCP/IP port of 1433

## Clarion IP Driver and MS SQL Server

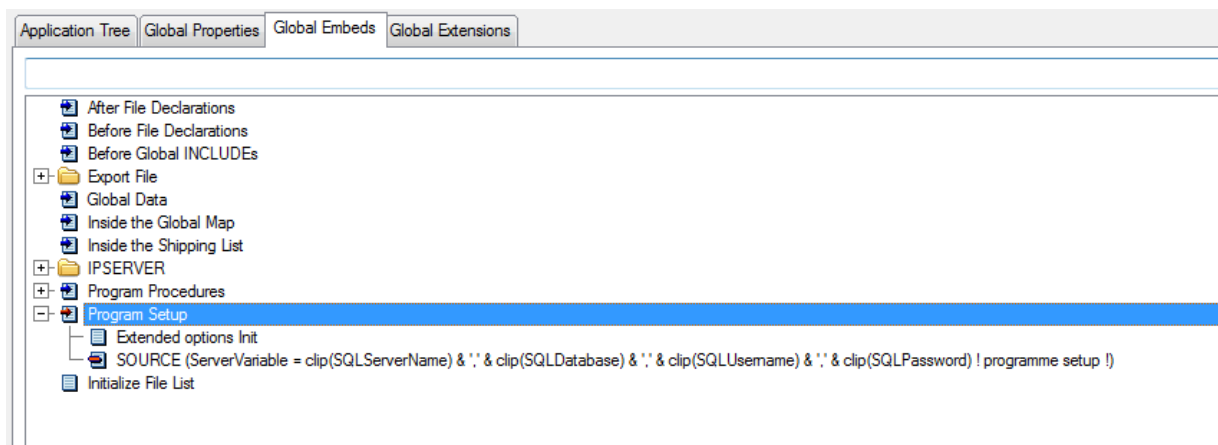
### IP Data DLL

In the IP Data DLL after creating the variables, using the IPDRV Extended options make the variables available for the IP Client Application



IP Data DLL - Embed Point for concatenating the variables and assigning to the ServerVariable

Global Embeds > Program Setup > after Extended Options Init  
Pass the values to the ServerVariable here.

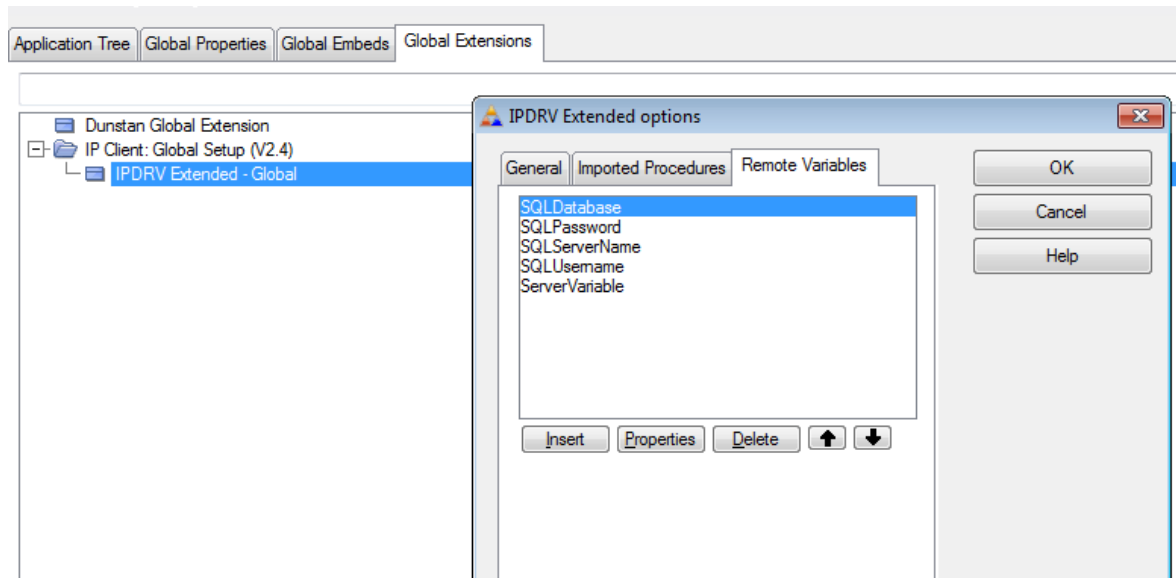


That's all that's required for the Data DLL, next the IP client will need some settings defined.

## Clarion IP Driver and MS SQL Server

### IP Client Application

In the IP client application you'll need to check that the IP Data DLL variables are available. This is available as a global extension - IP Client - IPDRV Extended



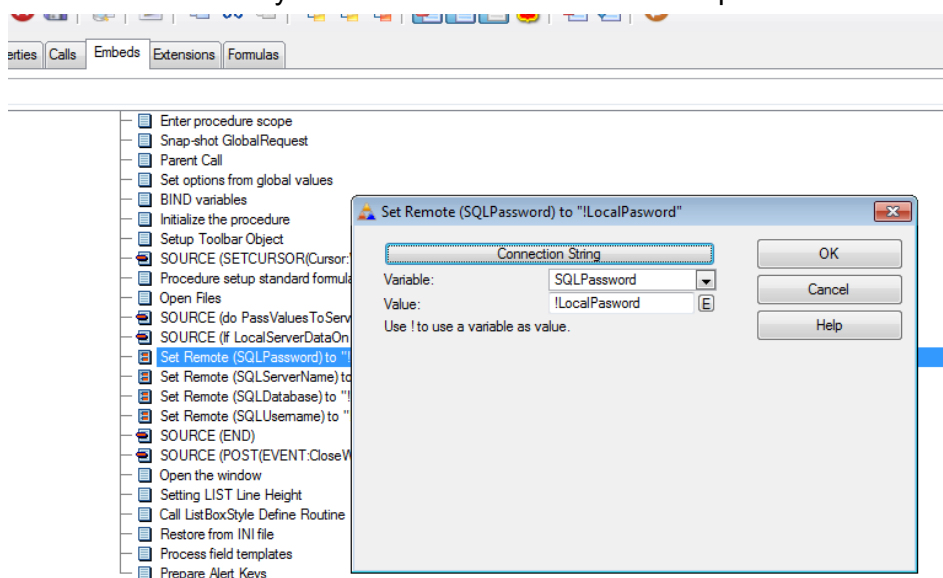
### Assigning the variables

There are many ways you can store and retrieve the data to assign to the variables; INI File, Local Data File, TPS File on the IP Server and many other options.

In my testing I decided to use an IP Driver enabled TPS file to store the details for the SQL Server. **TIP** - makes sure you open and retrieve the data from this file and set the variables on the IP Server prior to opening the SQL Table

### Setting the Variables on the IP Server from the Client

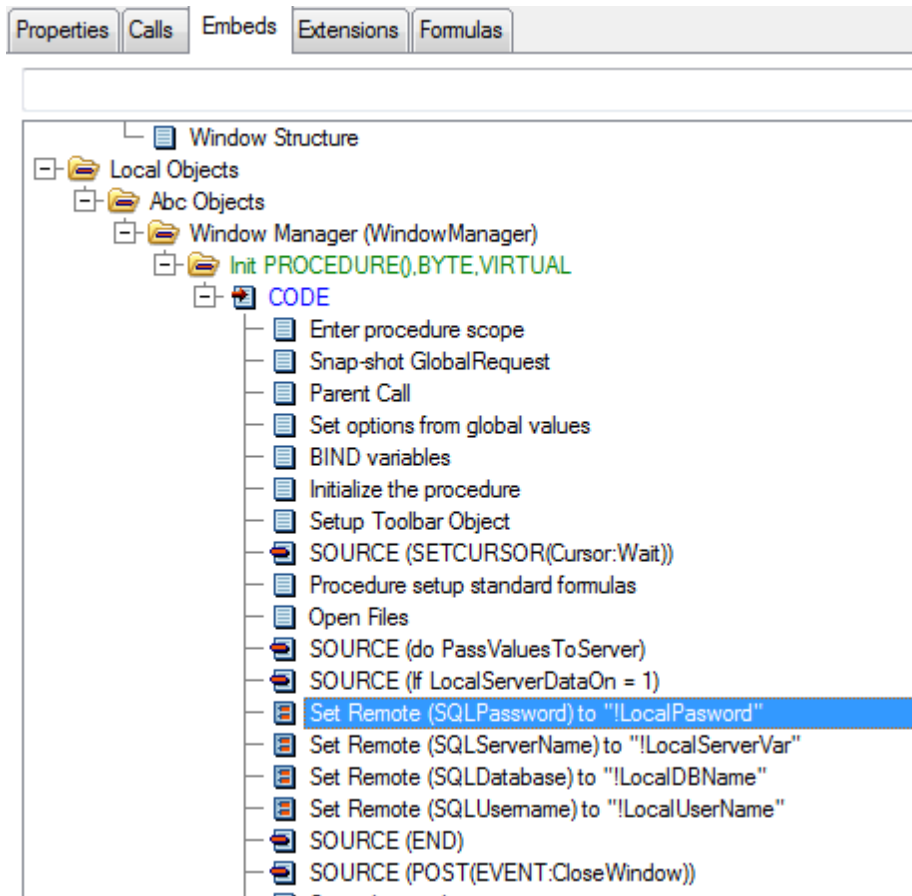
To set the variables you can choose to use the IP template extension "set remote"



## Clarion IP Driver and MS SQL Server

---

In the example I call the Data settings table to retrieve the variable data after the open files embed in the Main procedure. (I'm not opening the SQL Table until I select a browse containing the data).



### Things to know

Trying to set the ServerVariable on the IP Client doesn't work. The IP Server appears to treat the comma separated values of an SQL connection string as different parameters and only takes the first value (the SQL Server Name)

Make sure that the SQL Server security settings allow the SQL user you've defined for the IP Server are valid and have the correct access attributes

Changing SQL Servers and SQL Databases has been tested and works.

### Why Use SQL and IP Driver?

There have been a number of discussions around this. We have a number of IP Driver apps that sit well in the small business/commercial space. In this instance a client needed to have client details imported in real time from their MS SQL based CRM.