**INSYS.WORKER / INSYS BACKGROUND SERVICE**

**(BAGWIS)**

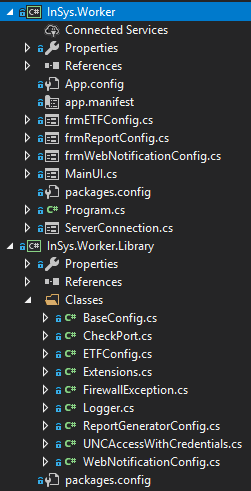
**TABLE OF CONTENTS**

1. InSys.Worker & InSys.Worker.Library
2. Services
   1. ETF Service
   2. Report Generator Service
   3. Web Notification Service

**Rossu M. Belmonte**

Systems Developer

1. **INSYS.WORKER & INSYS.WORKER.LIBRARY**

**** InSys.Worker / InSys Background Worker AKA **BAGWIS** is a control panel for the window services used by the InSys.Suite. This application is built to lessen the processes inside the InSys.Suite.

These 2 projects are both essential to each other and the services they work with. The InSys.Worker is the face of the Control Panel. This is where all the configuration and setup are found. While the InSys.Worker.Library is the storage making sure all the configuration are stored properly so the services can pull the data without any problem.

* **frmETFConfig** – This is the UI for the configuration of the ETF
* **frmReportConfig** – This is the UI for the configuration of the Report
* **frmWebNotificationConfig** – This is the UI for the configuration of the Web Notification
* **MainUI** – This is the control panel where you can Stop, Start and Open the configuration of the services.
* **ServerConnection** – This is the UI for the configuration of the database of the InSys.Suite.
* **Build & Debug** – Since all the process are services you may need to debug 1 by 1 of it. Remember that when cleaning this application all the configuration you made will return to default settings defined in the **App.Config**. You may encounter error when rebuilding / building because of the following reasons:

The **InSys.Worker** is open.

**MainUI**

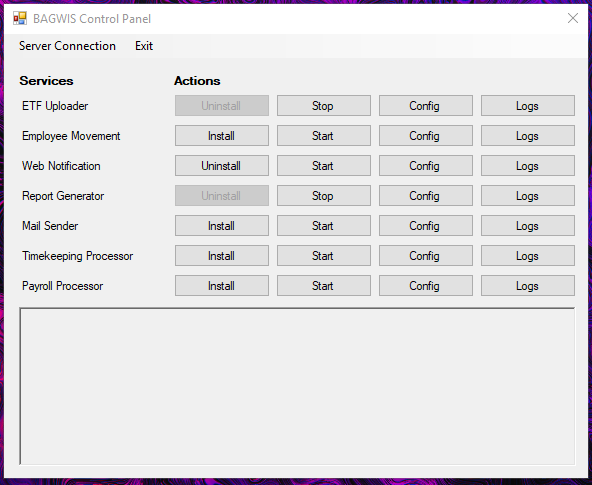


Figure 1 : BAGWIS Control Panel

**ServerConnection – SQL Server Tab**

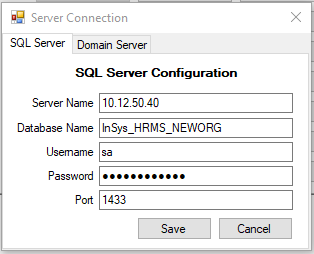


Figure 2 : Sql Server Connection of the InSys.Suite

**ServerConnection – Domain Server Tab**

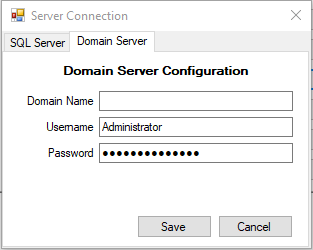


Figure 3 : Account of the Server where the Contents is located.

**frmETFConfig**

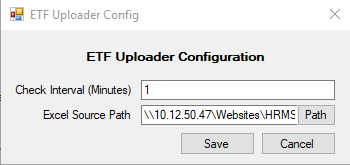


Figure 4 : Configuration for the ETF

**frmWebNotificationConfig**

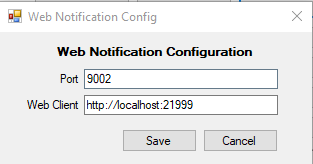


Figure 5 : Configuration for the Web Notification

**frmReportConfig – SQL Configuration Tab**

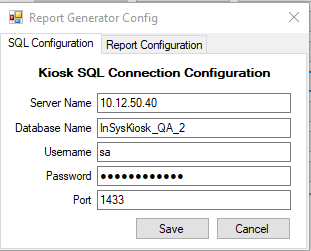


Figure 6 : In case InSys.Kiosk database is still separated from InSys.Suite provide the sql connection.

**frmReportConfig – Report Configuration Tab**

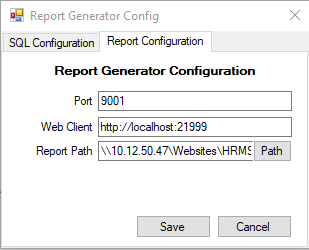
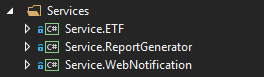
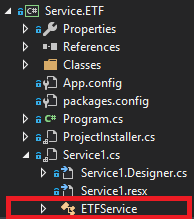


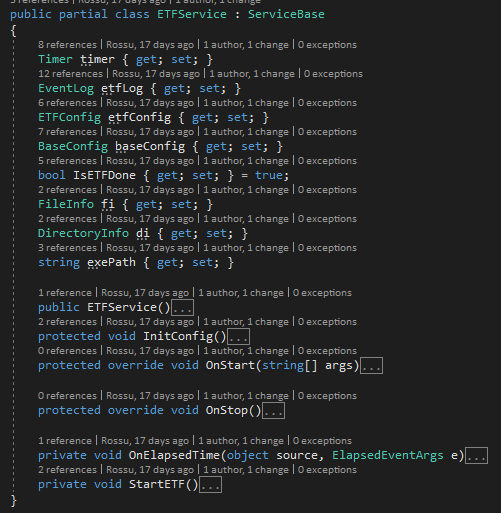
Figure 7 : Configuration for the Report Server

1. **SERVICES**



* **ETF Service** – This is the service that uploads ETF template files in the Initial Migration module in the InSys.Suite system.

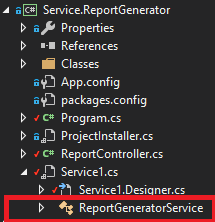
****

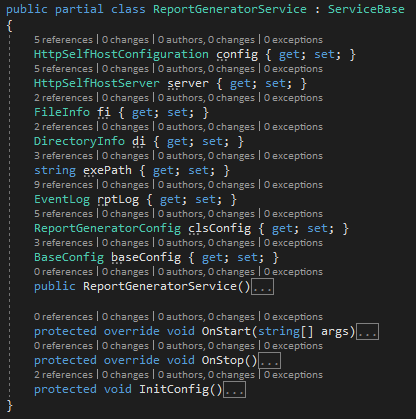


* + **ETFService** – This initializes the logger configuration and the timer.
  + **InitConfig** – This initializes the configuration for sql connection and etf configuration
  + **OnStart** – This is the method called when starting the service. This is responsible for starting the **Timer**.
  + **OnStop** – This is the method called when stopping the service. This is responsible for stopping the **Timer**.
  + **OnElapsedTime** – This is the method called when the timer hits the interval. This is where the process for uploading the etf is called.
  + **StartETF** – This is the process called on the **OnElpasedTime**.
* **ETF Service Build & Debug** – Remember this is a service, meaning when debugging, you need to recreate this service as a windows form application in order to debug or use the logger to find the error or missing values. When building this service it is already path to the **InSys.Worker > bin > debug > Services** folder. When building / rebuilding, you may encounter some errors because of the following reasons:
  + The Service is still Running
  + **Invalid Username and Password** – The cause is the configuration in **Domain Server**, under the **Server Connection** menu. The ETF Service connects to a specific path configured in the **Config** of the **ETF Service**.

After building / rebuilding the service, you need to **uninstall then install** again the service.

* **Report Generator Service** – This is the service that generates the PDF File and Excel File Report for the End-User.

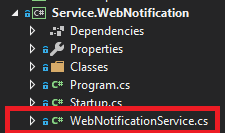
****

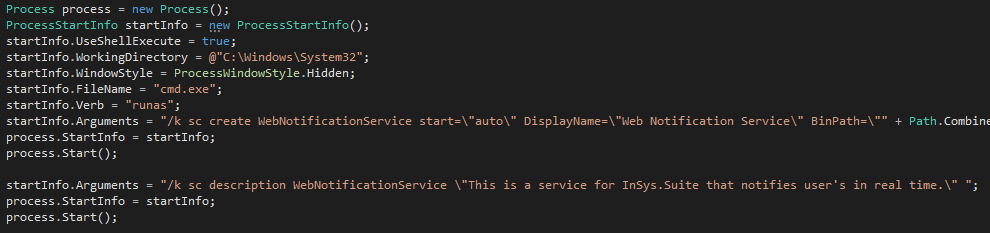


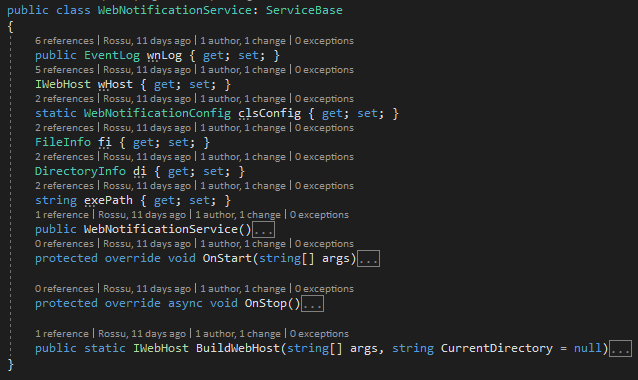
* + **ReportGeneratorService** – This initializes the logger configuration.
  + **InitConfig** – This initializes the configuration of the **ReportGenerator**. This service is a self-hosted Api.
  + **OnStart** – This is the method called when starting the service. This is responsible for hosting the Api.
  + **OnStop** – This is the method called when stopping the service. This is responsible for stopping the hosted Api.
* **Report Generator Service Build & Debug** – Remember this is a service, meaning when debugging, you need to recreate this service as a **Web Api** in order to debug. When building this service it is already path to the **InSys.Worker > bin > debug > Services** folder. When building / rebuilding, you may encounter some errors because of the following reasons:
  + The Service is still Running
  + **Invalid Username and Password** – The cause is the configuration in **Domain Server**, under the **Server Connection** menu. The Report Generator connects to a specific path configured in the **Config** of the **ReportGenerator**.

After building / rebuilding the service, you need to **uninstall then install** again the service.

* **Web Notification Service** – This is the service that notifies the users and watch for connecting users to the server. This is a **.NetCore 2.2** application it is basically not a service because the service template does not exist for **.NetCore** **2.2**. This is created using the command **sc create** through the **BAGWIS Control Panel**.

****





* + **WebNotificationService** – This initializes the logger configuration.
  + **OnStart** – This is the method called when starting the service. This is responsible for hosting the SignalR Server.
  + **OnStop** – This is the method called when stopping the service. This is responsible for stopping the SignalR Server.
  + **BuildWebHost** – This is the method responsible for creating the **Web Host** and calling the **Startup** to initialize the SignalR Server and the Sql Dependency for the service.
* **Web Notification Service Build & Debug** – Remember this is a service, meaning when debugging, you need to recreate this service as a **Web Site** or use the logger to find the error or missing values in order to debug. When building this service you need to **publish** this. I mean it **PUBLISH** not build / rebuild. It is already path to the **InSys.Worker > bin > debug > Services** folder. When **publishing**, you may encounter some errors because of the following reasons:
  + The Service is still Running

You may also encounter an error when starting the service immediately after stopping it. You may need to wait for a minute before starting again, or change the port if some application is already using it.

After **publishing** the service, you need to **uninstall then install** again the service.