**DAAS DATA CAPTURE – INSTALLATION AND SETTINGS**

**New Table DAAS\_EXPORT**

This table keeps track of the MD5 hash key of the JSON messages already sent. Its SQL script:

*CREATE TABLE [dbo].[DAAS\_EXPORT](*

*[DAAS\_KEY1] [varchar](15) NOT NULL,*

*[DAAS\_KEY2] [varchar](15) NOT NULL,*

*[DAAS\_KEY3] [varchar](15) NOT NULL,*

*[DAAS\_KEY4] [varchar](15) NOT NULL,*

*[DAAS\_KEY5] [varchar](15) NOT NULL,*

*[DAAS\_SET\_NAME][varchar](15) NOT NULL,*

*[DAAS\_MD5] [nvarchar](100) NOT NULL,*

*[DAAS\_UPDATE\_TIME] [bigint] NOT NULL,*

*CONSTRAINT [PK\_DAAS\_EXPORT] PRIMARY KEY CLUSTERED*

*(*

*[DAAS\_KEY1] ASC,*

*[DAAS\_KEY2] ASC,*

*[DAAS\_KEY3] ASC,*

*[DAAS\_KEY4] ASC,*

*[DAAS\_KEY5] ASC,*

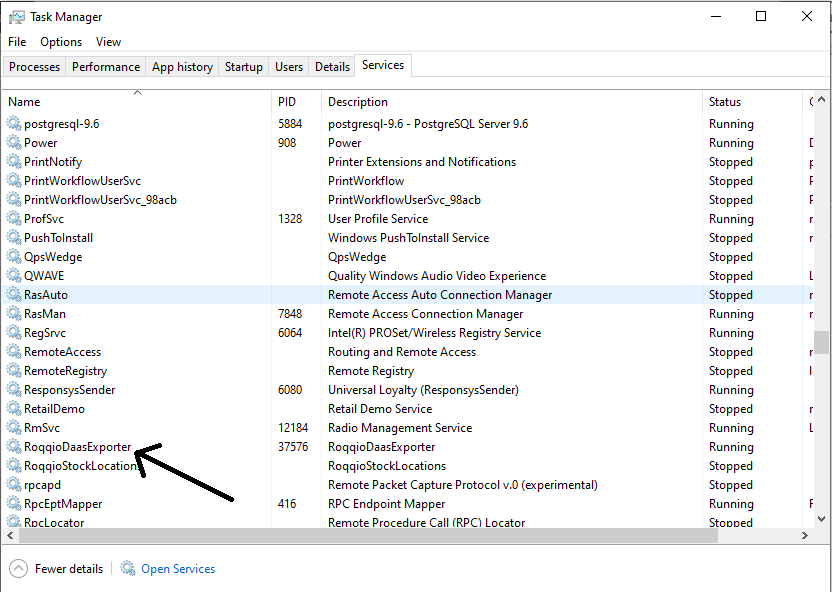
*[DAAS\_SET\_NAME] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

**Installation**

The binaries consist of one executable, **PolitixDaas.exe**, the DLLs **RdeIniFile.dll, Apache.NMS.dll** and **Apache.NMS.ActiveMQ.dll** and the configuration files **PolitixDaas.ini** and **PolitixDaas.exe.config**. The installation is very simple, just create a directory in the C drive and copy the binaries there, together with the configuration files**.** The service is installed running the executable, **PolitixDaas.exe** with -i as a parameter, and it is uninstall running it with the parameter -u, in both cases with administrative rights.

****

When installed, it will appear in the Services list as “RoqqioDaasExporter”.

**Settings**

The settings are in the file **PolitixDaas.ini**. Its contents:

*[SMTP]*

*host=smtp.gmail.com*

*username=PANCHOVILLA@gmail.com*

*password=XXXXXXXX*

*port=587*

*tls=1*

*emailfrom=francesco.stambecchi@gmail.com*

*recipients=rodolfod@futura4retail.com.au,rodolfothegreat@hotmail.com*

*emailsubject=Politix Daas...*

*[SQL]*

*server=futura-sql.politix.com.au*

*database=* *FAS\_AU\_HO\_DEV*

*ersdatabase=FUTURA\_DEV*

*userid=xxxxxxxxx*

*password=xxxxxxxxxx*

*osauthentication=0*

*[SYSTEM]*

*SalesDateFrom=0*

*SalesDateTo=0*

*ShipmentFromDate=0*

*ShipmentToDate=0*

*ActiveMQUrl=futura-dev.politix.com.au:61616*

*intervalmins=3*

*MinSendDate=20190101*

*Debug=1*

*ResultSet=0*

*LocationUpdate=0*

*ProductUpdate=0*

*PriceUpdate=0*

*PermanentMarkdownUpdate=0*

*SaleUpdate=0*

*InventoryUpdate=0*

*TransfersFromHOUpdate=0*

*OrdersUpdate=0*

*ShipmentsUpdate=0*

*InventoryAdjustmentsUpdate=0*

*LocationModule=1*

*BlockLocations=1*

*BlockProducts=1*

*BlockPrices=1*

*BlockSales=1*

*BlockPermanentMarkdowns=0*

*BlockInventory=1*

*BlockInventoryAdjustments=1*

*BlockTransfersFromHO=1*

*BlockTransfersFromBranches=1*

*BlockOrders=1*

*BlockShipments=1*

*[Queues]*

*Orders=Orders*

*Locations=Locations*

*Prices=Prices*

*Products=Products*

*Sales=Sales*

*PermanentMarkdowns=PermanentMarkdowns*

*Inventory=Inventory*

*InventoryAdjustments=InventoryAdjustments*

*TransfersFromHO=TransfersFromHO*

*TransfersFromBranch=TransfersFromBranch*

*Shipments=Shipments*

In the INI file we have these sections:

**SMTP**

SMTP server details. The service emails the logged errors at the end of a cycle. It works with TLS.

**SQL**

MS SQL server details. At the moment the service is not compatible with other database systems.

**SYSTEM**

The parameters in this section:

* Intervalmins: Interval time in minutes
* StartingDate: The format is YYYYMMDD. Data older than that setting will not be considered. When installed in Production it should be set to the installation date or few days before.
* ActiveMQUrl Url including the port number of the used Active MQ.
* Debug: When set to 1, the generated log file contains more information.
* ResultSet: Used during development, it limits the number of rows returned by SQL statements. When set to 0 (Default), there is no limit to the result sets.
* LocationUpdate: When set to 0, the service returns all the locations instead of just the delta.
* ProductionUpdate: When set to 0, the service returns all the production items instead of just the delta.
* PriceUpdate: When set to 0, the service returns all the item prices instead of just the delta.

**QUEUES**

Names of the queues.

**LOGGINGS**

The service creates a “log” directory in the installation path. There are two kind of log files. logYYYYMMDD.txt and errorlogYYYYMMDD.txt, where YYYYMMDD is the current date. The service includes a housekeeping, all log files older than three months are deleted.