Prudent Management Associates

Portal Documentation

January 2016

The Prudent Management Associates (PMA) portal uses 5 related projects written in Java and using several frameworks, including Spring, Hibernate and Freemarker. These are all popular open source systems in wide use in software development. The systems is organized using Maven a software project management and comprehension tool.

The system is actually quite simple. A nightly job converts client username, password and account information from the existing PMA system to a new set of tables in the PMAMASTER database located on the SQL Server of the PMAWEB server.

The web application presented to PMA Clients is a Java application (PMAWebApp) running on a Tomcat server located on the PMAWEB server. Tomcat was set up as a service and starts automatically whenever the server is restarted. The PMAWebApp uses a combination of data from the PMAMASTER database for security and the PMAData database for client account information.

# The Source Code – 5 main parts

* 1. PMAData – Java Models of PMA internal data that is updated via your VB Application including the new Reports table that contains information about the PDF files. This also include client account information such as holdings.
  2. PMAWebData – Java Data project to access the PMAMASTER database which has security, client questions and answers and account associations
  3. DataTransfer – Contains the Java code to copy data from PMAData to PMAWebData. Currently runs twice daily.
  4. PMAWebApp – Java code for the portal. This uses PMAWebData and PMAData.
  5. PMAMasterReport – Used to develop tests to retrieve PDF files. Not used in the Web Portal Application

# Nightly Batch job

1. Transfers data nightly from PMAData database to PMAMASTER database
2. To run change to the java directory
3. The command to run is “**Java –jar DataTransfer.jar**”
4. Currently scheduled to run at 1am and 11:57am but could be updated to run more frequently

# Database

1. PMAMASTER – used for portal, contains tables for security and to associate users with accounts
   * 1. Users, Passwords
     2. Client Questions
     3. Client Answers

## WebClient

This table holds the information for clients who log into the web site. It currently contains so me additional fields which could be used for enhanced security. This table if filled by the DataTransfer program

**CREATE TABLE [dbo].[WebClient]**(

[userId] [numeric](19, 0) IDENTITY(1,1) NOT NULL,

[accountNonExpired] [tinyint] NOT NULL,

[accountNonLocked] [tinyint] NOT NULL,

[credentialsNonExpired] [tinyint] NOT NULL,

[enabled] [tinyint] NOT NULL,

[masterClientName] [varchar](255) NULL,

[masterClientNumber] [varchar](255) NULL,

[password] [varchar](255) NOT NULL,

[username] [varchar](50) NOT NULL,

PRIMARY KEY CLUSTERED

(

[userId] ASC

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

UNIQUE NONCLUSTERED

(

[username] ASC

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

) ON [PRIMARY]

## ClientAnswer

This table holds the client’s answers to the security questions. The question ID is linked to the PotentialQuestion table. If you need to reset a client’s answers. Find the rows in this table the client using the userId field and remove them.

**CREATE TABLE [dbo].[ClientAnswer]**(

[answerId] [numeric](19, 0) IDENTITY(1,1) NOT NULL,

[answer] [varchar](255) NOT NULL,

[userId] [numeric](19, 0) NOT NULL,

[question\_questionId] [numeric](19, 0) NULL,

PRIMARY KEY CLUSTERED

(

[answerId] ASC

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

) ON [PRIMARY]

CREATE TABLE [dbo].[ClientRole](

[roleId] [numeric](19, 0) IDENTITY(1,1) NOT NULL,

[authority] [varchar](255) NULL,

[userId] [numeric](19, 0) NULL,

PRIMARY KEY CLUSTERED

(

[roleId] ASC

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

) ON [PRIMARY]

PotentialQuestion

This table holds the potential questions that a client can choose for the security questions. If you wish to add a question you should NOT reuse an existing id because if the client has already associated an answer with this question it will no longer make sense.

**CREATE TABLE [dbo].[PotentialQuestion]**(

[questionId] [numeric](19, 0) NOT NULL,

[question] [varchar](255) NOT NULL,

PRIMARY KEY CLUSTERED

(

[questionId] ASC

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

UNIQUE NONCLUSTERED

(

[question] ASC

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

) ON [PRIMARY]

There are several other tables in the PMAMASTER database however they are not currently being used by the portal application.