**PopMedNet Build:**

Building PMN for Development and Deployment

## 

Version 0.1

September 20, 2016

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Description | Modified by |
| September 20, 2016 | 0.1 | First draft | Daniel Dee |

Contents

[PopMedNet Architecture 4](#_Toc462130588)

[Prerequisites 5](#_Toc462130589)

[Web Server 5](#_Toc462130590)

[Database Server 6](#_Toc462130591)

# PopMedNet Architecture

The PopMedNet application is composed of 3 web components:

1. Portal – This covers the UI and supports the legacy PMN components.
2. API – This is the web service that supports all PMN functionalities. (Also known as DNS API.)
3. Resources – This contains the shared files and scripts. In theory, this can be shared by multiple PMN instances with the same version, but it has not been put in practice.

In addition, it depends on the following external components for certain services:

1. Windows Task Scheduler – Used to periodically hit the API to send off scheduled notifications via email.
2. Quartz – This is a scheduler that is gradually being deprecated. It currently performs two tasks:
   1. Hits the API to scan and collate request metadata.
   2. Executes scheduled legacy requests.

# Minimum Tool and Framework Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools and Frameworks** | **Ideal** | **May Work** | **Note** |
| Visual Studio | 2015 | 2015 |  |
| Web Server (IIS) | 8.5 or newer  or use Visual Studio built-in IIS Express | 7.5 | 1. You will need a wildcard SSL certificate for your domain. (\*.YourDomain.com)    * This is because there will be multiple sites that intercommunicate. 2. You will need to add the following Roles and Features to the webserver: |
| .NET Framework | 4.6.1 | 4.6.1 |  |
| Entity Framework | 6 | 6 |  |
| Razor Generator |  |  | Install I<https://github.com/RazorGenerator/RazorGenerator> |
| TypeScript | 1.8 | 1.8 | https://www.typescriptlang.org |
| ASP.net MVC 4 |  |  | https://www.microsoft.com/en-us/download/details.aspx?id=30683 |

## Database Server

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools and Frameworks** | **Ideal** | **May Work** | **Note** |
| SQL Server | 2012 | 2008 R2 | 1. You will need to enable the Filestream and Common Language Runtime features.    * How to enable Filestream: <http://msdn.microsoft.com/en-us/library/cc645923.aspx>    * How to enable CLR: <http://msdn.microsoft.com/en-us/library/ms131048.aspx> 2. We recommend using [Transparent Database Encryption](https://msdn.microsoft.com/en-us/library/bb934049.aspx) ***if*** you have “encryption at rest” requirements. |
|  |  |  |  |

# Building the Web Portal

To build a complete PopMedNet deployment package, you need to build the PMN Web Portal and the DataMart Client.

The Web Portal consists of 3 components. Using Visual Studio, set the Build Configuration to Debug or Release:

1. Resources: Open and build Lpp.Pmn.Resources solution.
2. API: Open and build Lpp.Dns.Api solution.
3. Portal: Open and build DistributedNetworkSolution solution.

Visual Studio deposits the binary executable in the “bin” folder of the respective solution.

At this point, you can run PopMedNet locally within Visual Studio assuming that you have a proper PMN database and that the connection strings are setup properly. Use the ConnectionStrings.config-template in Lpp.Dns.Api and Lpp.Dns.Portal (DistributedNetworkSolution) to create proper ConnectionStrings.config files.

# Building the DataMart Client

The DataMart Client consists of the DataMart Client Core and the Adapter Packages. Using Visual Studio, set the Configuration to Debug or Release:

1. Open Lpp.Adapters solution.
2. For any Build Configuration other than Debug, in each Adapter Project under the solution (Lpp.Dns.DataMart.Model.\*), look for “SignFile” and update it to match your certificate. (You can also remove it, but this is used to sign the Assemblies for Windows to accept the download as from a known source.)
3. Build Lpp.Adapters. This will build the Adapter packages to the App\_Data folder of the API.
4. Right click the Lpp.Dns.DataMart.Client project, select the “Publish” tab, specify the “Publishing Folder Location” and publish. Copy the contents of the published folder to the Portal/DataMartClient folder.

# Deploying PopMedNet to IIS

The 3 components of the PMN Web Portal is built to their respective “bin” folder. The Adapter packages are built, packaged, and deployed directly to Lpp.Dns.Api/App\_Data on successful build of each adapter project. The DataMartClient installer folder should have been copied to Portal/DataMartClient.

To deploy PopMedNet, publish the following:

1. Resources: Right click on Lpp.Pmn.Resources and select Publish.
2. API: Right click on Lpp.Dns.Api and select Publish.
3. Portal: Right click on Lpp.Dns.Portal and select Publish.

Select Custom and create Profile in each case. You can choose to deploy directly to IIS or to a local folder.

Please follow the instructions described in the document “Deploying a PMN Environment V2” under the section Brand New Deployment, and do the following in order:

1. Restore the PMN database, either a starter one or a backup.
2. Launch Resources, Api and Portal in order.
3. Note that once the Api is started, the site should be visited prior to starting the Portal so that database migrations are executed to upgrade the database if necessary to the latest version, and any migration errors will be visible.