© 2016, Michael E. Legatt, Ph.D., Electric Reliability Council of Texas, Inc.

README: Macomber Map Open Source 2.0

This project contains an updated version of the Macomber Map Open Source Project, which utilizes WCF to build a client/server architecture.

# Thanks

First and foremost, this project would not be possible with the countless hours and energies of control room operators and support engineers at ERCOT and SPP’s control rooms. While we wrote code, they took the time to help us understand what we needed to give them to best set them up for success. They never gave up on us, and always trusted us with their questions and advice. This is their tool – by and for them.

This project never would have started without Gary Macomber, and his dedication and vision. We honor you with this work, and hope you would be proud of it. To his wife and daughter, thank you for sharing him with us at ERCOT.

So many people have played roles in the development, building, and testing of the Macomber Map across almost a decade. Please see the client’s about menu for a full list. I’d like to single out and thank Cody Parker, Jeff Parker, and Tim Van Prayoon at SPP, who have worked with their operators to so significantly enhance the tool. If you look at the graphics and performance the map now provides, and say, “wow!” – please thank them for it. On the ERCOT side, Murali Boddeti, Paul Li, Yong Ma, Helen Li, and Mike Duffy have done the same, enhancing the map for our blackstart training, and supporting its growth into the future. Murali Boddeti build TEDE (The ERCOT Data Extractor) to efficiently move data between our core systems and the Macomber Map Server. Richard Howard and Steve Daniels, thank you for your vision and leadership.

# Implementation Notes

In order to adapt the Macomber Map to your needs, the following steps should be completed:

* If you plan on using encryption for database connectivity or other functions, generate an IV (Initialization Vector) and Key for the Triple DES algorithms in MacomberMapClient.Data\_Elements.SystemInformation.MM\_Data\_Source and MacomberMapIntegratedService.Oracle.MM\_Entryption, and place in both Decrypt(String EncryptedText) and Encrypt(String ClearText).
* Edit the settings in MacomberMapIntegratedService, pointing to the folders containing model files, one-lines, log files, ports, etc. Do the same for MacomberMapClient and MacomberMapAdministrationService, and match up the WCF ports for client<->server communications. Note that for this demo, M:\ is used as the base folder.
* If you are assigning area permissions, update the LoginOperatorships.csv file, and place in M:\Examples. Note that the area ‘ERCOT’ are seen as the master area, while other areas allow control within the one-lines only if you have an operatorship match with one of the areas specified in that file. For example, if I were to log in, and it validated the line “MLEGATT,A,B,C” then I could only operate equipment with operatorships of A, B, or C, but if I were “MLEGATT,ERCOT”, then I could operate all equipment.
* If you have a client and server on different subnets or segments of your network (and a broadcast UDP packet can’t reach from server to client), put those server names comma-separated in the ‘ServersToPing’ settings on the MacomberMapClient (and on the MacomberMapAdministrationConsole if needed).
* If you plan on sending control commands from a one-line control panel, add those commands and configuration in MacomberMapConfiguration.xml for those particular components. If you want commands to be sent from the generator control panel, update the SendUnitValues() in MacomberMapClient.User\_Interfaces.OneLines.MM\_Generator\_ControlPanel.

# First-time Startup Instructions

* Unzip the Macomber Map Open Source archive to a folder on your system.
* Note that right now, the server is set to use relative paths (e.g., ..\..\..\Examples\OneLines) to assist with ease of startup. Please feel free to change as needed, and note that the folder structure is anticipated to be like MacomberMapClient\bin\x64\Debug.
* If you want to create your own branding for the components of Macomber Map (including the company logo that appears on both the client startup and the main map, and the icons throughout the program, edit the CompanyLogo PNG and ICO images in BrandingSupport, then run “ApplyBranding.bat”.
* Run the Visual Studio project. At this point, the Server (MacomberMapIntegratedService), Administrator console, TEDE simulator and the Macomber Map Client.
* A randomly-generated (and ugly – my apologies!) sample of the network model will be loaded, in MacomberMap-Combined.MM\_Model when the server starts up.
* If you want to load the simulated state data also, once the Administrator Console is running and it’s found the server, click on the server name in the header bar, and choose ‘Load Server Savecase’. In Examples\System Files, you’ll find MacomberMap-Combined.MM\_Savecase, which you should load. That’s where all the state data comes from. You know it’s successful when some information in the Administrator Console turns green (indicating a status change), such as the Breaker Switch count. Over time, that text will fade back to white.
* The client should pop up and show the local system after the server starts up. Double click on that, and use \*\*MacomberMapTesting\*\* for your login credentials.

If you have any questions or need any assistance, please ask in the user forums at <https://groups.google.com/forum/#!forum/macombermap>, or watch the YouTube videos at <https://www.youtube.com/channel/UCD-eZbsPdIiRDNpj0FTMU8w> (videos coming soon).