How to Create Tribal CDMS

Table of Contents

[Overview 2](#_Toc152945015)

[Instructions 3](#_Toc152945016)

[Frontend 3](#_Toc152945017)

[Backend 5](#_Toc152945018)

[Database 5](#_Toc152945019)

[Prepare SSMS (SQL Server Management Studio) 5](#_Toc152945020)

[Add the necessary structure and data 9](#_Toc152945021)

[Populate the database 9](#_Toc152945022)

[IIS 13](#_Toc152945023)

[Updates 15](#_Toc152945024)

# Overview

This document provides information for how to “strip down” CDMS for the Tribes. As necessary, you may contact CRITFC or CTUIR, if you have any questions.

Assumptions

This document assumes that you have the following software items available and installed.

* Microsoft Visual Studio 2022 (Community Edition will work)
* Git
* NPM
* Microsoft SQL Server Management Studio (SSMS)

# Instructions

## Frontend

* RDC into the Dev server
* Using This PC (or File Explorer) create the location where you want the site to be.
  + **Note:** We will actually place the site a couple folders down, not in this top-level location
  + Example: D:\Websites\cdms-dev-gc
* Add the permissions necessary for the folder to be used as a site in IIS
  + IUSR (full control)
  + IIS-IUSRS (full control)
* In this location, create the following two folders
  + cdms
  + services
* Using a cmd prompt (DOES NOT need to be run as administrator)
  + Navigate to the cdms folder that you created above
    - Frontend example: D:\Websites\cdms-dev-gc\cdms
  + Inside the cdms folder, clone the frontend code from GibHub
    - Git command: git clone [GitHub URL]
    - Example: git clone <https://github.com/CTUIR/cdms-fe>
  + Navigate into cdms-fe
  + Run this command: git branch
  + The results should resemble those in the following image – notice the color (unless you have custom colors); if masterV3 is not green something is not quite right  
    Text

    Description automatically generated
* Using This PC (or File Explorer), copy the config.ts
  + From: Wherever you have it, or can obtain good template
  + To: your Frontend Dev folder   
    (Example: D:\Websites\cdms-dev-gc\cdms\cdms-fe\src\appjsLegacy\config.ts)
* Using Notepad++ (or other text editor)…
  + Open the config.ts file in your Dev
  + Update file as necessary for your system
  + You may contact CRITFC or CTUIR for a starting file
* The frontend is now ready to build
  + Open a cmd window on the server and navigate to the development folder for the site (Example: D:\Websites\cdms-dev-gc\cdms\cdms-fe)
  + If this is the first time the site is being set up, do the following steps
    - Enter the command: npm install
  + In the cmd window, navigate to the src folder (D:\Websites\cdms-dev-gc\cdms\cdms-fe\src)
  + Enter the command: npm run start
    - This script runs ng build -–watch. Every time you save, it will rebuild the deployable files in the Dev folder, which is what goes in the website folder, which is what the site points to in IIS
* Making updates:
  + Open another cmd window and navigate to the src folder (example: D:\Websites\cdms-dev-gc\cdms\cdms-fe\src)
  + Enter the command: git branch
  + If you are not on branch master, you will need to checkout that branch
    - Stash, or add/commit any changes to your current branch
    - Checkout master
  + Create a branch for the bug fix or new feature, where the branch name relates to what the issue is
    - Command format: git checkout -b [NewFeature]
    - Example: git checkout -b DatasetQueryImprovement1
  + You are now ready to make code changes
  + When you are satisfied with the new feature or bug fix, and it has been thoroughly tested, you then push the branch up to the repository in GitHub
  + Once the branch is in GitHub, you will submit a pull request to CTUIR
* **Note: As the frontend is wrapped in Typescript, after making code changes, or updating the config.ts file, the frontend MUST BE built again, before the changes will show in the browser**

## Backend

The backend may be installed on either a local machine, or on the server, whatever the policy is for your organization. For clarity, the example given here is for when you install the backend on your local machine.

* Using This PC (or File Explorer) create the location where you want the backend code to reside
* Using a cmd prompt (DOES NOT need to be run as administrator)
  + Navigate to the services folder that you created above
    - Backend example: C:\CdmsTribal
  + Inside the CdmsTribal folder, clone the backend code from GibHub
    - Git command: git clone [GitHub URL]
    - Example: git clone https://github.com/CTUIR/tribalCdms-be
    - You may rename it to something you prefer if you wish
  + Navigate into tribalCdms-be/services folder
  + Run this command: git branch

The results should resemble those in the following image – notice the color (unless you have custom colors); if masterV3 is not green something is not quite right.

Text

Description automatically generated

* Copy the web.config into the services folder from wherever you have it, or have obtained it
* Using a text editor, update the web.config to what your local settings should be
* You may contact CRITFC or CTUIR for a starting file
* Using Microsoft Visual Studio 2022 (Community Edition or better), open the backend
* Build the services solution; there should be no errors
* Using MS VS, publish the files to the services folder that you created in the Frontend section
* Making updates
  + In the same manner as specified in the frontend section, use git commands to create branches for new features, or bug fixes
  + When the fix or feature is ready, push it to the GitHub backend repository (where you cloned it from)
  + Submit a pull request for the branch to CTUIR for review

## Database

### Prepare SSMS (SQL Server Management Studio)

Administrators may choose to configure the CDMS system however they choose. However, if they use the standard configuration used by CTUIR, it will be much easier to obtain assistance from CTUIR, if/when necessary. This section provides steps to setup SSMS and the database, starting from an empty database. These instructions assume neither database CDMS\_DEV nor user gisptadmin exist on your system.

* In SSMS, if it is not already showing, show the Object Explorer
  + Menu-bar -> View -> Object Explorer (F8)

Graphical user interface, application

Description automatically generated

* Create an empty database called CDMS\_DEV
  + In the Object Explorer, right-click on Databases and select New Database
  + The following form will appear

Graphical user interface, text, application, email

Description automatically generated

* + Enter the database name, shown highlighted in the image above
  + Accept all other settings and click on the OK button at the bottom of the form
  + In the Object Explorer, expand the Databases item and the database CDMS\_DEV should be visible
* In the Object Explorer, go to Security -> Logins, as shown highlighted below

Text

Description automatically generated

* Create a login called gisptadmin with the following properties
  + Right-click on Logins, select New Login
  + General

Graphical user interface, application

Description automatically generated

* + Server Roles

Graphical user interface, application

Description automatically generated

* + User Mapping

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* + Edit the columns in the following way for database CDMS\_DEV
    - Map: Check the box
    - User: Enter gisptadmin
    - Default Schema: dbo
  + Down below, in the Database role membership for: CDMS\_DEV…
    - Check the db\_owner line
  + Click OK.
  + The user gisptadmin should now exist and have full permissions on database CDMS\_DEV
  + We are ready to populate the database

### Add the necessary structure and data

* In VS (Tools -> NuGet Package Manager -> Package Manager Console (PMC), run “Update-Database –SourceMigration Initial –TargetMigration Initial”
  + This action creates all the database tables, including the \_\_MigrationHistory table
  + When this author used the -Script option to review the SQL commands, VS fussed; therefore, he ran it as shown above
* Review SSMS, refresh the db CDMS\_DEV... and verify that it now has lots of tables, including the \_\_MigrationHistory table

### Populate the database

Developers may run all the commands together via the Package Manager Console (PMC), or individually as shown here, referring to the following image.

Text

Description automatically generated

* Steps that will be repeated
  + In VS PMC, run the following command to create and review the first SQL file

Update-Database **-Script** –SourceMigration Initial –TargetMigration Populate1 Populate1

* + - In VA, a tab with SQL commands in it will appear
    - Review the commands to verify nothing looks weird; the commands should be resemble the following image

Graphical user interface, text, application, chat or text message

Description automatically generated

* + Open a tab in SSMS and enter the following command, using the name of the database you created in the Database section:

use [TheNameOfYourDatabase]

GO

* + The GO **MUST** be on the line below the use… line, or SSMS will indicate a problem
  + Copy all of the commands in the tab and paste them into a tab in SSMS, **below** the GO line
  + Execute the commands
    - There should be no errors
    - If there are errors, troubleshooting will be required
* Repeat the steps for the file Popluate2 , which looks like the following

Text, letter

Description automatically generated

* Repeat the steps for the file Popluate3 , which looks like the following

Graphical user interface, text, application

Description automatically generated

* Repeat the steps for the file Popluate4 , which looks like the following

Graphical user interface, text, application

Description automatically generated

* At this point, assuming that you have the following parts installed
  + Frontend code
  + Backend code

## IIS

* Create your website in IIS
* The services folder must be converted to an application
* If you want to run your website with https, you must install an SSL certificate (available from GoDaddy, or other places) on the server
* Once the SSL certificate is installed on the server, go to the site bindings to point the https at the correct certificate
* In order to use one IP address, and stack lots of website on it, use Server Name Indication and refer to the following image

Graphical user interface, application

Description automatically generated

* x

# Updates

|  |  |  |
| --- | --- | --- |
| **Date of Last Update** | **By Whom** | **Brief description of what was updated** |
| 12/8/2023 | George Clark | Initial creation. |
|  |  |  |