**General instructions for setting up the Marginal Tax Rate Calculator (MTRC)**

The MSG Project team will be providing each partner's IT team access to a repository of files that can be used to set up a Marginal Tax Rate Calculator. The first state for which we provided this package of codes was New Hampshire, and we have since provided it to our partners in Allegheny County, PA, as well. We are pleased to provide this package now to Maine.

This calculator is intended for use by low-income residents of your state or jurisdiction, and also can be used in case management settings. It asks users to answer a series of questions about their household characteristics, employment, receipt of public benefits, and expenses, in order to generate outputs that estimate whether an employment change that users anticipate may result in the loss of eligibility for a range of safety net programs. It is not intended to guarantee whether or not a benefit recipient will remain eligible for a particular program, but rather more as a financial literacy tool that can indicate whether this may be a concern as they seek to assess a potential change in their financial situation. The calculator will provide estimates.

In order to spin this tool on your website, you will need a server the can run the following software languages and programs:

PHP

Perl

MySQL

JavaScript

HTML

HOW THE TOOL FUNCTIONS:

The repository for which MSG will be providing you access is set up to run this tool via [root]/me.php, the main page for the Maine tool. Here and below, the files contained in "[root]" are available at family-resource-sim-mtrc/tools/frs\_stage/ in the BitBucket repository. Downloading the repo will require you to change that root path to make your tool operational.

The user interface is written in flat PHP, with some JavaScript for field validation checks. The PHP interacts with a MySQL database only to generate outputs from user inputs; no user inputs are ever logged in the MySQL database. The PHP collects both user inputs and outputs generated by the MySQL calls as $\_SESSION variables, that are then saved as a Perl file (in [root]/inc/page\_8.php). The PHP code in page\_8.php and nccp\_simulator.php run the Perl codes through the command line, to generate a CSV file of output. The PHP then reads that output and displays it to users in Step 8 of the tool (also via [root]/inc/page\_8.php). It then erases both the Perl file of inputs and the csv file of outputs, although the $\_SESSION variables are retained so that users can navigate to previous steps and change their entries, to generate potentially new results. The below Architecture Diagram visually represents the process by which output is generated in this program.

WHAT YOU MUST DO IN ORDER TO SET UP THE CODES ON YOUR SERVER

Please note that these instructions are written for a website absent any content management systems such as WordPress or Drupal. The MTRC is written in PHP and Perl, with Perl running though a command line (being executed once per use). Basic steps include:

1. Download the files from the repo that MSG will be sending you, from the master branch.
2. Import the .sql file that includes the "FRS" databases at [root]/sql onto a website's (or local) server. The latest version of the SQL import file will be located in that folder; it is currently calcproj\_2021-08-05.sql.
3. Open [root]/lib/db.php and enter in your database's credentials on line 10. Depending on which version of the file you are using, you may also need to enter these credentials in [root]/lib/nccp\_simulator.php, line 13.
4. Open [root]/lib/nccp\_simulator.php and change the paths on line 102 to where you are staging the tool, such that it points to [root]/lib/frs.pl" file in the first part of that line. This part is needed for output to be generated, otherwise you will encounter errors on page 8. Please change the path after --dir=" to your root directory as well.
5. Try to access [root]/me.php on your website. If the paths and credentials are working correctly, and the database has been imported, you will hopefully be able to open the Maine MTRC.
6. Contact the MSG team for any difficulty in steps 1-5 above. As mentioned above, and as project staff has conveyed to partners in earlier discussion, the MSG team has developed this absent any CMS, to allow for greater flexibility on the part of partner sites. We feel the PHP is flexible enough to accommodate different systems, but can direct you to resources if it is difficult to adapt.

**Benefit Cliff Calculator Architecture Diagram**

\*SERVER

PhP variables (including variables derived from MySQL data) are pushed to Perl

PHP pulls federal/state/local policy information from MySQL database. MySQL database only holds federal/state/local policy information.

MySQL

PhP

CSV file is temporary and deleted after processing

PhP interprets the csv and present result back to the client

Perl processes the user’s inputs and generates a temporary csv file

Javascript – processes field validation and if/then statements

CSV File

JavaScript

Client Computer

Perl

WHAT FILES ARE IMPORTANT AND WHY/BASIC TREE

|  |  |
| --- | --- |
| [root] | *On MSG's BitBucket repository, this directory is family-resource-sim-mtrc/tools/frs\_stage/* |
| me.php | The PHP "master" PHP file that begins the assembly of PHP variables stored as the program runs, and the file that dictates the navigation between the steps of the tool. |
| intro\_me.php | The landing page, that provides introductory text and links to pa.php |
| metest.php | A "testing" page similar to pa.php that can be used to aid debugging. See below for further note about debugging |
| index.php | A "master" PHP file that does not include Maine specific text or options, used primarily for demonstration purposes |
| indextest.php | Similar to metest.php, this can be used to aid debugging when using the tool without identifying the state. |
| helpwithbenefitcliffs.php | The "tips" page that users can see via a link in step 8. This includes some initial suggestions – developed by the project team – on potential ways to reduce the impact or onset of benefit cliffs. It is intended to be a starter document that partners can add on to if they wish, or remove if they prefer. |
| frs.css | The css file that codes how the MTRC's HTML elements are to be displayed on the screen |
| /sql | See below. |
| /inc | See below. |
| /lib | See below. |
| /temp | See below. |
| /images | See below. |
| /scripts | See below. |
|  |  |
| [root]/sql |  |
| *The latest .sql file containing relevant data for determining expense estimations and some policy data (e.g. fair market rents) is included in this folder.* | |
|  |  |
| [root] /inc |  |
| *The following are the .php files that the MTRC actively uses:* | |
| page\_[#1-7].php | The PHP code dictating the text and conditional logic of steps 1-7 of the user interface, and from which the $\_SESSION variables are defined. The me.php, metest.php, index.php, and indextest.php codes generate these steps from the code. |
| page\_8.php | The PHP code that sends the $\_SESSION variables to Perl, later defines final output variables generated from the resulting csv file, and displays results to users. |
| form\_5\_unsubsidized.php | Of the form\_\*.php files, this is the only one that is active with the rest of the MTRC codes. It dictates the text, variables, and conditional logic for child care variables when users select that they do not receive publicly funded child care assistance. |
| choice\_[#1-7].php | The PHP code dictating the text that appears at the bottom of step 8, describing some (but hardly all) of the inputs that users have provided. The numbers of the files correspond with the steps where these input data have been entered. |
| *The remaining files in /inc were used by a predecessor tool to the MTRC that provided the base set of coding for the MTRC. These remaining files have not been deleted yet partially to inform any debugging that may occur.* | |
|  |  |
| [root]/lib |  |
| *The following are the files in /lib that the MTRC actively uses:* | |
| frs.pl | The Perl code that is run from the command line, via the run\_frs function in nccp\_simulator.php that page\_8.php executes. This Perl file uses the unique session id to collect all the Perl inputs defined by [id]\_inputs.pl, identifies the year and state where other Perl files are run, runs the frs.pm package, identifies the sequence of Perl functions to be executed through running defaults.pl, runs the Perl files, generates the net income, net expenses, and net resources variables, and generates the [id]\_private.csv file, before closing. The page\_8.php code then uses that csv file to generate output |
| nccp\_simulator.php | This PHP file contains most of the functions used by the PHP codes in the root directory and in the /inc directory, including functions that use the MySQL database to generate additional $\_SESSION variables. It also includes the necessary path and database credential information, as described above. |
| frs.pm | This Perl module defines or refines a fairly large set of input variables, that the other Perl files need to run. One of the more important aspects of this Perl file is that it sets to 0 a number of variables required in the various Perl codes but possibly not defined in the user interfaces (possibly because user selections implied answers to later questions). When these variables are not defined, it can lead to errors. |
| db.php | The PHP file storing the credential information and functions for the PHP to extract data from the MySQL database |
| /2021 | See below |
|  |  |
| [root]/lib/2021 |  |
| *\*.pl* | The Perl files that the frs.pl code runs to produce output for 2021 MTRCs, regardless of state, are located within this folder. These reflect calculations or estimations of some basic expenses that do not include state-specific coding structures or custom policy rules, as well as policy formulas that also do not include state-specific logic. (State-specific logic and policy variables are contained in the state-specific subdirectories.) |
| */ME* | Seebelow |
| */PA* | Not relevant for ME MTRC |
| */DC* | Not relevant for ME MTRC |
| */NH* | Not relevant for ME MTRC |
|  | |
| [root]/lib/2021/ME |  |
| defaults.pl | The defaults.pl file contains, most importantly, an array that the frs.pl program uses to set the sequence of Perl programs/functions (in Perl, called subroutines) run to produce the output collected in the output private csv file. Comments in the defaults.pl file detail what policies or expenses are gathered within each of the codes, and a justification of the sequence for why one function/subroutine is run before the other. |
| \*.pl | The Maine-specific Perl files that the frs.pl code runs to produce output for the 2021 Maine MTRC, which include execution of policy logic and definitions of policy variables specific to Maine. The 2021 directories for each of the four states of project partners are organized in a similar fashion, such that the frs.pl runs the same federal/generic Perl codes along with state-specific codes. Some of the Perl files in this directory (such as ssp\_mtrc.pl) contain only the definitions of policy variables, while others (such as tanf\_mtrc.pl, hlth\_mtrc.pl, and ccdf\_mtrc.pl) contain policy logic that may be distinct from other states. |
|  |  |
| /temp |  |
| \*[id]\_private.csv | The .csv file that frs.pl constructs based off of the outputs generated in frs.pl, frs.pm, and the Perl files in /2021 and /2021/ME. This is the csv file that the PHP code in page\_8.php uses to generate variables that are displayed in the table and accompanying text. Each private csv file uses a unique alphanumeric identifier [id]. This file, along with all the other files in the temp directory associated with the id generated by each session, is deleted after the code in page\_8.php displays the output to the user. |
| \*[id]\_public.csv | A version of the [id]\_private.csv file that includes much of the same data as the private csv file, but with variable labels instead of variable names. This is not used in the active MTRC code. |
| \*[id].csv | A version of the [id]\_private.csv file that includes much of the same data as the private csv file, but with variable labels instead of variable names. The data generated on this csv file is not used in the active MTRC code. This file was formerly used to generate graphs on the results page, but is no longer used. |
| \*[id]\_inputs.pl | A Perl subroutine specific to each id generated within the page\_8.php code. This collects all the session variables, once run by the frs.pl code, and redefines them as Perl inputs. Collecting these session variables into a single file rather than simply including them as arguments in the command line was necessary because the amount of space needed to pass on all the session variables to the command line can exceed the character limit for command lines of some operating systems. |
| \*[id]\_inputs.txt | A reproduction of the [id]\_inputs.pl file, but saved as a text file. As described below, this can be helpful for debugging purposes. |
|  |  |
| /images |  |
| \*.gif | The images used in the MTRC, including the images for the navigation arrows and "step" buttons, |
|  |  |
| /scripts |  |
| \*.js | JavaScript libraries used for field validation and the definitions of some HTML commands employed in various places throughout steps 1-8 on the user interface |

A NOTE ON POTENTIAL DEBUGGING:

As indicated above, the metest.php file provides some additional functionality that programmers or testers can use to aid any debugging or updating process. The primary differences between metest.php and me.php are:

* As a tester or developer navigates through the steps of the tool, the bottom of the screen populates with a display of the $\_SESSION variables being stored.
* The top of the results screen (step 8) also includes a downloadable link of the Perl inputs file, as a .txt file, which can be saved by a tester who does not have access to the directory where an MTRC is hosted.
* The output files and the Perl input files are not deleted following the display of results on step 8. This can be useful if fatal bugs emerge or if the output appears incorrect.
  + Please note that because the private csv file is the file that Perl writes on, it cannot be open while attempting to generate results for a particular session. If you have opened the private csv file for inspection and wish to run the Perl execution again, you must close the private csv file.
* The top of the results screen (step 8, controlled by page\_8.php) will include the Perl command sent to the command line. This command includes the path set in nccp\_simulator.php and a few $\_SESSION variables. This command can be copied and pasted into a server (or local) command line to run the Perl portion of the program. If the user interface for a session has already been run, the Perl input csv file will be available in the /temp directory, identified by the session ID, which, as described above, defines the necessary input variables to Perl for the various subroutines to run. As with the user interface, a private csv file will be generated if the Perl runs without error (and similarly, it must be closed in order for the Perl to run without error). When running the programs on the command line, various variables appear on screen that have been helpful in debugging this tool. You can adjust which of these variables appear by opening most of the Perl files and adjusting the array of variables appearing below "#debug" near the ends of these modules.