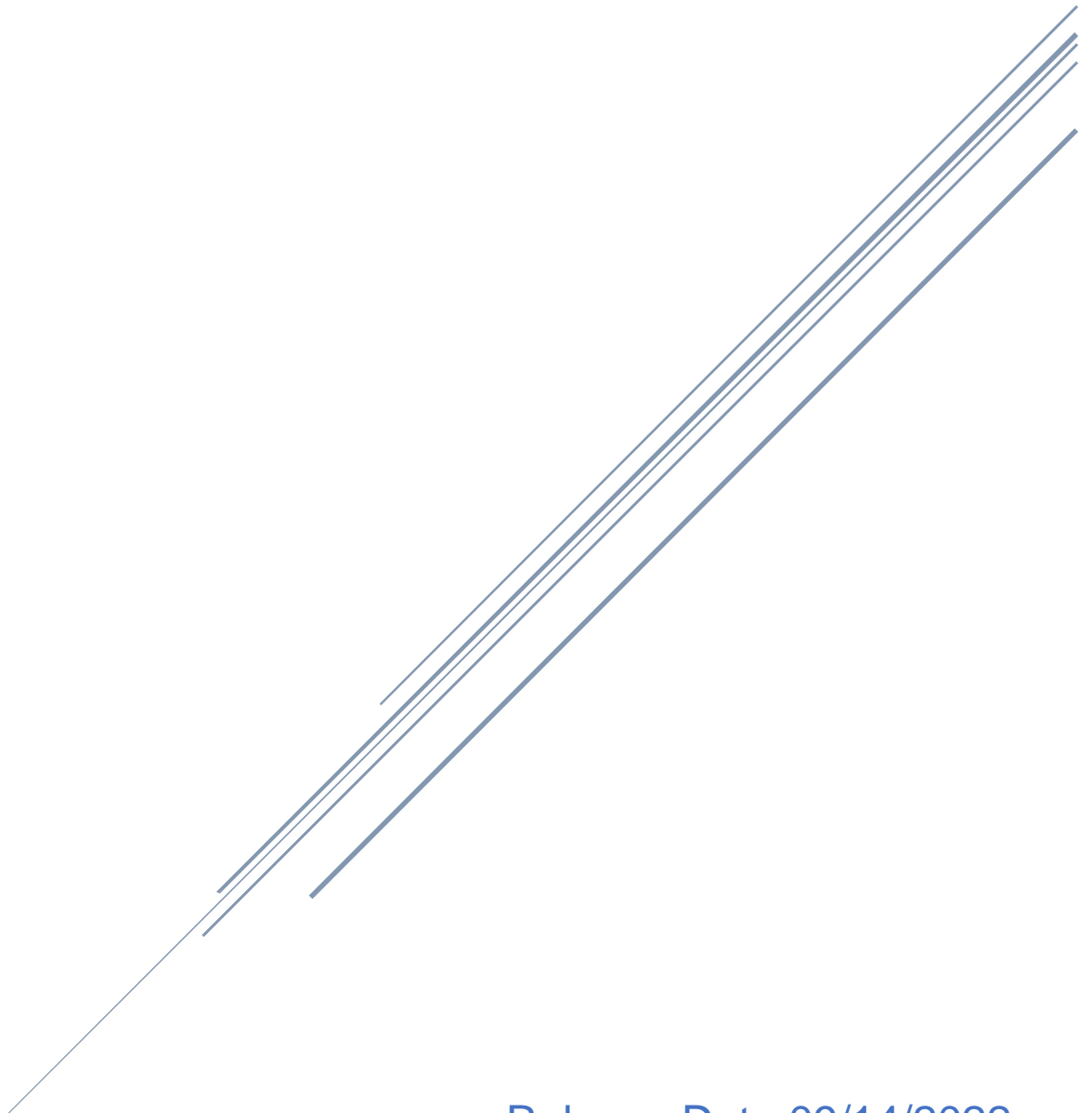


XML GENERATOR

National Environmental Public Health Tracking Network



Release Date 09/14/2022
Version 1.3

VERSION HISTORY

Version #	Author	Revision Date	Reason
1.0	Madhusudan Chaganthi	9/19/2016	Initial Draft
1.1	Madhusudan Chaganthi	10/6/2016	Tidying up the text
1.2	Madhusudan Chaganthi	3/2/2017	Added information regarding the submitter section-related enhancements.
1.3	Kent Nardin	7/21/2022	Updated format, screenshots, and text of multiple sections.

TABLE OF CONTENTS

Section 1 – Overview	3
Section 2 – System Requirements	3
Section 3 – Installation & Running	3
Section 4 – Input File.....	4
Section 5 – How to Use.....	5
Section 6 – Output File.....	7
Section 7 – Log File.....	7
Section 8 – Common Errors	8
Section 9 – Creating Template Files	8
Section 10 – Important Notes.....	9
Section 11 – Questions	10
Figure 1: XML Generator.....	4
Figure 2: Ready to Begin Processing	6
Figure 3: Successful Processing	7
Figure 4: Example TEST Template	9

Section 1 – Overview

The recipients of the Centers for Disease Control and Prevention (CDC) Tracking Program submit data to the Gateway during a data call by sending in Extensible Markup Language (XML) files via Secure Access Management Services (SAMS). Grantee's create the XML data by using various mechanisms like SAS, R, SQL scripts, or other custom means. This task usually needs someone who is reasonably familiar with the data dictionaries as well as the XML technology. This is potentially a daunting task if there is a resource turnover and/or a new data feed is introduced. CDC developed the XML Generator tool as a means for grantees to generate valid XML files from a comma-separated values (CSV) file.

This document outlines the steps required to install and run the XML Generator, how to use the tool, files that are generated, and common errors.

Section 2 – System Requirements


The XML Generator was developed and tested to work on machines using a Windows operating system. For the tool to run, Java 1.8 or higher must be installed on the machine.

Section 3 – Installation & Running

1. Download "XMLGenerator.zip" from the [GitHub repository](#).
2. Move the XMLGenerator.zip file to the desired location on the machine and unzip it. Once unzipped, there will be two new folders – "lib" and "templates," four new "properties" files, and an "XMLGenerator.jar" file.
3. To run the XML Generator, double-click the XMLGenerator.jar file. An alternative running method is to open a Command Prompt in Windows and execute the command, "java -jar XMLGenerator.jar."

CDC Tracking Network - XML Generator

File Help

 CDC Tracking Network - XML Generator

Step 1: Fill in the submitter information

Select State

Submitter Name

Submitter Email

Submitter Title

Step 2: Fill in the data file details

Select Data Feed

Enter Data Year

Enter MCN

Step 3:

Step 4:

Figure 1: XML Generator

Section 4 – Input File

The XML Generator utilizes CSV files as input. The tool will display a warning popup if a different type of file is selected for processing. Remember these notes regarding the CSV input files

- The first line in the CSV file must be the header row. The name of the columns in the header row should match the key values used in the Row section of the Data Feed template file that will be used.
 - Ex: For Birth Defects, the template file is found in the “template” folder and is named “BD.txt.” Within the <Row> section are 15 fields beginning with “RowIdentifier.” All of these field names must be present in the header row of the input file, even “LBFD TWD,” which is labeled as optional.
- Double quotes must surround any data that contains a comma.
 - Ex: Correct: “walking, hiking, running” Incorrect: walking, hiking, running

- Data can be omitted for optional fields, but care must be taken that the correct number of commas are present in the row.
 - Ex. 1: The header row has columns: A,B,C,D,E. Column C is optional.
 - Correct data: a,b,,d,e
 - Incorrect data: a,b,d,e
 - Ex. 2: The same header row as Ex. 1, but Column E is optional.
 - Correct data: a,b,c,d,
 - Incorrect data: a,b,c,d

Section 5 – How to Use

1. Begin by running the XML Generator.
2. Fill all required fields.
3. Select a state from the drop-down menu.
4. The submitter information is determined by the “submitter.properties” file located in the root folder. Users can change the entries in this file prior to using the tool so that the same information is populated into the XML file.
5. Select the data feed from the drop-down menu. The data feed selected determines which template file is used to generate the XML file.
6. Enter the data year as a four-digit number.
7. Enter a valid Metadata Creation Number (MCN) that is applicable for the state, data feed, and year.
8. Click the “Choose CSV input data file” button, navigate to the desired file, and click the “Open” button. Alternatively, the fully qualified file name (along with the directory path) can be typed into the field below the button.

CDC Tracking Network - XML Generator

File Help

Step 1: Fill in the submitter information

Select State: GA

Submitter Name: John Smith

Submitter Email: johnsmith@cdc.gov

Submitter Title: CDC Employee

Step 2: Fill in the data file details

Select Data Feed: Birth Defects

Enter Data Year: 2022

Enter MCN: 12345

Step 3: Choose CSV input data file

C:\temp2\input\Test_2.csv

Step 4: Create XML File Clear Details

Figure 2: Ready to begin processing

9. Click the “Create XML File” button to begin processing. You will know the tool is done processing when a popup message appears indicating that processing was successful. If the tool encounters an issue during processing, a popup message will appear with the error message. This same error message will also be recorded to the log file located in the “logs” folder. Log files will be covered in Section 7.
10. Once the output file is created, click on the “Clear Details” button and supply the values to the fields in Step 3 onwards to generate another file.
11. To close the XML Generator, click on the “X” button in the top right corner or select “File” → “Exit” from the top left corner.

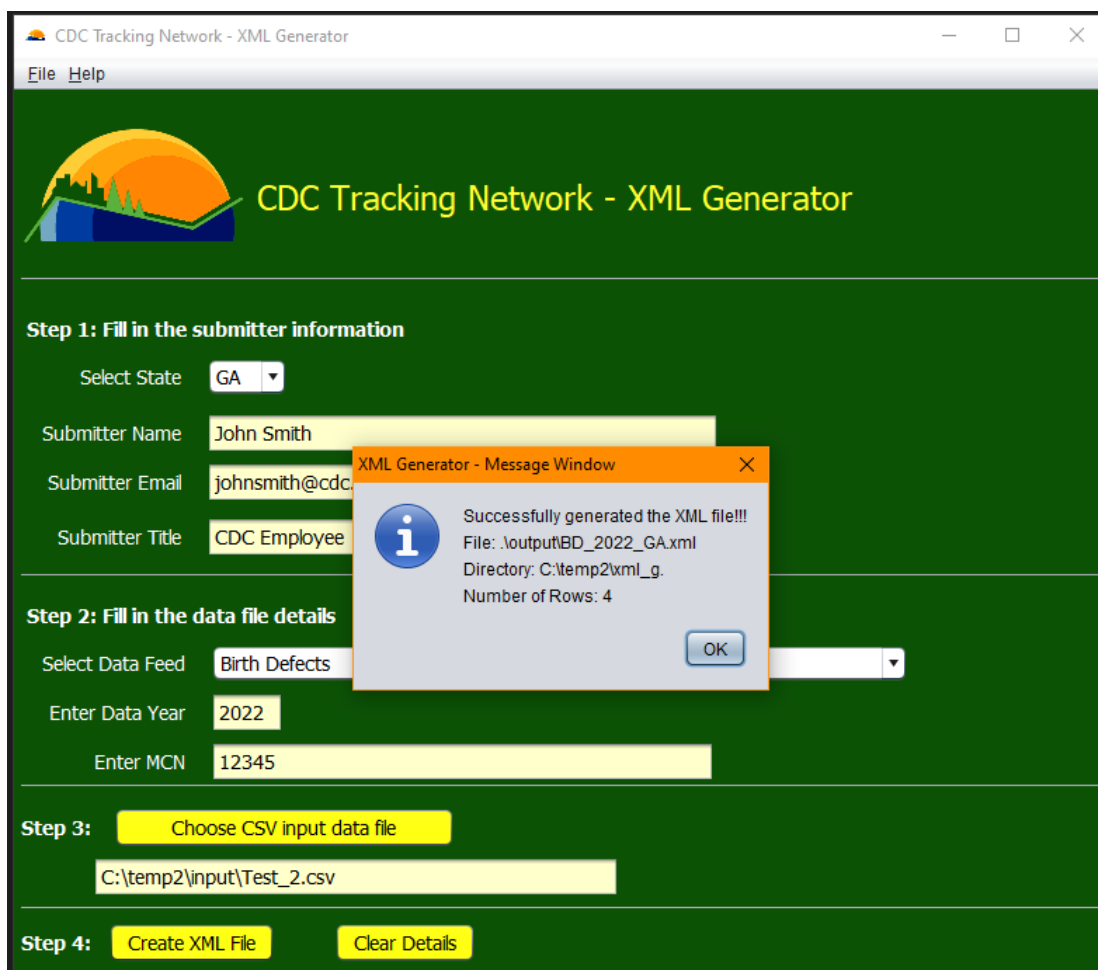


Figure 3: Successful processing

Section 6 – Output File

The output files created by the XML Generator are located in the “output” folder. Each output file follows the naming convention below:

[data feed abbreviation]_[data year]_[state abbreviation].xml

Therefore, a 2020 Birth Defects data feed for Georgia would have the output file name, “BD_2020_GA.xml.”

Section 7 – Log File

When the XML Generator executes, it creates an “XMLGenerator.log” file within the “logs” folder. This log file contains messages generated by the tool as it processes input files as well as messages regarding any errors encountered.

Note: When the XML Generator is run and a file is processed for the first time in a given day, the tool will automatically create a log file for the date on which the XML Generator was the last run. For example, if the tool was run on March 1st and then again on March 4th, upon processing a file on March 4th, a log file for March 1st would be created. This file would be named “logsXMLGenerator-info-error-2022-03-01.log.” The purpose of this logging is to create a historical record of previous executions.

Section 8 – Common Errors

Many of the errors encountered during the processing of an input CSV file will be a result of issues with the data in the input file. For these common errors, the XMLGenerator.log file will include a “**TO FIX**” statement that provides some guidance on how to fix the error that was encountered. Below are some common data issues that will cause errors during processing.

- Empty input file
- Header row
 - Missing header row
 - Missing or misspelled header fields
 - A missing header field may also have missing data, so both should be investigated if this error occurs.
- Data rows
 - Header row present but no data
 - Mismatch between the number of header fields and the number of data fields in a row
 - This was touched upon in Section 4 regarding optional fields. This error occurs when both data and comma(s) are missing from a row. Data that contains commas but not enclosed in double quotes can cause this error.

Section 9 – Creating template files

For the XML Generator, it is possible to create additional templates for other data feeds that may be needed. The easiest path to a new template file would be to use an existing template found in the “templates” folder as a guide. Below are the steps for creating a new template file.

1. Open the “dataFeeds.properties” file in the root folder and delete the “#” character in front of the data feed for which the template will be created. This will enable the data feed to now be selected from the corresponding drop-down menu.

- a. If the data feed does not exist in this file, then a new entry can be made for it. The format is: [data feed abbreviation]=[drop-down menu text]. Example: TEST=Test Template. The data-feed abbreviation cannot contain spaces.
2. Create a new template file whose name is: [data feed abbreviation].txt. Example: TEST.txt.
3. Copy all the contents from an existing template that closely matches the structure of the new data feed into the new template file. Below is an example test template.

```

1  <?xml version="1.0"?>
2  <TestData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ephtn.org/NCDM/PH/SubCountyHOSP">
3    <Header>
4      <MCN>${dataFileDetails.mcn}</MCN>
5      <JurisdictionCode>${submitterInformation.stateAbbreviation}</JurisdictionCode>
6      <ContentGroupIdentifier>${dataFileDetails.dataFeedCode}</ContentGroupIdentifier>
7      <SubmitterInformation>
8        <SubmitterEmailAddress>${submitterInformation.emailAddress}</SubmitterEmailAddress>
9        <SubmitterName>${submitterInformation.name}</SubmitterName>
10       <SubmitterTitle>${submitterInformation.title}</SubmitterTitle>
11      </SubmitterInformation>
12      <StateFIPSCode>${submitterInformation.stateAbbreviation}</StateFIPSCode>
13    </Header>
14    <Dataset>
15      #foreach( $record in $records )
16        <Row>
17          <RowIdentifier>${record.get("RowIdentifier")}</RowIdentifier>
18          <Column1>${record.get("Column1")}</Column1>
19          <Column2>${record.get("Column2")}</Column2>
20        </Row>
21      #end
22    </Dataset>
23  </TestData>
24

```

Figure 4: Example TEST template

4. Modify the content of the new template.
 - a. On line 2 and at the bottom of the file, outlined in the pink boxes in the above screenshot, “TestData” should be updated to the name of the data feed. Note: XML element names cannot contain spaces.
 - b. On line 2, indicated by the light blue arrow, the URL should be updated to the correct data feed URL.
 - c. Starting on line 17, indicated by the green arrow, elements should be added/updated based on the fields in the data feed. Other template files can be used as guides for this purpose as well.

Section 10 – Important Notes

- Executing the XML Generator for the same data feed, year, and state combination will overwrite any previous output files with the same parameters.

Section 11 – Questions

Please direct any questions regarding this document or the XML Generator to nephtrackingsupport@cdc.gov.