XML_Proc

Program to convert Nokia 2G and 3G XML configuration files into an Excel table, with one sheet for each selected Managed Object.

Version: 2.0

Date: 21-04-2020

Author: Ciro Emmanuel Martínez

Contents

Disclaimer

The current version of this program (***XML_Proc***, version 2.0) has been tested reasonably well, but **not** fully tested. More specifically, it could happen that not all the possible Managed Objects in a real network configuration have been taken into account.

Hence, it must be stressed the importance of checking the warnings and statistics shown by the program regarding lines that could not be decoded (and hence dropped), to assess the impact of those exceptions.

In summary, it is highly recommended to make a quick review of the results generated by ***XML_Proc*** and validate them before using those results to make decisions about any changes to be implemented on live mobile networks, and you will be doing this at your own risk.

Open issues / known bugs

Unable to read the XML from server, only local files are allowed (improvement)

When deployed in a web-server configuration, the file explorer that opens to select the source XML file focus on the local directories. This entails a little longer execution time while the source XMLfile (which is usually huge) is uploaded into the server. At the moment, the server folders are not accessible for the program to select the XML file.

Result file can only be stored in the server (improvement)

When deployed in a web-server configuration, the result file is always stored at a specific location in the server filesystem. It is not possible to save the result in the local machine. This shouldn't be a problem if a proper FTP service is configured in the server to retrieve the result file.

Multi-level parameters implemented only up to one level (improvement)

Multi-level parameters (i.e. those parameters which have themselves a list of parameters, or a list of values inside them), are only dealt with up to the first level. Additionally, the naming of those parameters in the output does not exactly follow the naming convention of the OSS.

Description

Usually, Databuild teams generate CSV files each day with the current network configuration, but those files are usually from the whole network or from one full OSS system, and involving all the defined Managed Objects. Hence, that information is usually difficult to manage with usual ***MS Office*** tools (read ***Excel***).

However, in a day-to-day operation a given team focuses in a specific area and even in a specific subset of all the possible Managed Objects, and it can be useful to be able extract that information in tabular form from a customized XML export from the OSS.

XML_Proc allows to do that process from a 2G or 3G ***Nokia OSS*** XML export.

Inputs for the program are the XML file to be processed (one single file), and the types of the Managed Objects to be retrieved.

The program returns just one file: an ***Excel*** worksheet with each of the requested tables (Managed Objects) in a separate sheet, with all the parameters for each element of each table.

How the program works

XML_Proc just reads the XML_File and process it to extract the information requested by the user.

Two main information sources are used by the program: the 2G or 3G XML network configuration file (just one file), and the types of Managed Object the user is interested in retrieving.

The program then saves into an ***Excel*** file, in tabular form, the configuration of all existing elements in the requested Managed Objects present in the XML file.

All the parameters of a requested Managed Object are retrieved and stored in the ***Excel*** file. **Note:** there is a ***Jupyter Notebook*** version of the program which allows the exact definition of the parameters to be extracted for each of the requested Managed Objects, but it is more clumsy to use, and anyway it needs a working ***Jupyter*** environment to be used.

User’s guide

The program is launched by opening the file ***XML_Proc_v2.0.html*** file in a system running an Apache http server (for example, running it under the ***EasyPHP DevServer*** environment, or using a web-server based configuration). This html file gathers the inputs from the user and calls the program that makes all the calculations and generates the output files (the program itself is the file ***XML_Proc_v2.0.py***, under the ***src*** folder)

The program requires ***Python 3.6.4*** to work. The ***numpy*** and ***pandas*** modules are also required.

Depending on the specific computer environment, some adjustment could be required in the ***XML_Proc_v2.0.py*** source file, in order to indicate the program where to look for the Python interpreter in your specific system.

Also, be aware that many other software interactions in a particular system could prevent the program from working properly, especially in a web-server environment.

Inputs

The program just needs one input file: the XML configuration file from the ***Nokia OSS*** to be processed and transformed into tabular data.

The list of required Managed Objects to be processed must also be indicated.

2G / 3G Nokia OSS XML configuration file

The format of the XML is the ***Nokia OSS*** XML format (for either 2G or 3G technologies). Either "User values" or "Internal values" can be selected when exporting from the ***Nokia OSS*** (depending on user's preferences), as the parameter values are just conveyed to the result file, without any internal transformation.

Note: although the program was tested with pretty huge files, and it worked fine, it is highly recommended that their size be as contained as possible, by including only the planning area and the surrounding area. This is needed in order to minimize the probability of running into an out-of-memory error, and to minimize both the processing and file upload times. However, you must ensure that all the information you need is included in the file.

Requested Managed Objects

A list of comma-separated Managed Objects types must be indicated to the program.

If this list is not provided (i.e. the input field is left empty), the program will retrieve all the Managed Objects found in the XML file.

Output files

The program will generate one output file: an ***Excel*** file containing one sheet for each of the retrieved Managed Objects, with all the elements of that Managed Object type, and all the parameters for each element.

This ***Excel*** file will be stored at a specific location in the server, which is shown in the browser for ease of reference.

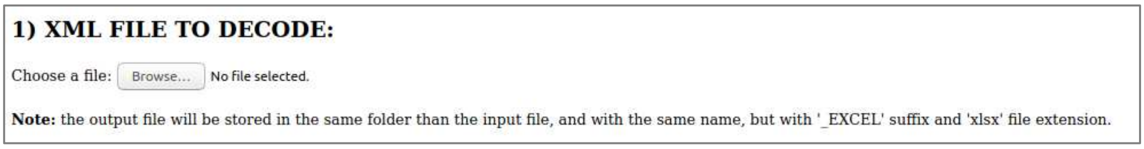
Additionally, the program can show several messages, warnings and results in the browser window.

Note: it is highly recommended to pay attention to the warnings shown in the browser by the program, specially at the final report indicating the line decoding stats.

Configuration parameters

In the user interface shown at the beginning of the program, the user must specify all the necessary information to run the program: where to look for source data, and the tables (Managed Objects) requested.

2G / 3G XML file



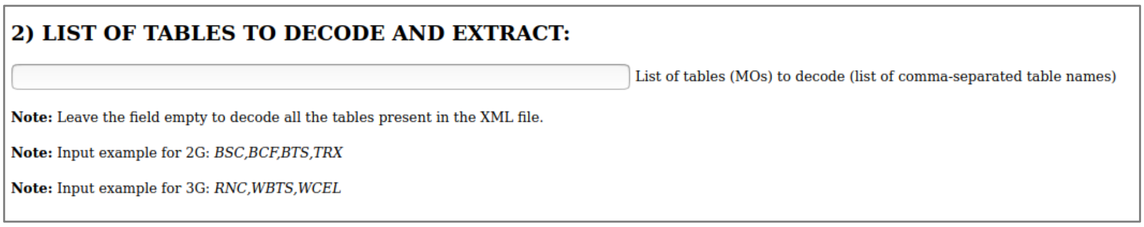
1) XML FILE TO DECODE:

Choose a file: Browse... No file selected.

Note: the output file will be stored in the same folder than the input file, and with the same name, but with '_EXCEL' suffix and 'xlsx' file extension.

In the first section, click on ***Browse...*** to select the XML file you want to process. A file selection dialog box will open to select the file:

Tables (Managed Objects) to retrieve



2) LIST OF TABLES TO DECODE AND EXTRACT:

List of tables (MOs) to decode (list of comma-separated table names)

Note: Leave the field empty to decode all the tables present in the XML file.

Note: Input example for 2G: *BSC,BCF,BTS,TRX*

Note: Input example for 3G: *RNC,WBTS,WCEL*

In the second section, specify the tables (i.e. Managed Objects) you are interested in retrieving from the XML file and saving to the ***Excel*** file.

Output files

The output file will be named the same as the input file, but with an "_EXCEL" suffix and with the ".xlsx" extension.

It will be saved at a specific file folder in the server location or, if that location does not exists, in the ".py" program current location.

Run program button

After all the desired configuration parameters have been set, click on the ***DECODE XML*** button at the bottom of the web form. The program will then read the XML file, decode it, and extract the requested information to the output ***Excel*** file.

You can also press ***Enter*** to run the program.

Any serious problem the program could find will be shown in the browser.