**XML filter TOOL**

Table of Contents

[1 Introduction 4](#_Toc176178055)

[2 Getting Started 5](#_Toc176178056)

[2.1 Set Up the Python Environment 5](#_Toc176178057)

[2.2 Install Required Libraries 5](#_Toc176178058)

[2.3 Navigate to Project Directory 5](#_Toc176178059)

[2.4 Run the Application 5](#_Toc176178060)

[3 How to use the tool 6](#_Toc176178061)

[3.1 Selecting and XML file 6](#_Toc176178062)

[3.2 Filtering Options 7](#_Toc176178063)

[4 Future enhancements 9](#_Toc176178064)

[Figure 1. Browse and Apply Filter buttons 6](#_Toc176178088)

[Figure 2. XML-like presentation of the input 6](#_Toc176178089)

[Figure 3. Grouping of filters 7](#_Toc176178090)

[Figure 4. Output presentation 8](#_Toc176178091)

[Figure 5. Save results 8](#_Toc176178092)

# Introduction

The XML Filtering Tool is designed to assist users in parsing, filtering, and manipulating XML files based on specific attributes and elements. This guide outlines the steps to effectively use the tool and potential enhancements for future development.

When dealing with a big XML file, parsing it and filtering its elements can become quite difficult. This tool is trying to retrieve the blocks you are searching for based on the filters the user provies and to display them in an XML-like format.

# Getting Started

To begin using and extending the XML Filtering Tool, follow these steps to set up your development environment.

## Set Up the Python Environment

Firstly, this step should be taken in account only if you would like to make some changes inside de code. Ensure you have Python installed on your local machine. The tool is compatible with Python 3.x versions. You can download Python from the official Python website.

You can install venv to your host Python by running this command in your terminal:

pip install virtualenv

To use venv in your project, in your terminal, create a new project folder, cd to the project folder in your terminal, and run the following command:

python<version> -m venv <virtual-environment-name>

cd project

python3.8 -m venv env

Now that you have created the virtual environment, you will need to activate it before you can use it in your project. To activate your virtual environment, run the code below:

source env/bin/activate

You can now install the 2 libraries:

pip install tk

pip install elementpath

## Install Required Libraries

The main libraries used in this project are:

Tkinter: This library is included with most Python installations and is used to create the graphical user interface (GUI).

ElementTree: This built-in library provides functions for parsing and manipulating XML files.

To check if you have them installed, you can try importing them in a Python shell:

**import tkinter**

**import xml.etree.ElementTree as ET**

If you encounter an ImportError, ensure you have the appropriate version of Python installed. No additional installations are required outside of the standard library for these libraries.

## Navigate to Project Directory

Once the files are extracted or cloned, navigate to the project directory where the main Python script is located. You can do this using the terminal or command prompt:

*cd path/to/your/project-directory*

## Run the Application

To start the XML Filtering Tool, run the main Python script using the following command:

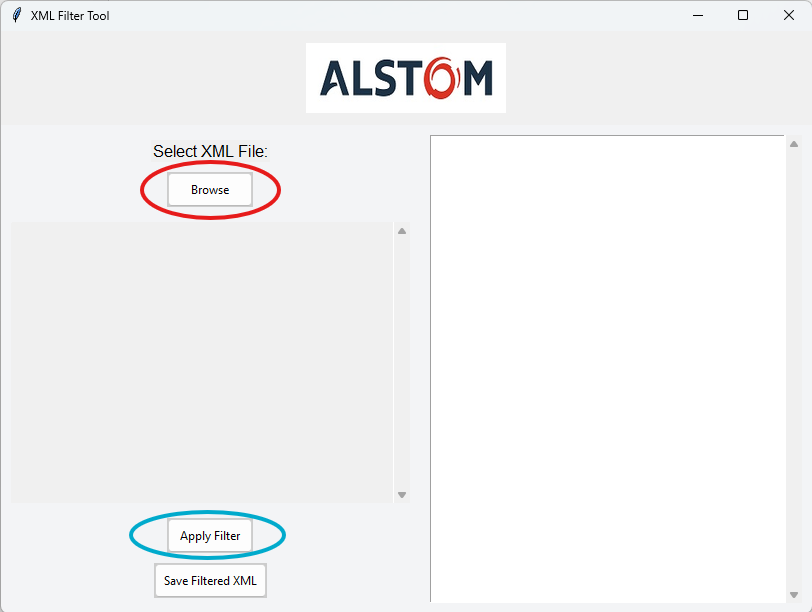
**python main.py**

Replace **main.py** with the actual name of the main Python file if it differs.

# How to use the tool

## Selecting and XML file

Launch the application and click on the "Browse" button to browse and select an XML file you wish to analyze.

Figure 1. Browse and Apply Filter buttons

After uploading the file in the program, the user can just click ‘Apply Filter’ button which will display the initial XML for a better understanding of the data or checking if the loaded file is the correct one. In any case, after selecting the file from the “Browse” button, the user is informed the path of the file.

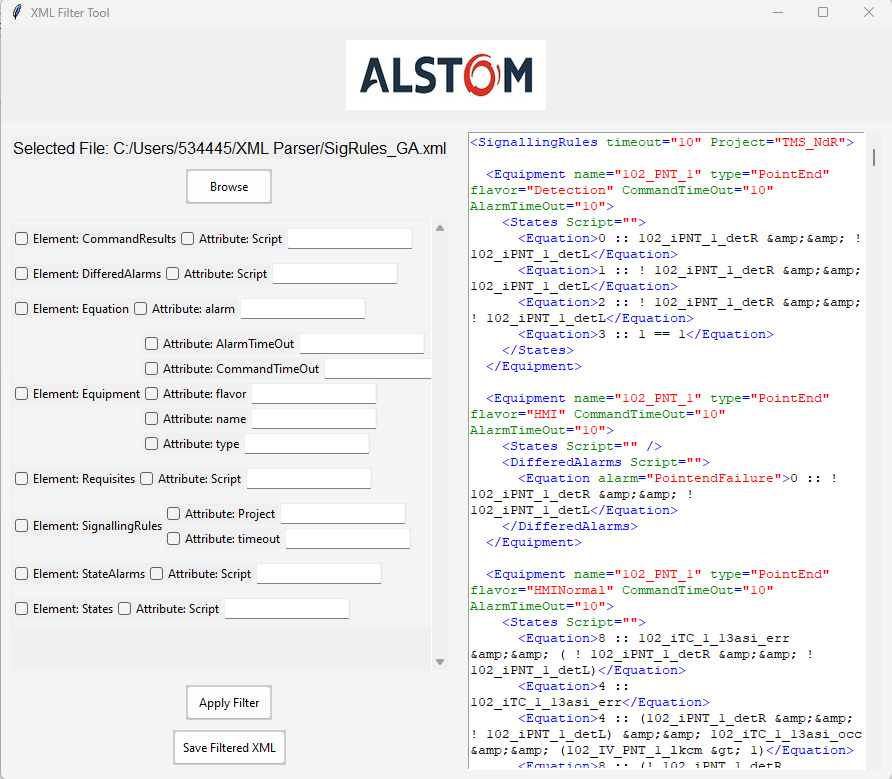


Figure 2. XML-like presentation of the input

## Filtering Options

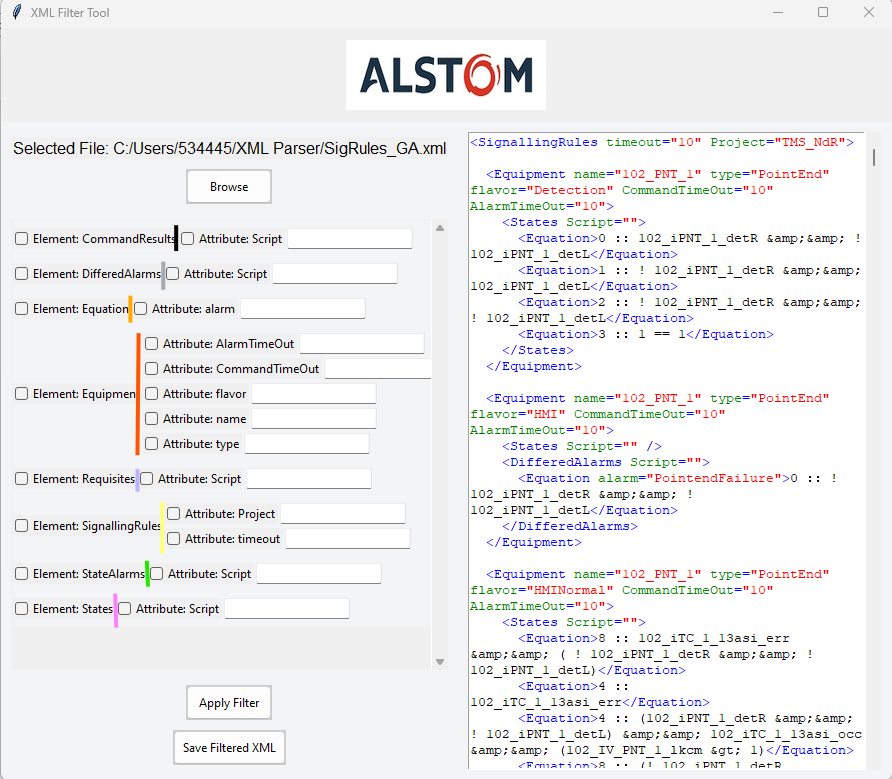


Figure 3. Grouping of filters

The program displays dinamically all the “Elements” and “Attributes” existing in the XML file and all the attributes of an element are displayed in the same line as shown above. (‘CommandResult’ has just ‘Script’, ‘DifferedAlarams’ has just ‘Script’, ‘Equipment’ has ‘AlaramTimeOut’, ‘CommandTimeOut’, ‘flavor’, ‘name’, ‘type’, etc.).

The user needs to tick the boxes he needs for filtering, otherwise they will not be taken into account.

For attributes of selected elements, you can specify inclusion or exclusion patterns:

Inclusion: Write the attribute value you want to match.

Exclusion: Precede the attribute value with a - to exclude matching entries.

For example, if we intend to filter the “Equipment” element to contain in the name the string “102\_PNT\_1” and the flavor to contain “HMI”, but to exclude “HMINormal” we can use the filters as below.  
Additionally, the window from the right with the XML-like format can be directly edited from there for quick modifications.

A screenshot of a computer

Description automatically generated

Figure 4. Output presentation

After filtering and adding the modifications, we can save the output by clicking the “Save Filtered XML” button.

A screenshot of a computer

Description automatically generated

Figure 5. Save results

# Future enhancements

Save user-defined filter settings to enable easy re-use in future sessions.

Export options:

* + Provide options to export filtered data into other formats like CSV or Excel.

User Profiles:

* + Provide options to export filtered data into other formats like CSV or Excel.