

## Overview

This document describes how to define an XML Schema Refactoring for QTAssistant configuration to provide a solution to a question posted [here](#).

## Install and Configure QTAssistant

QTAssistant may be installed on any version of Windows operating system running XP SP3 or above, 32-bit (x86) or 64-bit (x64), with Microsoft .NET Framework 4 and/or .NET Framework 3.5 SP1.

Download the latest version of QTAssistant from <http://paschidev.com/Secure/Downloads/QTSetup40.zip> (registration is required).

Download the zip file containing the material used to illustrate this document from <http://paschidev.com/Secure/downloads/xsd-structure-to-csv-as-path-type-cardinality.zip>; extract the files on your local computer.

Install and run QTAssistant. Once started, close the feature browser to return to QTAssistant's empty environment. For more information on QTAssistant's shell and the XSR module see the online help [here](#) or the embedded help by pressing the F1 key.

## How to Export XPathS described by an XSD

### Objective

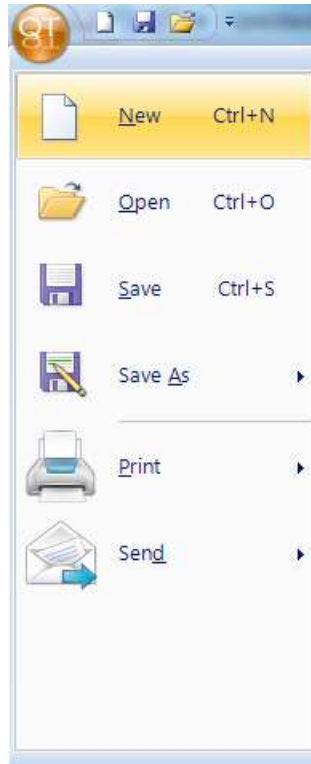
### Sample Files

The zip file contains the following files:

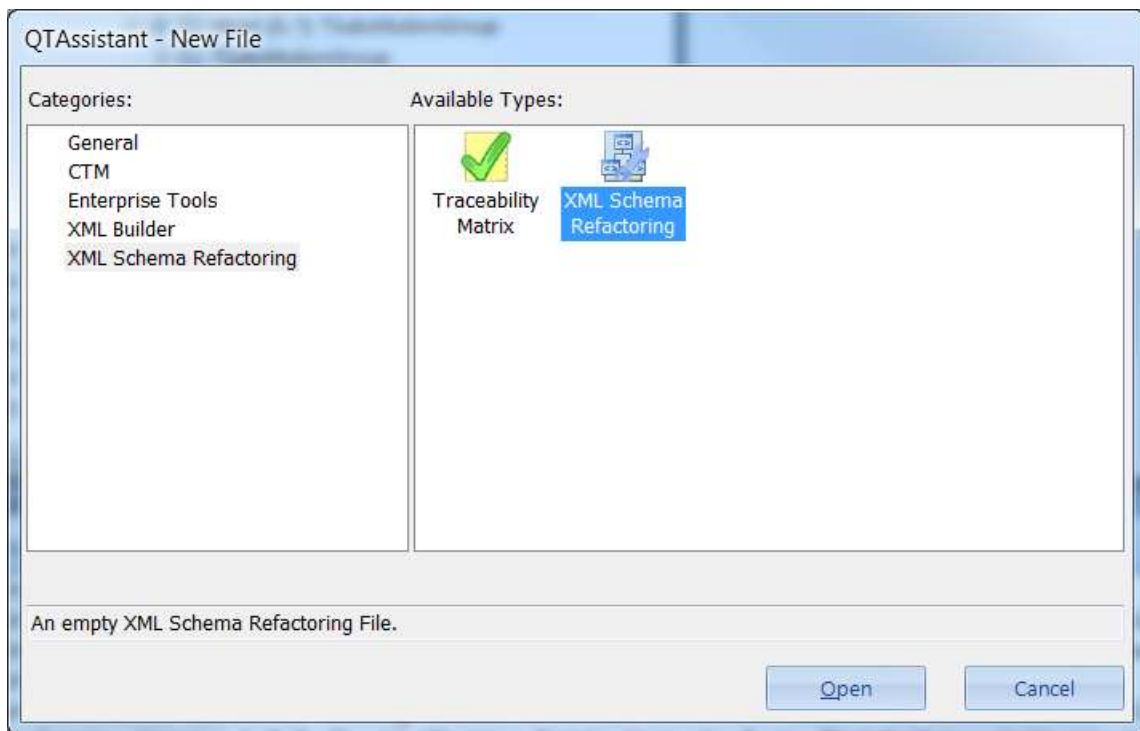
File name	Description
<b>CardinalityTemplate.xtmt</b>	eXtensible Traceability Matrix (XTM): file used to capture traceability between data presented in tabular format and XML Schema trees.
<b>XTM-AppendAll-Parent.xsd</b>	Sample XML Schema file used to illustrate an Append All test case using elements with abstract types and abstract elements (substitution groups); provides extensions to types defined in XTM-AppendAll.xsd
<b>XTM-AppendAll.xsd</b>	Sample XML Schema file used to illustrate an Append All test case using elements with abstract types and abstract elements (substitution groups); provides base definitions and the root element.
<b>XTM-AppendAll.xsr</b>	A pre-configured XML Schema Refactoring (XSR) file.

### Step 1: Create and Save an XSR file.

Click **New** from the Application Menu.



Double-Click **XML Schema Refactoring** file type.

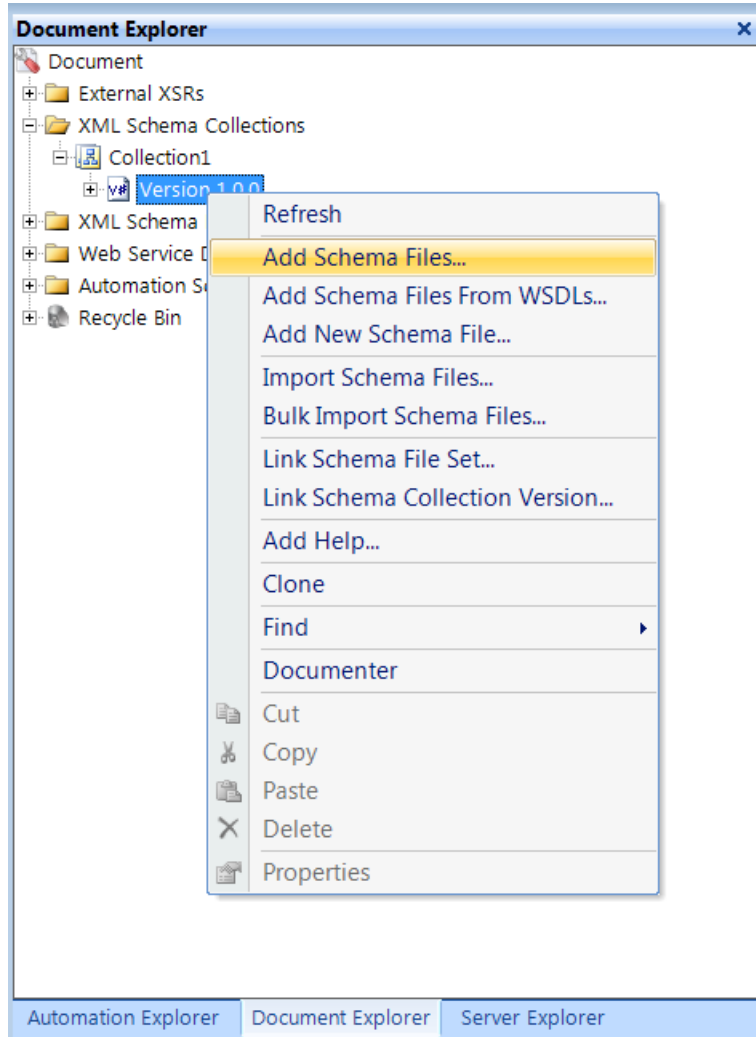


## Step 2: Create an XML Schema File Collection

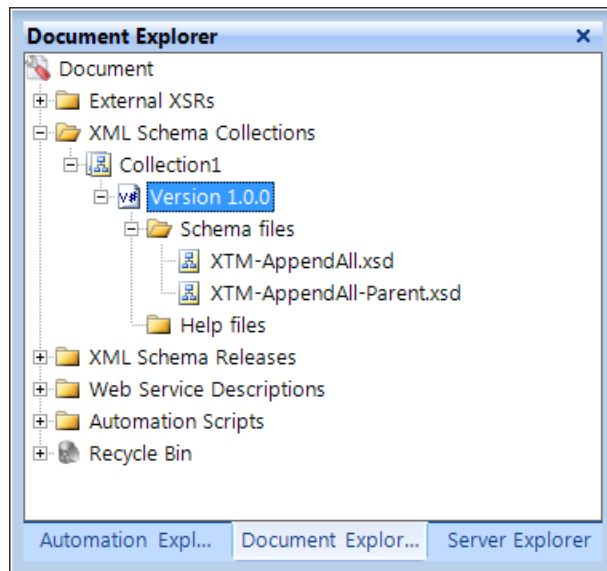
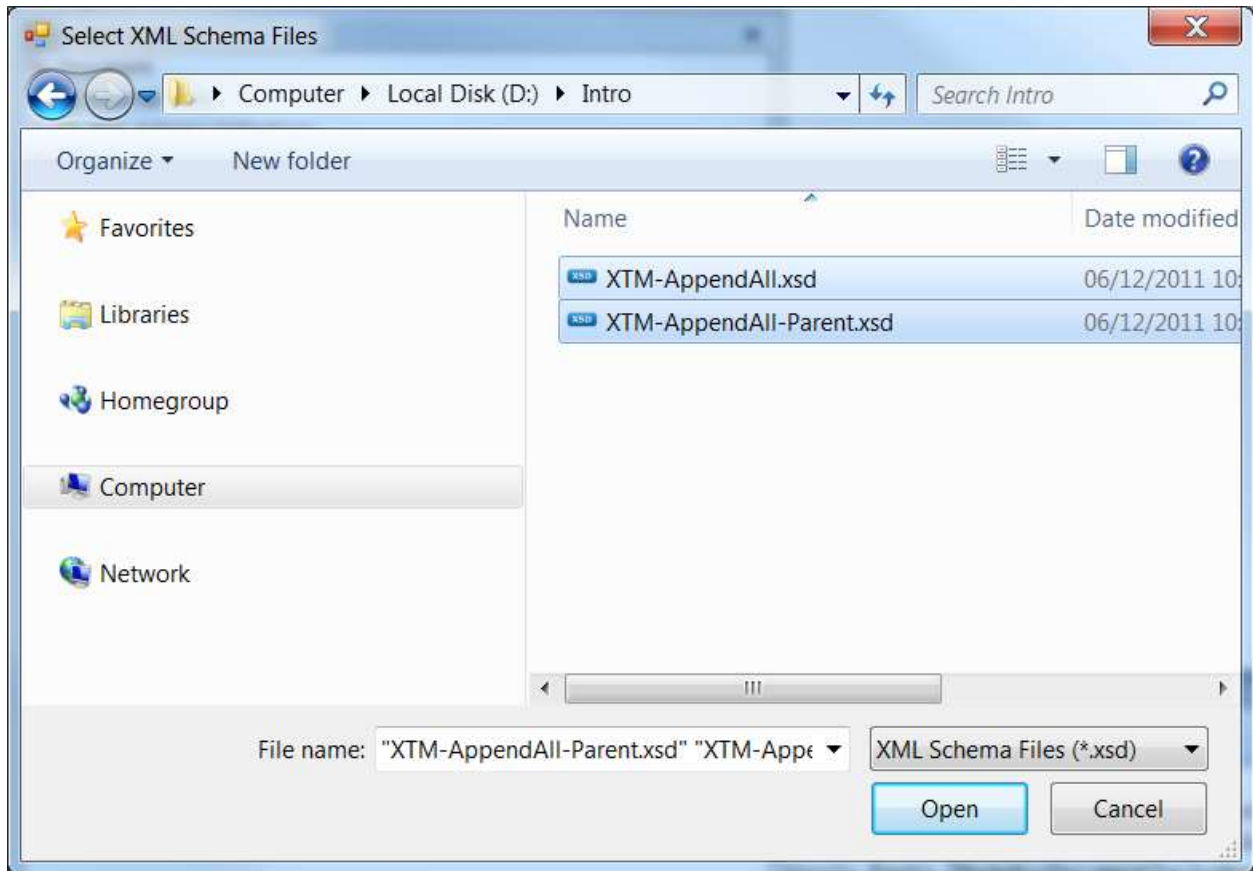
<http://paschidev.com/Secure/Documentation/XsrCreateXmlSchemaCollection.html>

### Step 3: Add XML Schema Files to Version 1.0

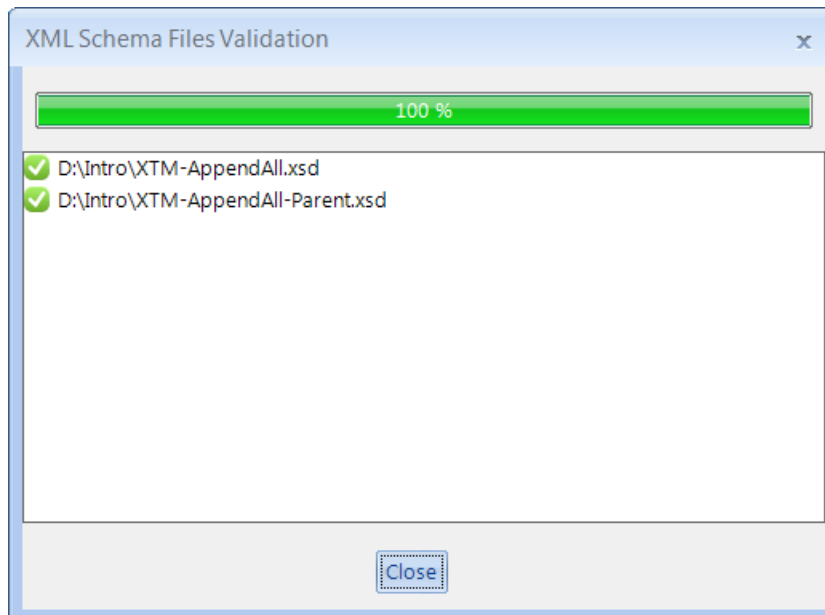
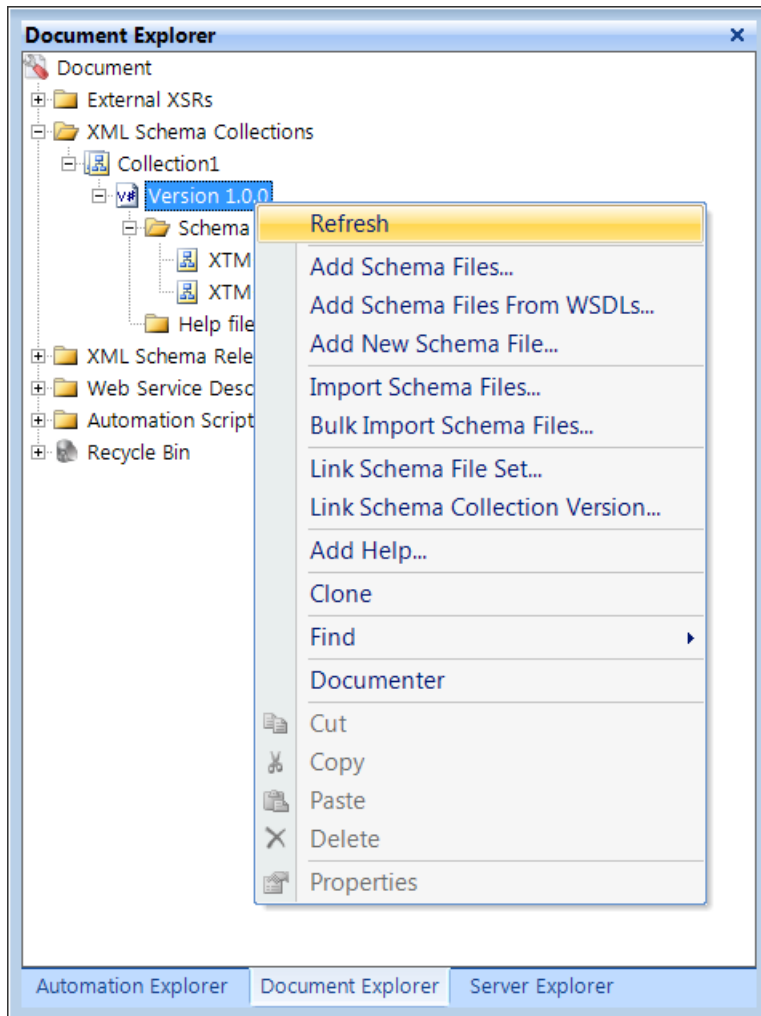
Right click on the Version 1.0 files and click on **Add Schema Files...**



Browse to the folder where your XSD files are stored (for illustration, browse to the folder with the sample files and select all the XSD files) and select the one you want to load.



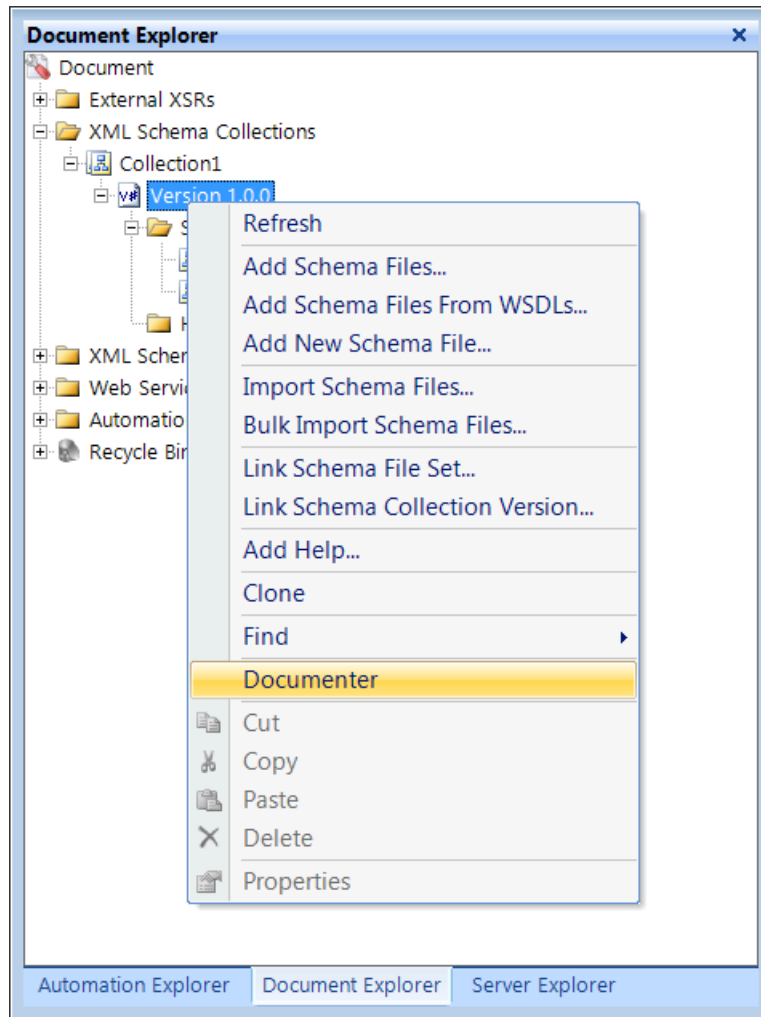
After loading the files, validate them to make sure your sources are OK. Right click on the Version 1.0 files and click on **Refresh**



Files with errors will be marked with a red stop sign. You'll have to double click on the file(s) with the errors, then close the dialog and fix the errors in the provided XSD editor.

#### Step 4: Open the Documenter for Version 1.0

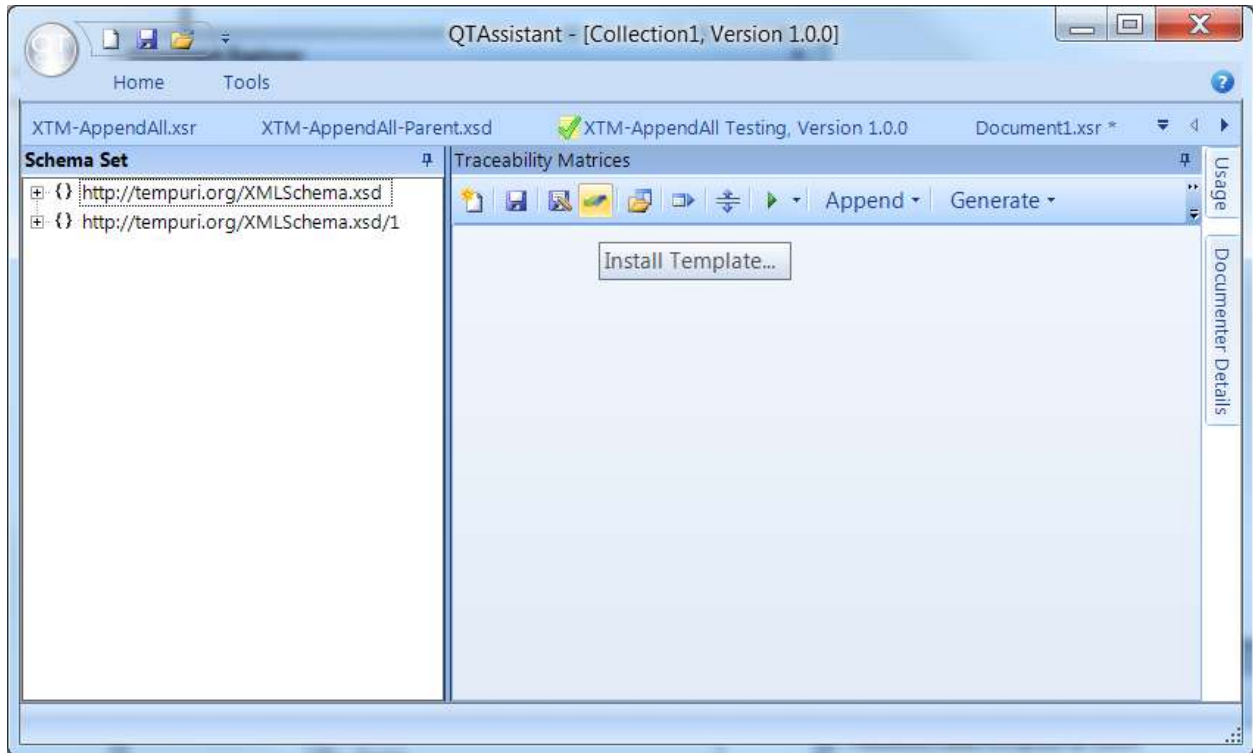
Right click on the Version 1.0 files and click on **Documenter**



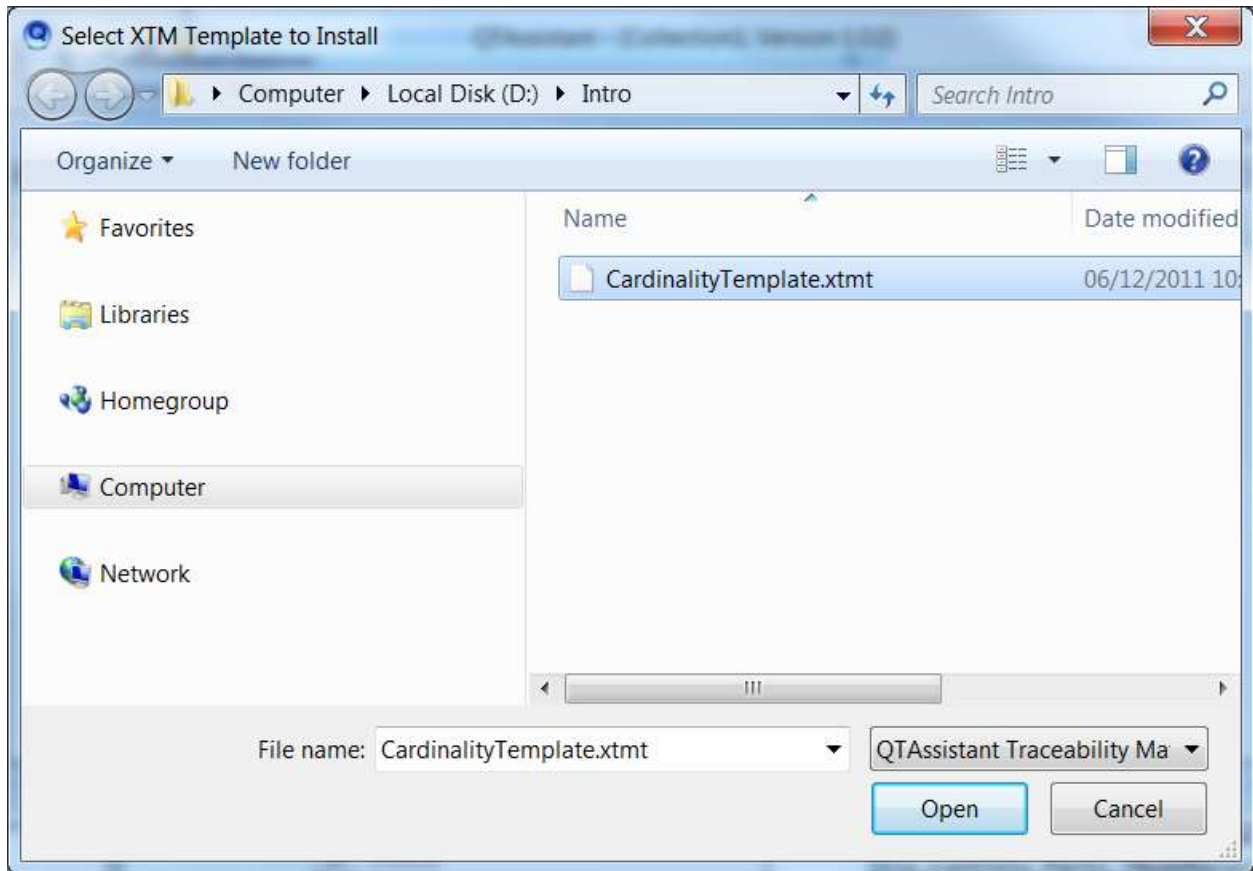
#### Step 5: Install the XTM Template

**Note:** This is one time exercise. I provide one to simplify your setup.

Click on **Install Template...**



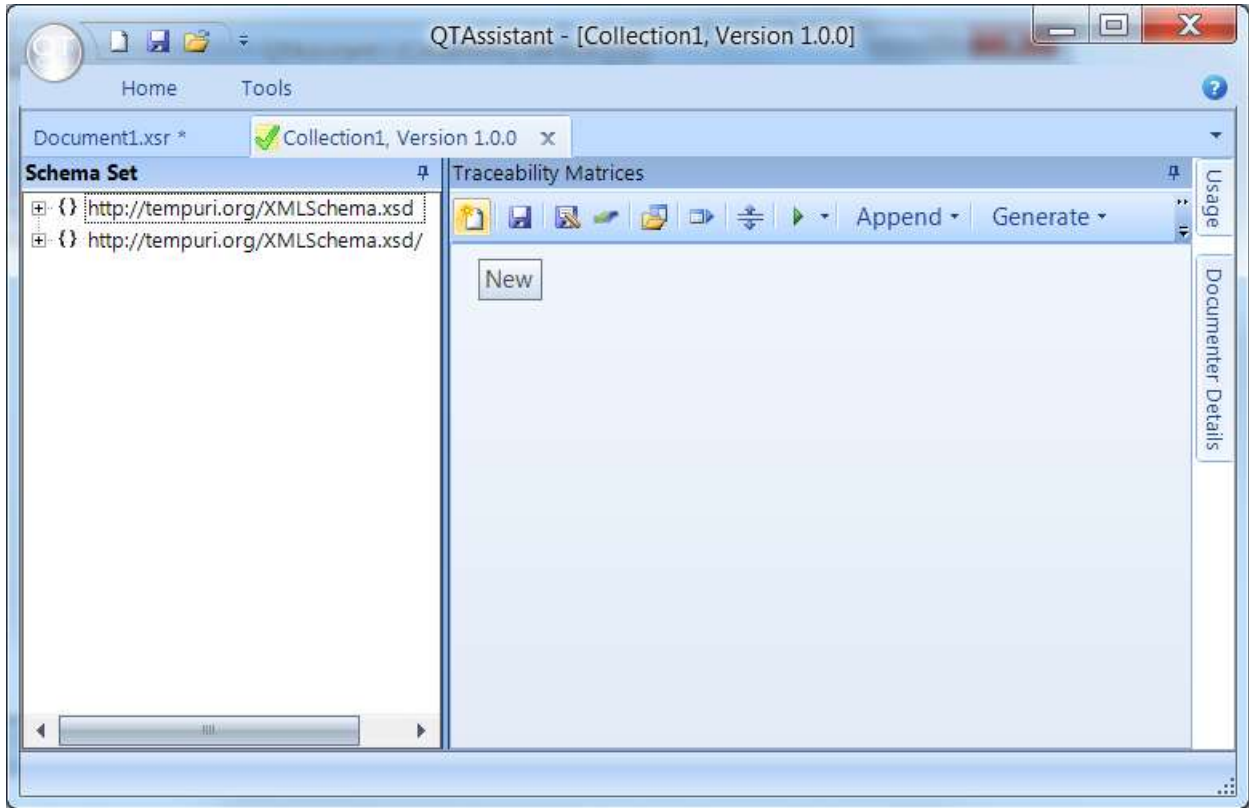
Browse to the folder where you stored your files from the zip and select the provided template.



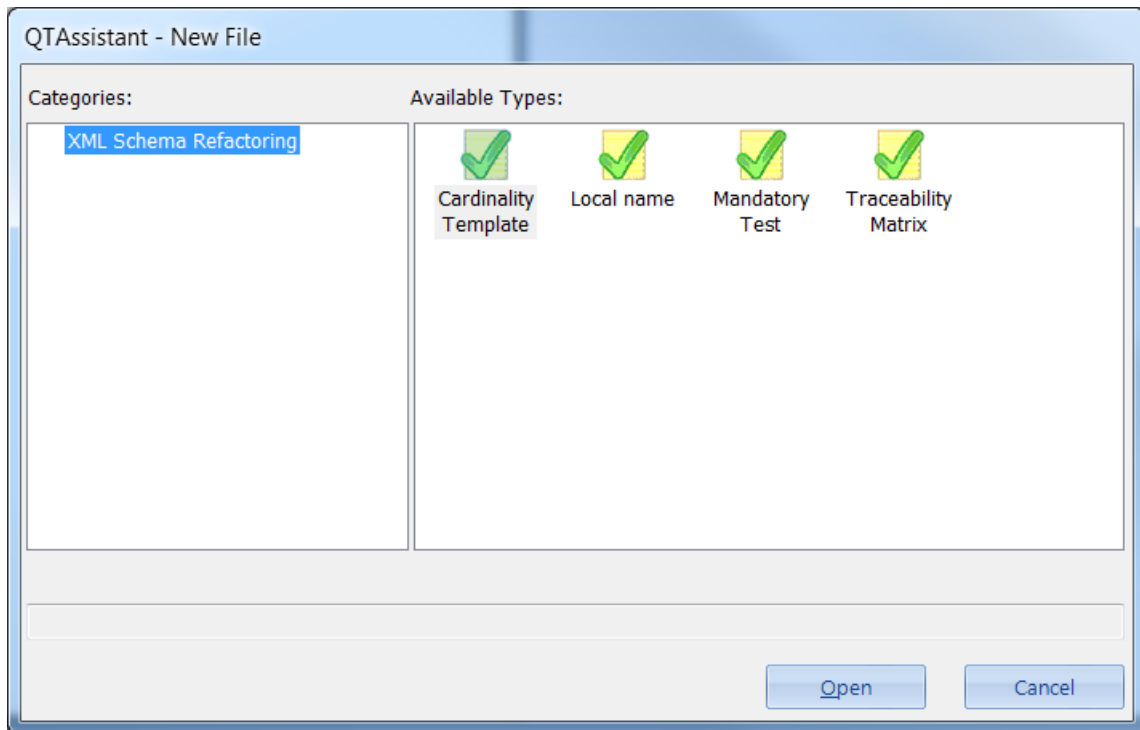
## Step 6: Create a New XTM File Using the template

Click on **New...**

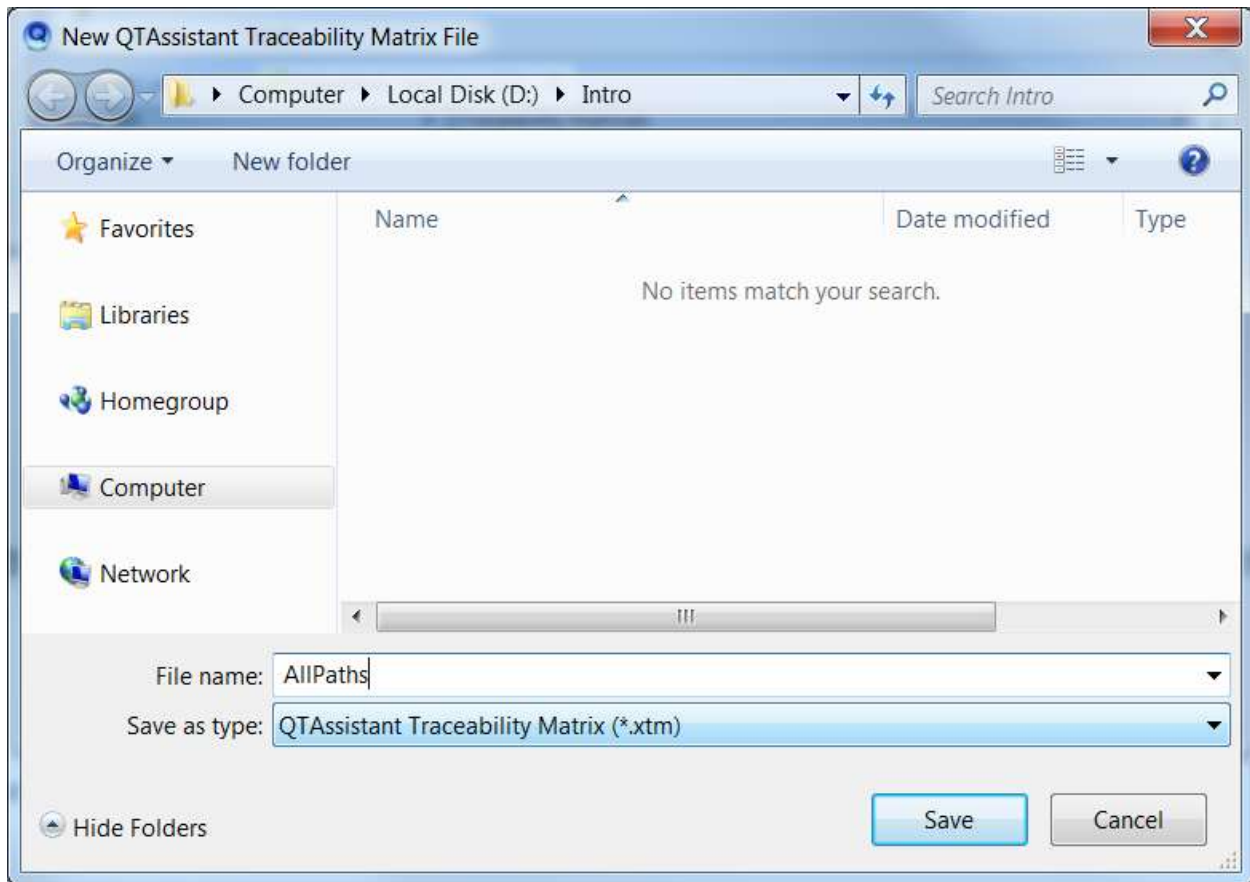




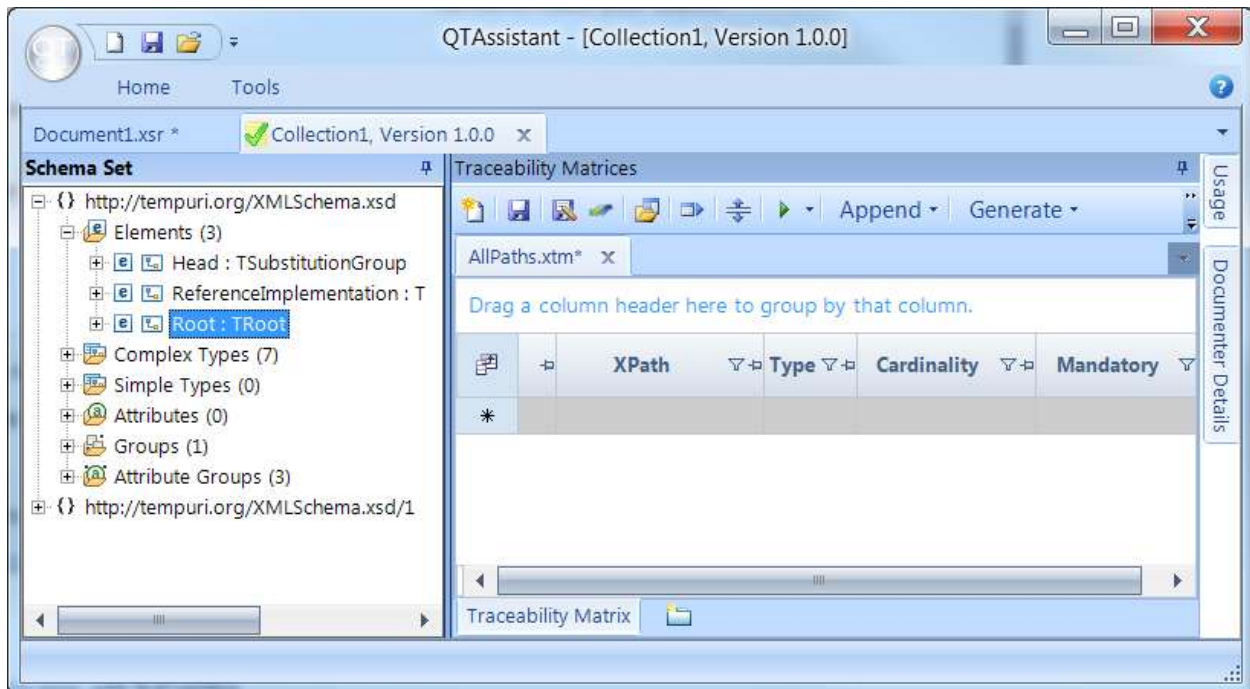
Double click on **Cardinality Template**



Specify a file to save the XTM file (this stores the metadata required to generate the Excel file)

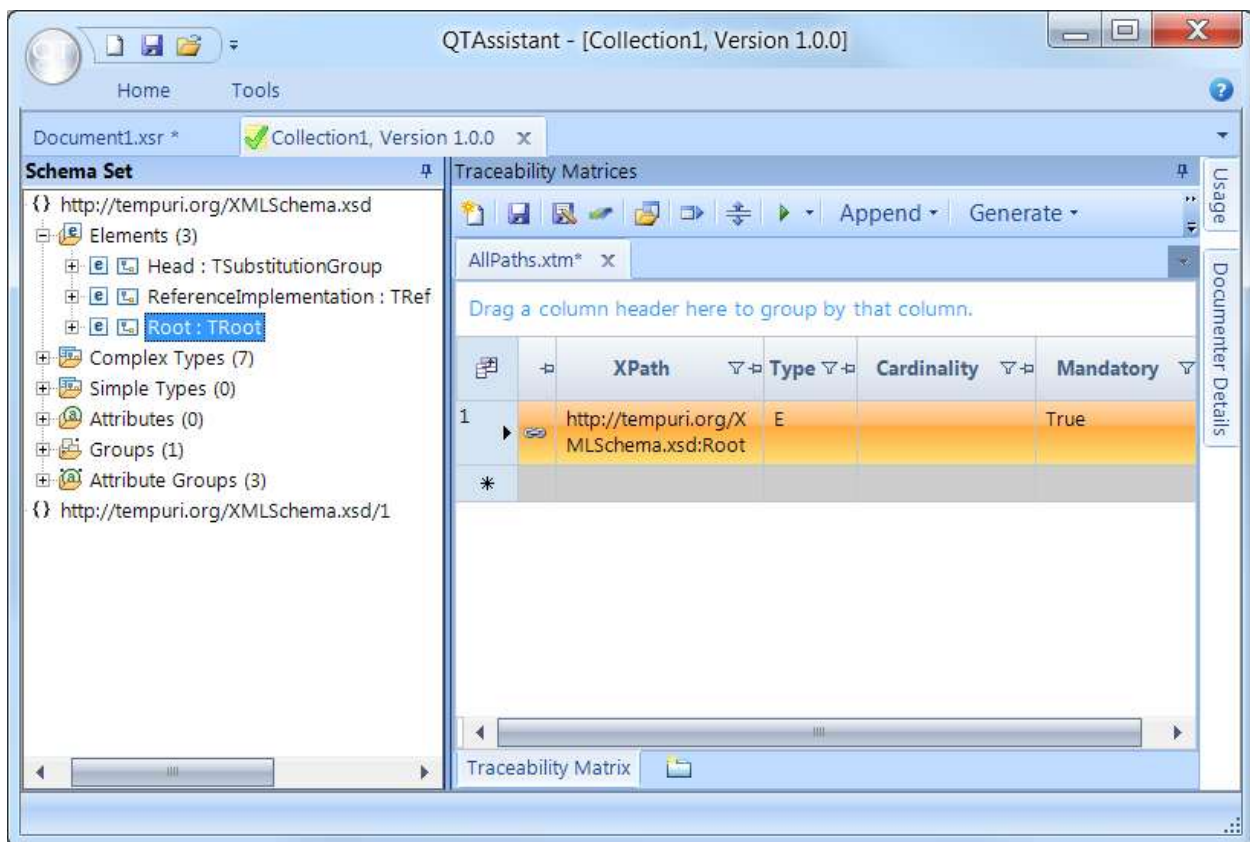


You should get something like this.

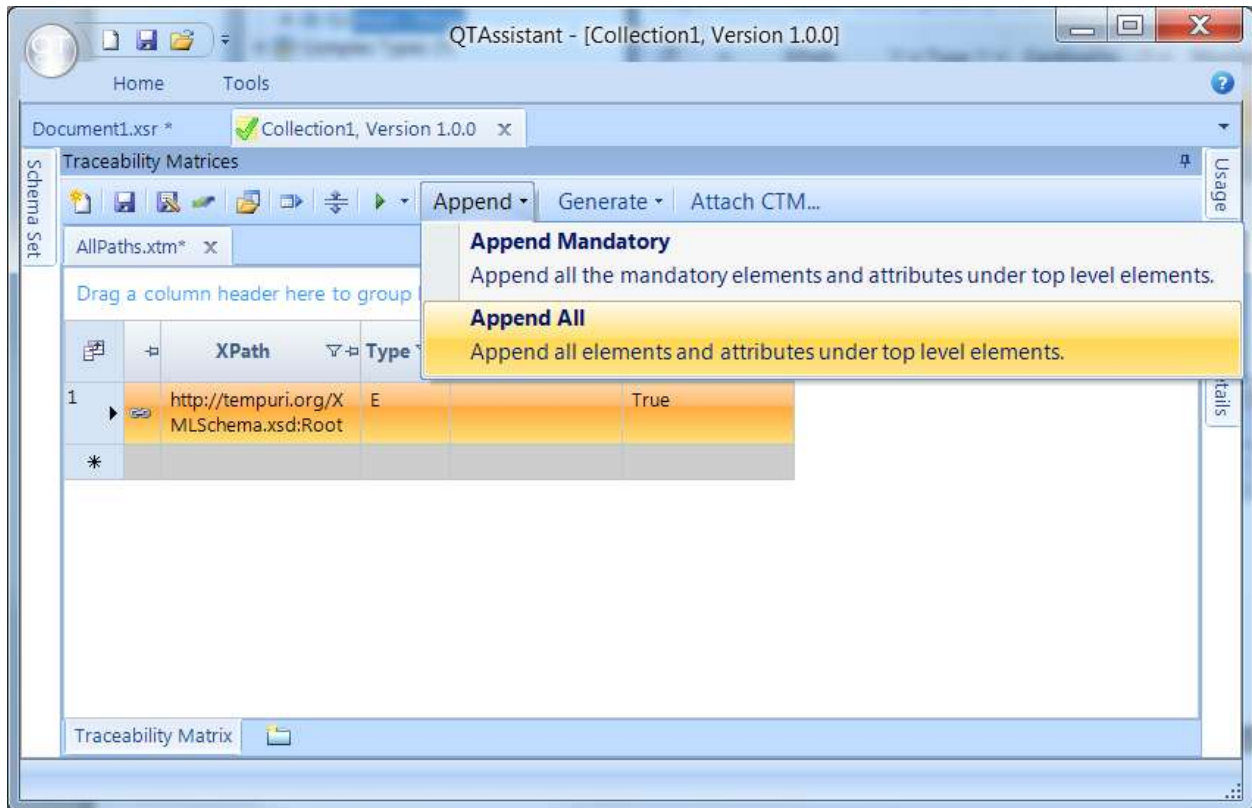


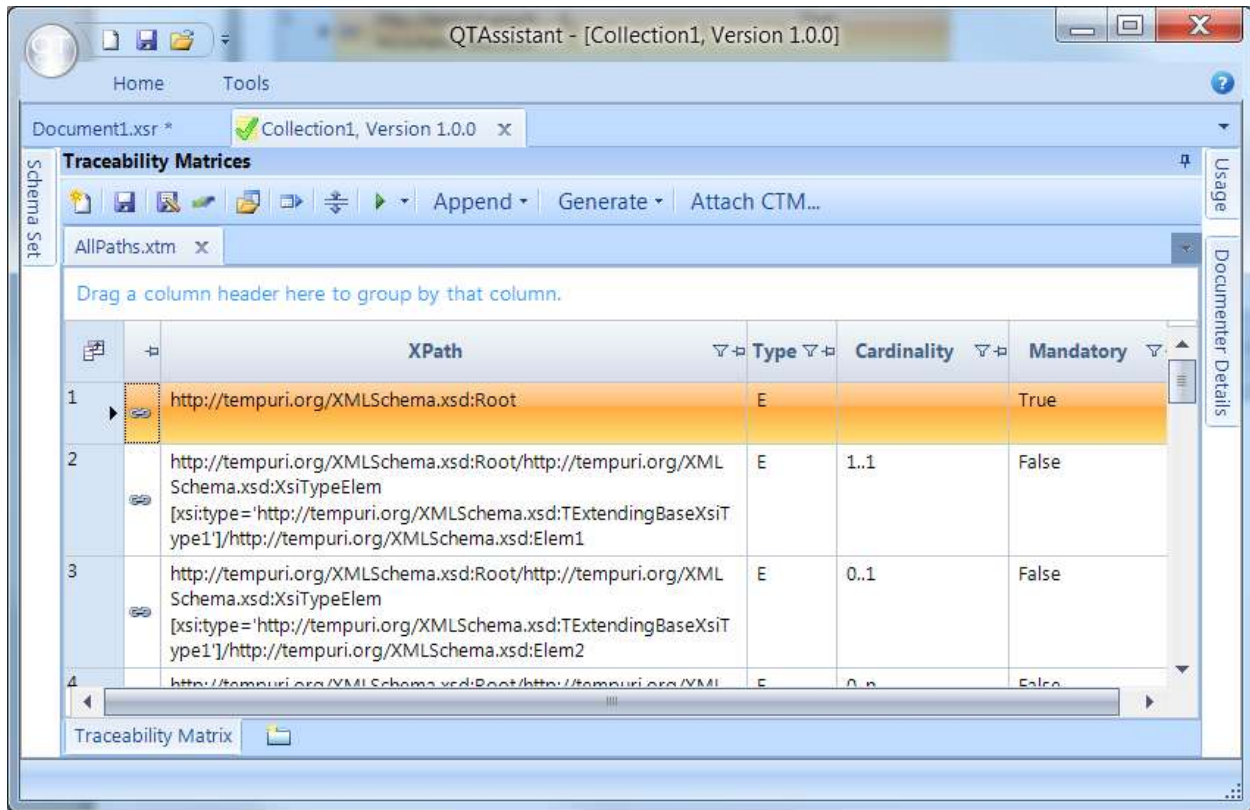
## Step 7: Generate all XPath

Drag and drop your root element(s) from the Schema Set to XTM file.



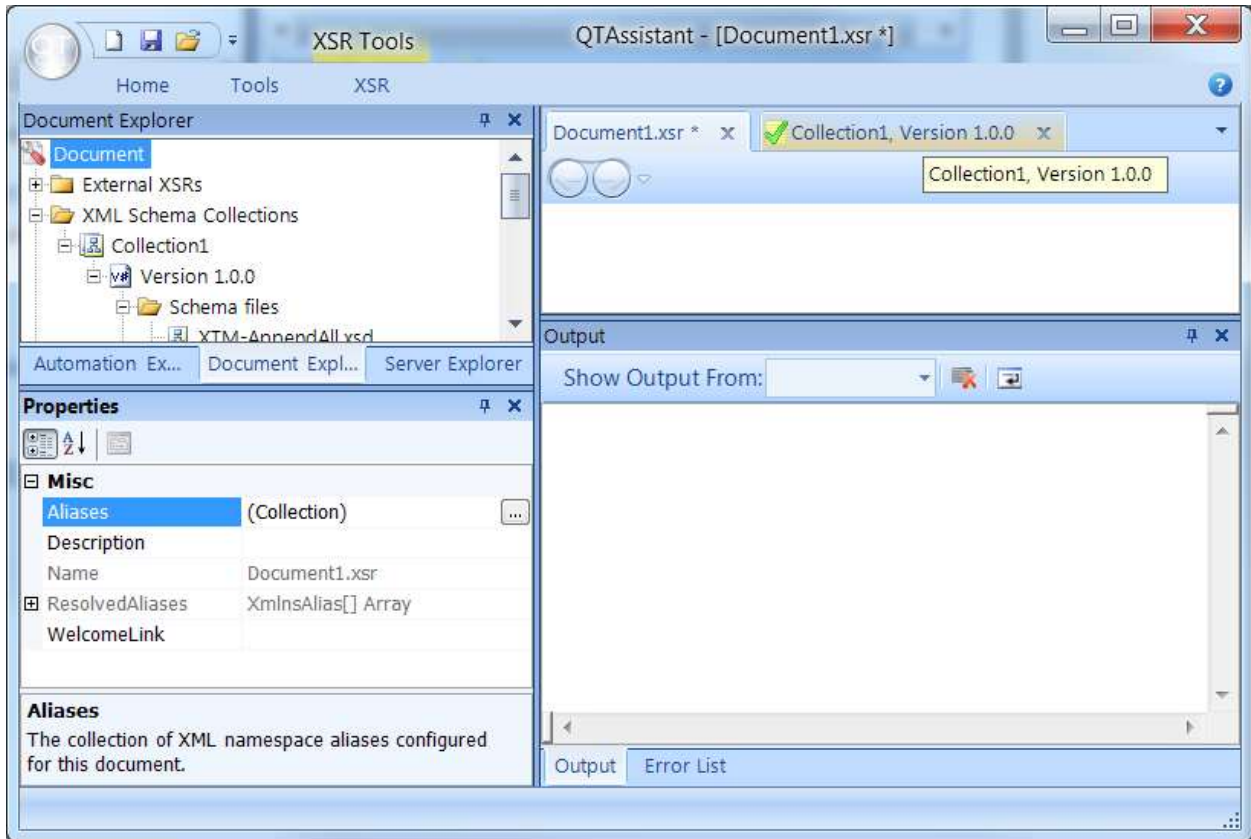
Click on **Append All**



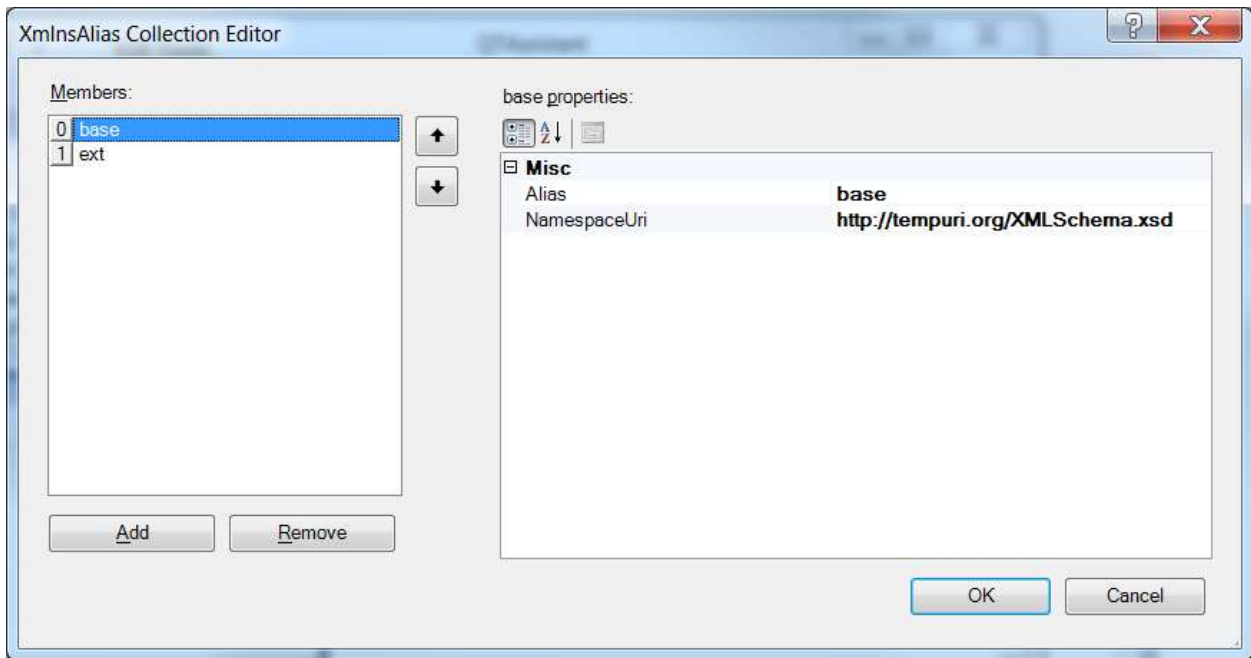


### Step 8: Fix any namespace prefixes that are missing, or assign different values

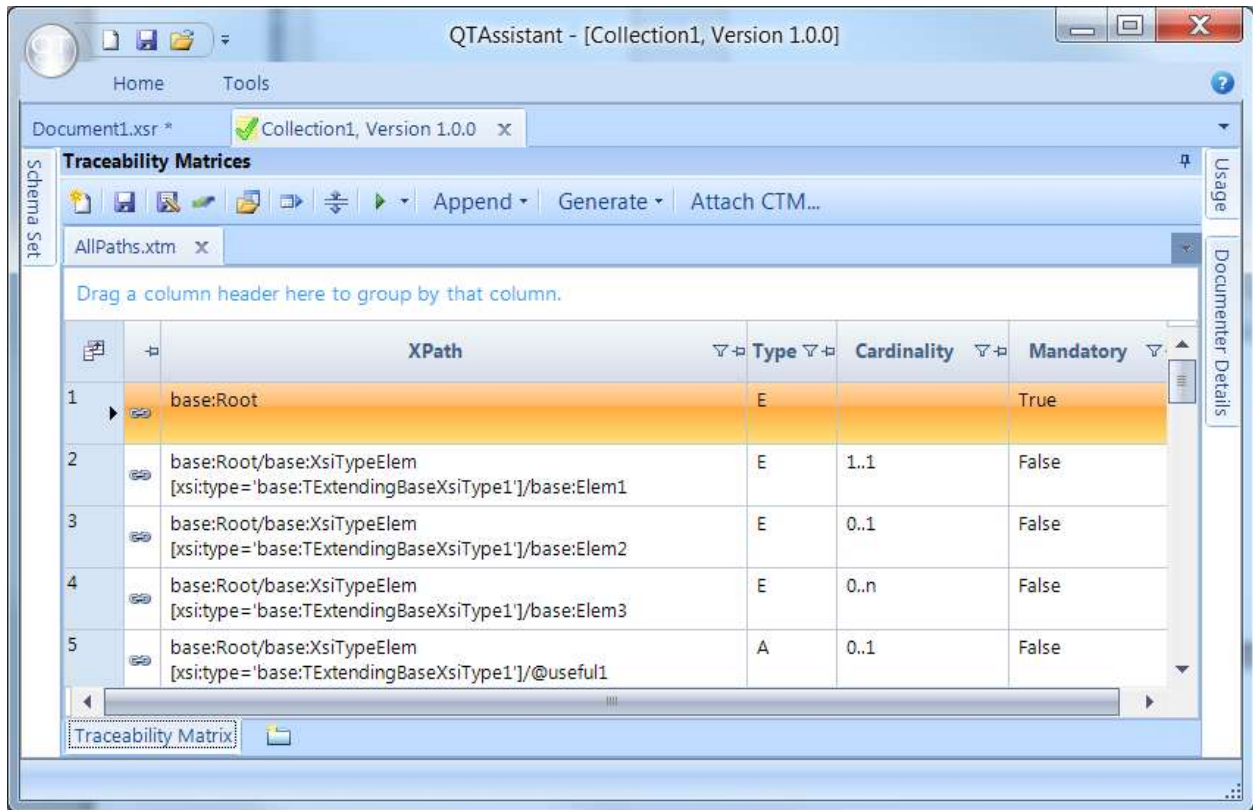
Click on the XSR document tab. Press F4 to display the Properties tool window. Click on the Document node to set its properties; click on Aliases property.



Enter your preferred aliases.

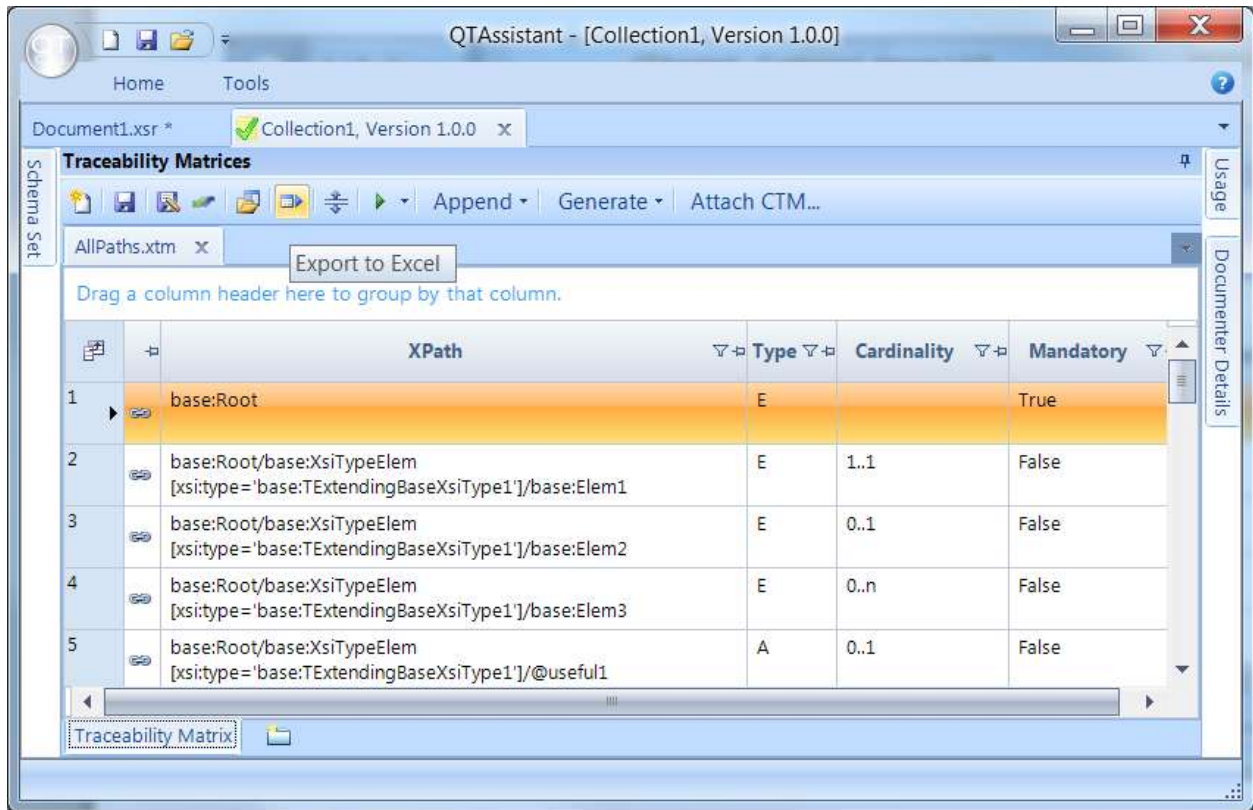


To see the effects of the aliases, you will have to close and reload the XTM (a known issue, it is not an automatic refresh).

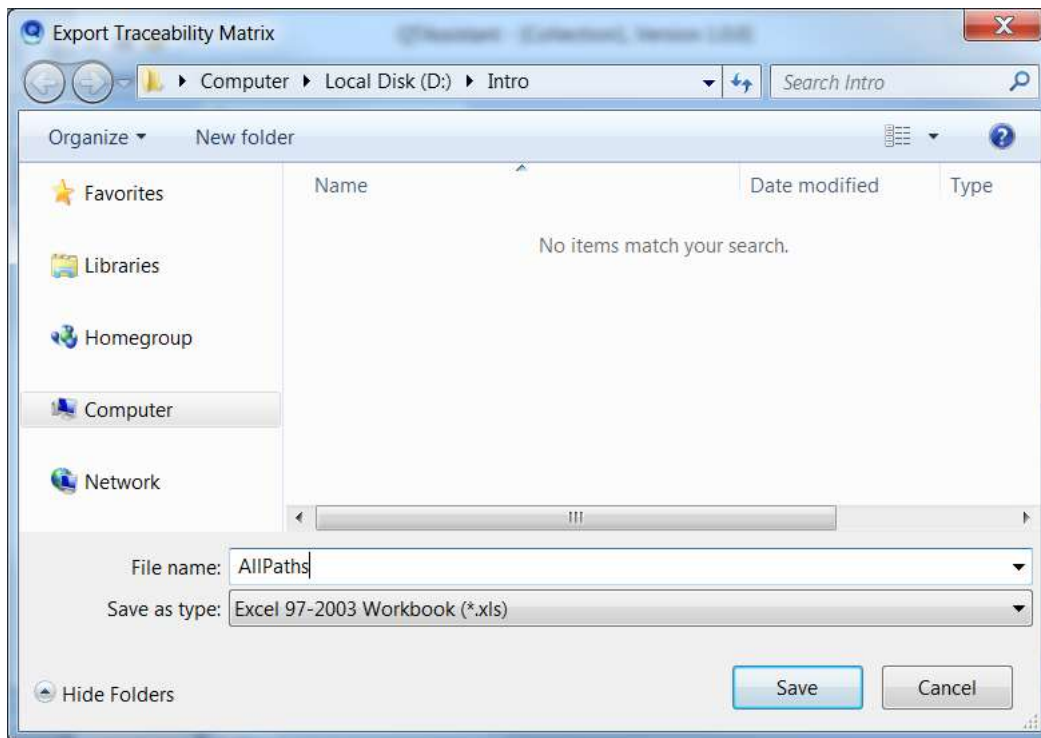


## Step 9: Export to Excel

Click on **Export to Excel**



Choose an Excel file.



The generated file is automatically loaded (assuming you have Excel installed).



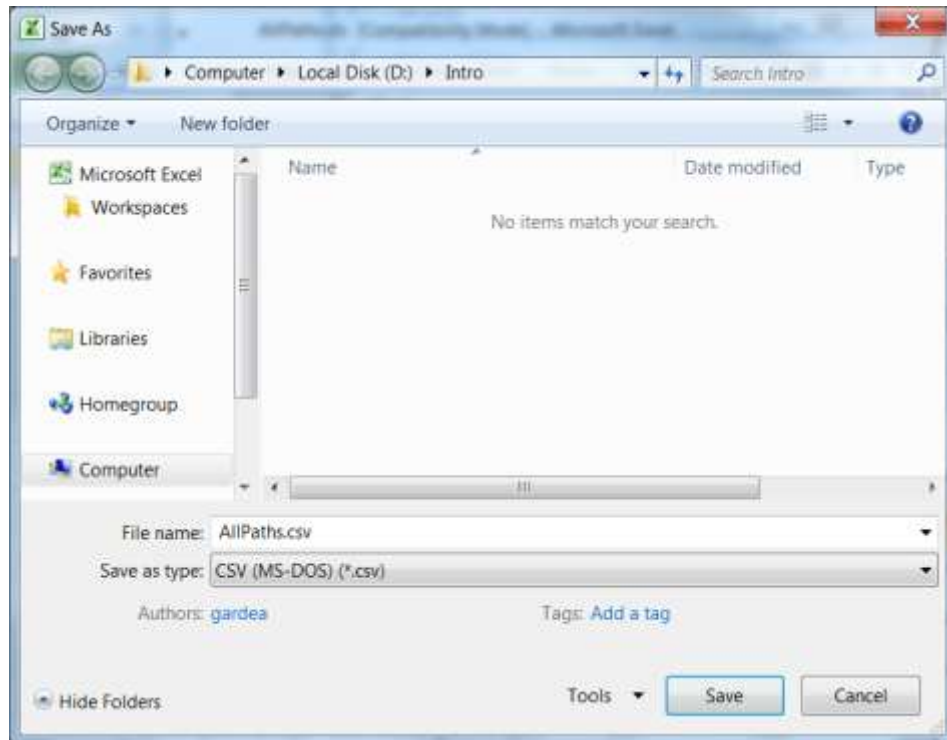
The screenshot shows a Microsoft Excel spreadsheet titled "AllPaths.xls [Compatibility Mode]". The spreadsheet contains a table with the following data:

	A	B	C	D
	XPath	Type	Cardinality	Mandatory
2	base:Root	E		True
3	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:Elem1	E	1..1	False
4	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:Elem2	E	0..1	False
5	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:Elem3	E	0..n	False
6	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/@useful1	A	0..1	False
7	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/@useful2	A	1..1	False
8	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:CoreElem1	E	1..1	False
9	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:CoreElem1/@useful1	A	0..1	False
10	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:CoreElem1/@useful2	A	1..1	False
11	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:CoreElem2	E	0..1	False
12	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 1']/base:EbElem1	E	0..1	False
13	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']/base:Elem1	E	1..1	False
14	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']/base:Elem2	E	0..1	False
15	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']/base:Elem3	E	0..n	False
16	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']/@useful1	A	0..1	False
17	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']/@useful2	A	1..1	False
	base:Root/base:XsiTypeElem[xsi:type='base:TExtendingBaseXsiType 2']	E	0..1	False

## Step 10: Use Excel to convert to CSV

**Note:** Unfortunately, I can't recommend another approach right now.

Click **Save As**



The generated file looks as below.

