THOMSON REUTERS®

RETURN XML GUIDE

FOR TAX YEAR 2021

Last Updated: October 19, 2021

THOMSON REUTERS®

COPYRIGHT NOTICE

Last Updated: October 19, 2021

TABLE OF CONTENTS

CHAPTER 1: RETURN XML

Return XML is a Thomson Reuters income tax compliance productivity tool that allows users to import data into and export data from our tax returns. It is based upon the XML (Extensible Markup Language) standards which define a set of rules for encoding documents in a format which is human and machine readable.

TAX APPLICATION REPORT AND SCHEMA FILES

The tax return data fields that are enabled for Return XML are defined in several files available to customers. For each field enabled, the import/export routine processes the field in batch mode. If the field is not enabled, it will not import or export. These fields are listed in the *Build_Report.XLS and *Schema.XSD files.

EXPORTS

During an export of a tax return locator, Return XML exports the value of every field defined in the corresponding *Build_Report as long as the data is active. Active is defined as other than a constant or carried value.

IMPORTS

For imports, the customer defines the document id combinations and associated data field values in the import file. This file is processed record-by-record against the appropriate tax application *Build_Report. If the document ID in the import record is not located in the *Build_Report, the import file data field value stops the import up front. No part of the import file will import. Another reason an import might fail is if the XML file used for the import is not well formed or valid. As part of the import process, the import file will be compared against the *Schema.XSD. The *Schema.XSD file contains the business rules for well formed, valid XML files.

While constants and carried values are not exported, data values can be imported into the tax return data fields populated by constants and carried values. Imports of this type result in an override symbol being displayed in the Organizer screen. The color of the override symbol resulting from a Return XML import is consistent with that resulting from the manual keyboard entry of an override value.

An exception is that override symbols are not displayed for drop-down and radio button options.

FILE FORMAT

The Return XML import and export files are XML files and have identical format and record structure – one header record followed by any number of data records. The files can be opened in Microsoft XML Notepad, Altova XMLSpy, other commercial XML editors, or a text editor like Notepad.

Opening an XML file in XML Notepad gives the user an easy means of changing the file content:

- Change return number (locator)
- Deleting existing records
- Add new data records
- Change existing data record values

The XML file can then be saved. This in turn can be imported back into a locator. The round-trip import/export capability allows the user to manipulate tax return data without having to be connected to the tax compliance application. Data from any source can be converted into the file format required by Return XML with the requisite data field identification. The data can then be imported into an existing tax return.

IMPORTING AND EXPORTING BY TAX APPLICATION

Return XML is enabled for 1040, 1065, and 1120. It is not enabled for 1041, 706, 709, 990, and 5500. Even if fields are enabled in 1040, 1065, and 1120, a field that is not listed in the *Build_Report file is not enabled for import and export.

CHAPTER 2: RETURN XML FILENAME CONVENTIONS

FILENAME CONVENTIONS: LIMITATIONS AND REQUIREMENTS

These limitations and requirements apply to Return XML import and export filenames:

- The filename for a Return XML import file is limited to 30 characters, including the 4-character filename extension. The filename can include spaces but cannot include special characters prohibited by the Microsoft DOS 8.3 file naming conventions.
- If the filename of a Return XML import file exceeds 30 characters in length, the import file is not uploaded from the user's workstation to the batch servers and cannot be imported into a tax return.
- The filename for a Return XML export file is created by the export program. The filename root is identical to the eight-character filename root of the tax return locator being exported.
- The filename extension for a Return XML import or export file is *.XML. For example, a Tax Year 2021 1040 looks like A1234561.XML.

FILENAME CONVENTIONS: FILE RECORD TYPES

Return XML import and export files are structurally identical – both file types contain the same record types in the same position. For that reason, Return XML export files can be used as import files without modification.

The following shows the first three rows of a Return XML file in Notepad.

```
<?xml version="1.0" encoding="utf-8"?>

<Return xmlns="http://www.thomson.com/Ria/DataExchange">

<ReturnData documentCount="1" Return="X1234L" Flag="H">
```

The first two rows are mandatory and cannot be changed. They tell the system what to expect and the standards followed. The third row has Return, and X1234L is the locator number. The 'ReturnData documentCount' equals one signifying data for one return will follow. If a return is exported, the 'Return' number can then be changed by the client after opening in XML Notepad to another locator number. After saving, the same file can be reused to import into another return.

The flag ‘H’ tells the system to delete all mapped data that exists in a specific document ID and import data from the file into it. An example is if Form 2106 exists in the import file (IRSFORM2106 in the figure below), then all mapped data in the Form 2106 folder will be deleted prior to import. This is to ensure that, upon import, the sequence of properties is maintained, and there is no trapped data. The flag ‘H’ has to remain static and no other flags are supported with Return XML.

Only data in the map will be deleted. If data exists in those screens not in the map, data could be left behind.

Records subsequent to the header can be modified by the customers. They can vary depending on the tax return. Looking at the same file as above, but in XML Notepad, it appears like this:

Figure 2:1

In this 1040 return, we see items for the 2106, brokerage statement, K1s, preparer data, and client profile. This is the beginning of the data records. We will address data records later with how they tie to the *Build Report and the tax application.

FILENAME CONVENTIONS: PROTOCOL FILE SET

The Return XML Protocol File Set was developed as the document ID name reference for use by our customer’s programmers. The programmers can develop their client’s proprietary matching import files, using Return XML as the import/export facility.

The protocol file set is a zip file containing two files for each tax application. They are listed as follows:

TAX APPLICATION	PROTOCOL FILE SET (PFS)
1040	2021 RXML 1040 PFS (YYYY-MM-DD).ZIP
1065	2021 RXML 1065 PFS (YYYY-MM-DD).ZIP
1120	2021 RXML 1120 PFS (YYYY-MM-DD).ZIP

TAX APPLICATION	INDIVIDUAL FILES WITHIN PFS
1040	AIEX1_BUILD_REPORT.XLS TTA1040IEX1Schema.XSD
1065	PIEX1_BUILD_REPORT.XLS TTA1065IEX1Schema.XSD
1120	CIEX1_BUILD_REPORT.XLS TTA1120IEX1Schema.XSD

The (YYYY-MM-DD) designation in the PFS filename represents the release date of the Return XML MAP files documented by the corresponding PFS.

Description of Worksheets inside the Build Report.XLS

Each BUILD_REPORT.XLS (an Excel file) contains the worksheets listed below. All worksheets are sorted by DocumentID, sections, and DataTag.

Org_XML_DataDict_Xref (DATA DICTIONARY) This lists all form and field names in the subject tax application enabled for Return XML. It includes pertinent information for the data fields such as DocumentID, DataTag, Xrefname, data type, length, and so forth. To determine if a tax return Organizer data field is enabled for Return XML, find the DocumentID and DataTag of the subject data field, then search the tax application Data Dictionary for said value. If the value is not included in this tab, the subject data field is not enabled for Return XML import and export.

Txf_XML_DataDict_Xref This is blank as tax forms are not enabled for Return XML.

Wkp_XML_DataDict_Xref This is blank as workpapers are not enabled for Return XML.

XML_Required_Fields This lists Document ID and DataTag combinations that are required to be in an import file if the activity member or group member is to be created. If a required field is missing from the import file but other items in the associated group exist, then the import will be stopped before it begins. We validate the import file before import.

XML_Master_Group_Forms This tab lists master group activities. Master group activities are different types of income or assets that use common fields. The DocumentID determines which master group to associate the common field data.

XML_Child_Forms This tab lists child forms. Child forms are activities with common fields that can be associated underneath different tax forms. For example the document ID for Form 2106 is a child form and appears within the Schedule C in an XML file.

Field_Collections This tab lists groups of fields where, if one field contains data, the others in that group must also contain data.

Field_Collection_Field_Rules This tab lists instances where having data in one field is mutually exclusive with having data in another field.

Field_Collection_Group_Rules This tab lists instances where having data in one group of fields is mutually exclusive with having data in another group of fields. For example, if some parts of a US address are populated, then no part of the corresponding foreign address should be populated.

XML_Discontinued_Fields This lists the field values included in the prior PFS. They have been removed from the mapping for the current period PFS.

XML_Changed_Fields This lists current and prior field values for DocumentID, sections, and DataTag.

XML_New_Fields This lists new fields added with values for DocumentID, sections, and DataTag. They were not in a prior PFS.

XML_Data_Changes This lists field values for current and prior data type, length, and precision. These changes are tied to DocumentID, sections, and DataTag.

XML_String_Values This lists all the string values used in the current PFS by DocumentID, section, and DataTag.

Discontinued_XML_String_Values This lists string values that were in a prior period PFS and are no longer in the current PFS.

New_XML_String_Values This lists new string values for DocumentIDs that have been added to the PFS that were not of the prior PFS.

Changed_XML_String_Values This lists previous string values from a prior PFS to the current string values.

Org_Navigation This worksheet lists the navigation in the return for enabled Organizer fields.

Txf_Navigation This worksheet lists the navigation in the return for enabled Tax Form fields, which is blank here as tax forms are not enabled for Return XML.

Wkp_Navigation This worksheet lists the navigation in the return for enabled Workpaper fields, which is blank here as workpapers are not enabled for Return XML.

Users of the Build Report.XLS and the *Schema.XSD

The BUILD_REPORT.XLS is easier for accountants to understand. They will be able to sort cross-reference values and notice if items are added or deleted.

The *Schema.XSD is easier for programmers to understand. The *Schema.XSD file specifies formal descriptions of the elements in the XML file. These elements have rules for the data such as field length, field type, and field precision.

Validation Routine

Data is only validated on import. The reason to validate is to prevent unacceptable data from impacting the tax return. Validation checks items like field length, field attribute, if required, and structure for groupings. Upon export, the data is not validated. This is so that clients will be able to see data layout in an XML file or so they can use the data for other purposes.

CHAPTER 3: VALIDATING AN XML FILE VIA XML NOTEPAD

There are many XML editors as previously mentioned. We will work through validating an XML data file using XML Notepad 2007. It is free while the others might not be.

In our example, we will use a 1040 tax return with limited data like taxpayer name, SSN, filing status, and one dependent name.

1. Once you have the tax return data entered, export the data via Return XML.
2. Then open XML Notepad, and open the file exported (*.XML). You should see something similar to this.

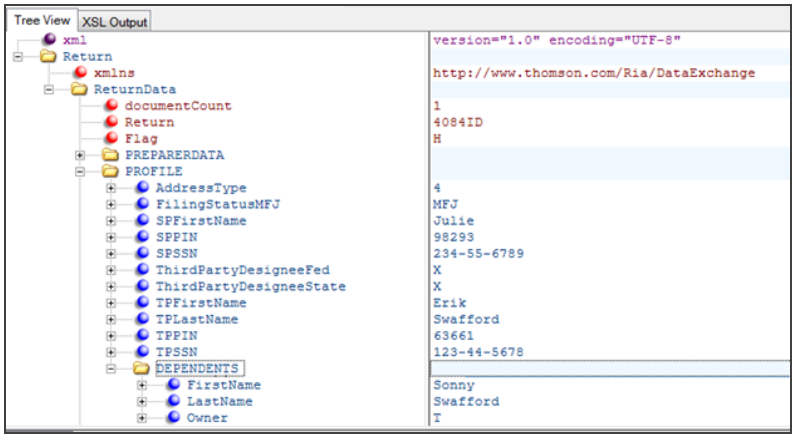


Figure 3:1

3. Select **View**, and then **Schemas**.
4. Push the button under the arrow shown in the figure below.
5. Navigate to the *Schema.XSD file.
6. Select it and push **Open**.

7. You should see something similar to our display. Select **OK** to save the schema selection.

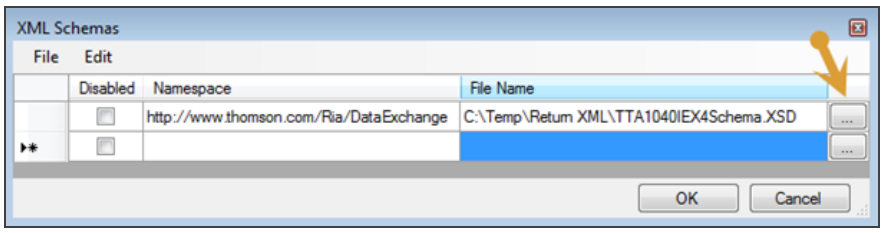


Figure 3:2

8. Underneath the **Dependents**, you will see an "Error List." At this point it is empty as the XML file is valid.

9. Delete FirstName for **Dependents**. An error appears at the bottom of the picture. If you click the error, the cursor moves to the related part in the data.

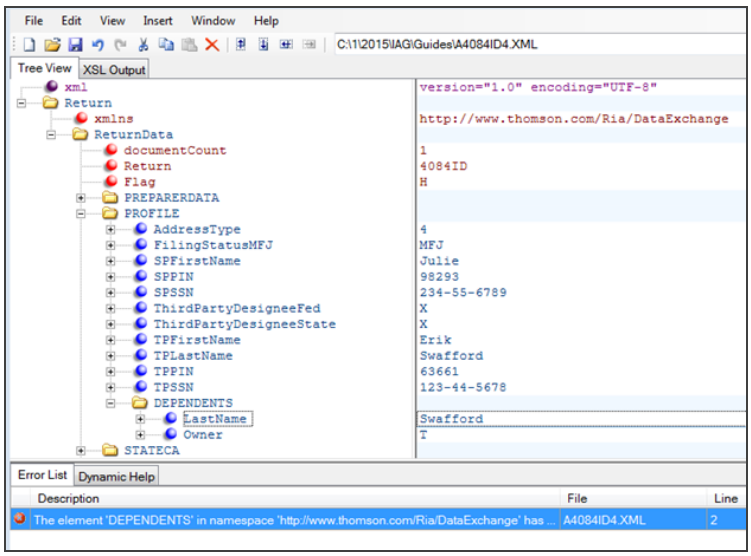


Figure 3:3

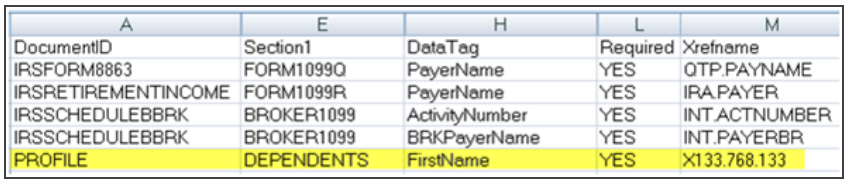
10. Select **Undo**, and the FirstName will revert back into the file.

11. If the file was saved without the Dependent FirstName, the import program will stop the import before it begins.

CHAPTER 4: TYING THE BUILD REPORT .XLS TO A CLIENT XML FILE

Continuing with the example above, we will tie the **Dependents > FirstName** in the XML to the Build Report. Illustrated in the first figure below is the top part of the tab **XML_Required Fields** in the 1040 Build Report. Some columns are hidden so that we can display the relevant ones. Required fields are the fields that must be in the data file for associated data. As demonstrated, if the required field for dependents is missing but other data for dependents are in the file, the import stops.

Build Report.XLS file: Tab "XML Required Fields"



A	E	H	L	M
DocumentID	Section1	DataTag	Required	Xrefname
IRSFORM8863	FORM1099Q	PayerName	YES	QTP.PAYNAME
IRSRETIREMENTINCOME	FORM1099R	PayerName	YES	IRA.PAYER
IRSSCHEDULEBBRK	BROKER1099	ActivityNumber	YES	INT.ACTNUMBER
IRSSCHEDULEBBRK	BROKER1099	BRKPayerName	YES	INT.PAYERBR
PROFILE	DEPENDENTS	FirstName	YES	X133.768.133

Figure 4:1

Looking at this tab in the Build Report, the **DocumentID** column has the item profile highlighted. We can see this in the XML file as a folder in the figure below.

In the tab, the **Section1** column has the item **Dependents**. **Section1** is a group of data within the DocumentID. Any **Section2** will be a group within **Section1**. A third **Section3** will be group data within **Section2**. The sections are only in the XML file if the data requires it. In the XML file shown in the figure below, the **Section1** is the folder **Dependents**.

Exported XML File

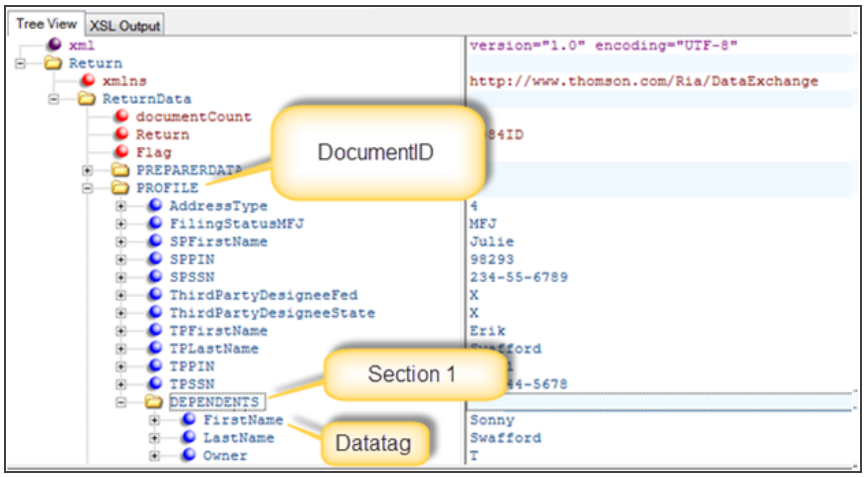


Figure 4:2

Moving over one column in the figure for the tab **XML Required Fields**, the **DataTag** column has FirstName for **Dependents**. When reviewing the **DataTag** FirstName in the XML file, you see the actual data stored for import on the right half of the screen. The bluish purple balls in the file are the DataTags. In our example, the DataTag FirstName value is *Sonny*.

The same rules apply to other items on the **XML Required Fields** tab. For the Broker1099, if the Datatag “BRKPayerName” or “ActivityNumber” are missing from the XML file but other items for Broker1099 exist, the file is no longer valid.

The final column for the tab **XML Required Fields** is **Xrefname**. This is a specific value for each field that allows you to search the spreadsheet quickly if you know it. Each **Xrefname** occurs only once in a tax application. Each field has an **Xrefname**, if enabled.

CHAPTER 5: FIELDS

DETERMINING A FIELD'S DOCUMENTID OR DATATAG

While inside a tax return, right-click an Organizer field. The following pop-up will appear. Next click **FormSource Name**.

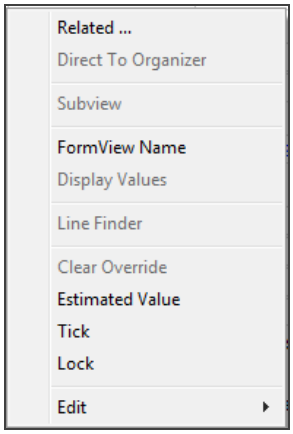


Figure 5:1

A screen containing the items below will appear. In our example, we have selected the taxpayer first name in a 1040 return. Other items are listed, but for our purposes we want the item titled *Return XML*.

Return XML

DocumentID	Section1	Section2	Section3	DataTag	Data Type	Data Length	Data Precision	Description
PROFILE				TPFirstName	Alphanu	15	0	Taxpayer First Name

Figure 5:2

DETERMINING A FIELD'S XREFNAME

While inside a tax return, press **CTRL + SHIFT** and right-click (or left-click) an Organizer field. The following pop-up will appear for the 1040 taxpayer first name. The Xrefname is FED.TPFIRST. This can be cross referenced on the worksheets in the Build Report to find other information for this field.

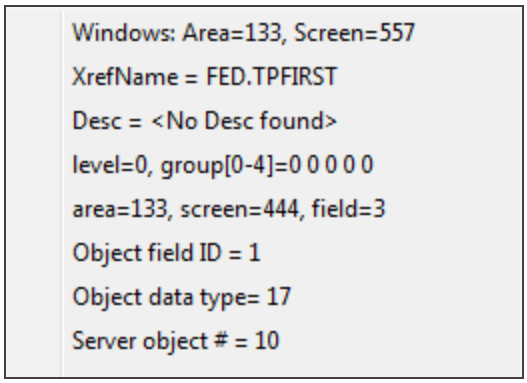


Figure 5:3

VIEWING A REQUIRED FIELD IN A TAX RETURN

When you are in a tax return, required fields are when you see an **Add new XYZ** item. An example is shown below. If the Sonny was not there, you would have to select **Add new First Name**. *Sonny* is data on a required field.

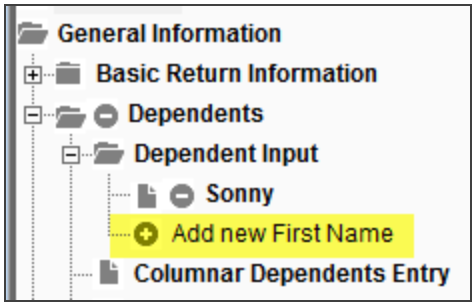


Figure 5:4

CHAPTER 6: DELETING DATA

When an import job is queued, several operations begin:

1. The import file is analyzed for validity.
2. The DocumentIDs are noted in the import file, and associated DocumentIDs in the tax return will have all data deleted prior to import.
3. The import occurs.
4. A full recompute is performed.

Post import, there will be one DocumentID for items that were deleted. For example, if you started with three W-2s at rollover of the return but only imported one, one W-2 will display. If you wish three 1099s to be entirely deleted, you will still have to have one in the import file to tell the program to delete 1099s. The one that remains post import will have to be manually removed from the return if necessary.

The same rules that apply to DocumentIDs apply to Section1, 2, or 3. If there are Section1 group items that are needing to be deleted, then one will still have to be entered. An example is Dependents again. If there are three dependents and there are now going to be two after import, then enter both dependents for Section1. The program will delete all three and then import the two new ones.

There are complex rules to be aware of when importing data onto a seemingly simple DocumentID.

- A simple DocumentID is associated with just one form.
- A complex DocumentID is associated with more than one form. One example of a complex DocumentID is Profile. Within the Profile DocumentID, there are form names for **Bank Information**, **Dependents**, and **General Information** among others, found in the Form Name column. If an import file contains items in the **Profile > General Information** form name, items in the **Profile > General Information** form name will be deleted. Other items in the tax return for **Profile > Bank Information** and **Profile > Dependents** will not be deleted prior to import.

CHAPTER 7: TERMS AND DEFINITIONS

DataTag All enabled fields have a DataTag name. A DataTag is unique within the DocumentID. It is a short name for the data in the input file. In XML Notepad, they are bluish, purple balls in the XML data file. Data is to the right of the DataTag in XML Notepad.

DocumentID Name of the top level folder in the XML file. For example, in the 1040 Return XML, IRSFORM2555 is a top level folder DocumentID with many Datatags under it.

Form Name and Field Names Other ways to denote specific fields. Used by FormSource to give the field a meaningful name.

Juris In the **Build_Report**, there is a **Juris** column on the first tab. The value F denotes federal or S denotes state fields.

Section1, 2, or 3 Grouping of data under a DocumentID. Section2 is underneath Section1. Section3 is underneath Section2. In 1040, the DocumentID IRSFORM2555 is the only one that uses all three.

Validation Term used to verify import file will import. Validation is not performed prior to exporting an XML file.

XML Notepad 2007 Free, recommended program used to open import or export files. These files end in *.xml.

Xrefname Unique field name. Each enabled field name in the tax application has only one Xrefname. Also known as Eorg or Xref.