

**Altova® EBA add-in for Excel, version 2020r2
Enterprise Edition**

User & Reference Manual

Altova® EBA add-in for Excel, version 2020r2

Enterprise Edition

User & Reference Manual

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Published: 2020

© 2020 Altova GmbH

Table of Contents

1	Introduction	5
1.1	Installation and Licensing.....	7
1.2	Installing Additional Taxonomies.....	9
2	Creating a New Report	10
2.1	Entering Data.....	16
2.2	Entering Data into Three-Dimensional Tables (Z-Axis).....	21
2.3	Controlling Accuracy of Cells.....	25
3	Common Tasks	28
3.1	Validating Data.....	28
3.2	Exporting Data to XBRL.....	31
3.3	Importing Data from XBRL.....	32
3.4	Batch Conversion from XBRL to Excel.....	34
4	Command Reference	36
5	COM API	38
5.1	Accessing the API.....	39
5.2	C# Example.....	41
5.3	VBA Example.....	42
5.4	API Reference.....	43
5.4.1	Interfaces.....	43
6	License Information	70
6.1	Electronic Software Distribution.....	71

6.2	Software Activation and License Metering.....	72
6.3	Altova XBRL Add-in Software License Agreement.....	74

Index		75
--------------	--	-----------

1 Introduction

The **Altova® European Banking Authority (EBA) XBRL add-in for Excel** enables your organization to prepare XBRL (eXtensible Business Reporting Language) reports conformant with the EBA (European Banking Authority) XBRL taxonomy, or with related [country-specific XBRL taxonomies](#)⁹.



The **Altova® European Banking Authority (EBA) XBRL add-in for Excel** enables preparers of XBRL supervisory reports to do the following:

- Enter XBRL data in Microsoft Excel, using a predefined template spreadsheet which maps to the XBRL taxonomy.
- Validate the report data directly from Excel, to ensure it conforms to the XBRL taxonomy.
- Export report data from Excel to XBRL format.
- Import data from existing XBRL reports into Excel.
- Batch convert XBRL files to Excel (.xlsx) format.

The supported XBRL taxonomies are as follows: 2.9.1.1, 2.9.1.0, 2.9.0.2, 2.8.1.2, 2.7.0.2, 2.6.0.1, 2.5.0.1, 2.4.1, 2.4, 2.3.2, 2.3.1, 2.3, 2.2, 2.1, 2.0.

The supported country-specific taxonomies are:

- Banque de France (ACPR) COREP SUBCON 2.8.1.1 - COREP, SubConsolidated (Prudential scope)
- Banque de France (ACPR) LCB FT 2.2
- Banco de Portugal (BdP) Reporting Framework 4.0
- Banco de Portugal (BdP) FINREP ITS 2.1
- Bank of England (BOE) Banking Taxonomy 3.3
- Bank of England (BOE) Banking Statistics Taxonomy 1.1

- Deutsche Bundesbank (BBK) 2.9.1.1
- De Nederlandsche Bank Financieel Toetsingskader (DNB FTK) 2.1
- National Bank of Belgium (NBB) FINREP 2.2.2.1 (Financial Reporting)
- National Bank of Belgium (NBB) TREP 1.0.2 (Trading Reporting)
- Single Resolution Board (SRB) LTD 3.0.2 (Liability Data Template)
- Single Resolution Board (SRB) RES 4.0.3
- Single Resolution Board (SRB) SRF 5.1.1

Notes

- The additional local taxonomies are not installed by default. You can download them at <https://www.altova.com/eba-xbrl/download#components>.
- For older versions of supported taxonomies, clear the **Show most current entry points** check box on the "Select Entry Point" dialog box, see [Creating a New Report](#) ¹⁰.

This documentation should be read in conjunction with the supporting documents included with the EBA XBRL Taxonomy, such as:

- "Description of DPM formal model"
- "EBA Architecture for XBRL representation of DPM"
- "EBA XBRL Filing Rules"

System requirements

- Windows 7 SP1 with Platform Update, Windows 8, Windows 10, Windows Server 2008 R2 SP1 with Platform Update or newer
- Microsoft Excel 2019, 2016, 2013, 2010
- .NET Framework 4.0 or later
- If you use Excel 2010, Visual Studio 2010 Tools for Office Runtime (<https://www.microsoft.com/en-us/download/details.aspx?id=48217>) must be installed.

The add-in is available for both Microsoft Excel 32-bit and 64-bit. Microsoft Excel 64-bit is recommended if you need to load big taxonomies such as COREP CON, COREP IND, FINREP. Otherwise, you may see "out of memory" errors when attempting to load such taxonomies with Microsoft Excel 32-bit.

Last updated: 10 March 2020

1.1 Installation and Licensing

To install the **Altova® European Banking Authority (EBA) XBRL add-in for Excel**, download the executable from the Altova Download Center (<https://www.altova.com/download>) and run it. Follow the wizard steps to complete the installation. You will need to accept the license agreement and privacy policy in order to proceed with the installation.

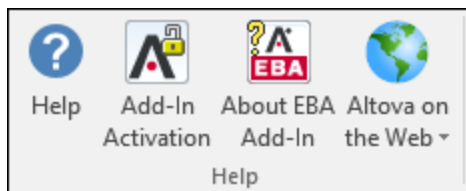
Make sure to download the executable corresponding to your operating system's and Excel platform (32-bit or 64-bit) The 32-bit executable can be installed on both 32-bit and 64-bit Windows; however, it supports only Excel 32-bit. Note that If you have Excel 32-bit and install the 64-bit version of the add-in, you will still be running the 32-bit version.

After installation, a new tab called **EBA** becomes available in the Excel ribbon.

Licensing

To use **Altova® European Banking Authority (EBA) XBRL add-in for Excel**, you need a valid license key code. To purchase a new key code, or request a free evaluation from the Altova website, take the following steps:

1. In the Excel ribbon, click the **EBA** tab.
2. Click **Add-In Activation**.



A dialog box appears with instructions for getting a new license from Altova, or managing an existing license.

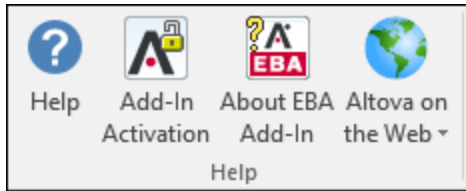
After you purchase a key code from Altova, follow the same steps as above to open the activation dialog box and enter the key code.

Alternatively, you can upload purchased licenses to an Altova LicenseServer running on your organization's network. Altova LicenseServer is a free product that helps organizations manage all Altova licenses in a centralized place. For more information about LicenseServer, see <https://www.altova.com/licenseserver>.

See also [License Information](#) ⁷⁰.

How to view the current version of the add-in

1. In the Excel ribbon, click the **EBA** tab.
2. Click **About EBA Add-In**.



1.2 Installing Additional Taxonomies

The EBA (European Banking Authority) XBRL taxonomy is installed by default on your computer at the same time when you install the add-in.

In addition to the EBA taxonomy available by default, you can optionally download and install the following additional XBRL taxonomies:

- Banque de France (ACPR) COREP SUBCON 2.8.1.1 - COREP, SubConsolidated (Prudential scope)
- Banque de France (ACPR) LCB FT 2.2
- Banco de Portugal (BdP) Reporting Framework 4.0
- Banco de Portugal (BdP) FINREP ITS 2.1
- Bank of England (BOE) Banking Taxonomy 3.3
- Bank of England (BOE) Banking Statistics Taxonomy 1.1
- Deutsche Bundesbank (BBK) 2.9.1.1
- De Nederlandsche Bank Financieel Toetsingskader (DNB FTK) 2.1
- National Bank of Belgium (NBB) FINREP 2.2.2.1 (Financial Reporting)
- National Bank of Belgium (NBB) TREP 1.0.2 (Trading Reporting)
- Single Resolution Board (SRB) LTD 3.0.2 (Liability Data Template)
- Single Resolution Board (SRB) RES 4.0.3
- Single Resolution Board (SRB) SRF 5.1.1

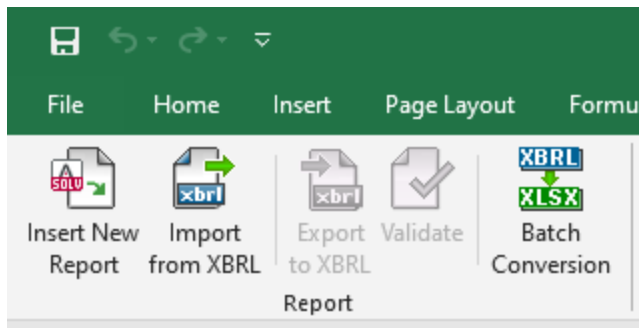
The additional taxonomies are available for download at <https://www.altova.com/eba-xbrl/download>. To install any of these taxonomies, run the downloaded executable file and follow the on-screen instructions. After installation, additional entries become available for selection on the "Select Entry Point" dialog box, see [Creating a New Report](#)¹⁰. To prepare XBRL data in Excel, the same instructions and best practices apply as further described in this documentation. For additional help, consult the documentation of the corresponding authority that supplies the taxonomy.

2 Creating a New Report

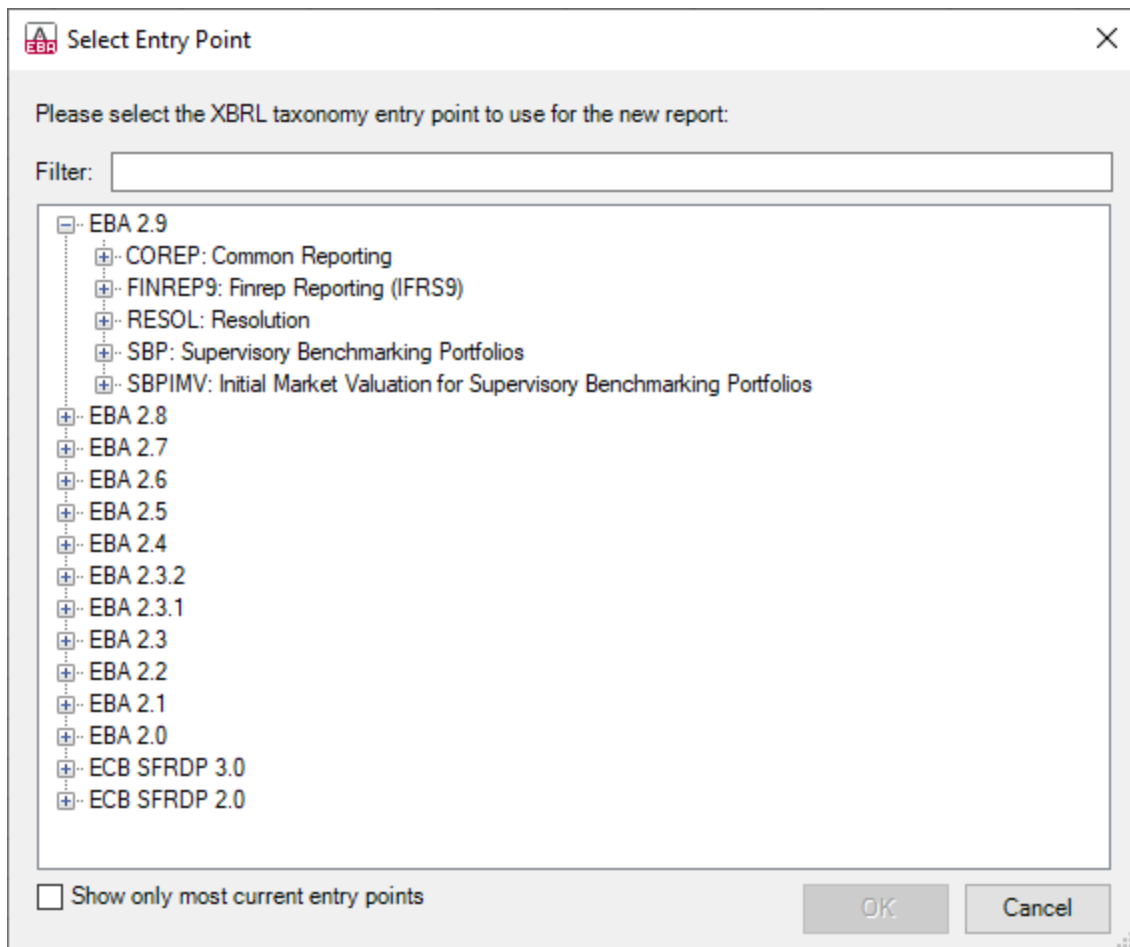
The instructions below show you how to prepare a new XBRL report based on the default EBA (European Banking Authority) taxonomy available in **Altova® European Banking Authority (EBA) XBRL add-in for Excel**. This XBRL taxonomy is installed by default on your computer at the same time when you install the add-in. Additional taxonomies can be installed separately, see [Installing Additional Taxonomies](#)⁹.

To create a new report:

1. In the Excel ribbon, click the **EBA** tab.
2. Click **Insert New Report**.

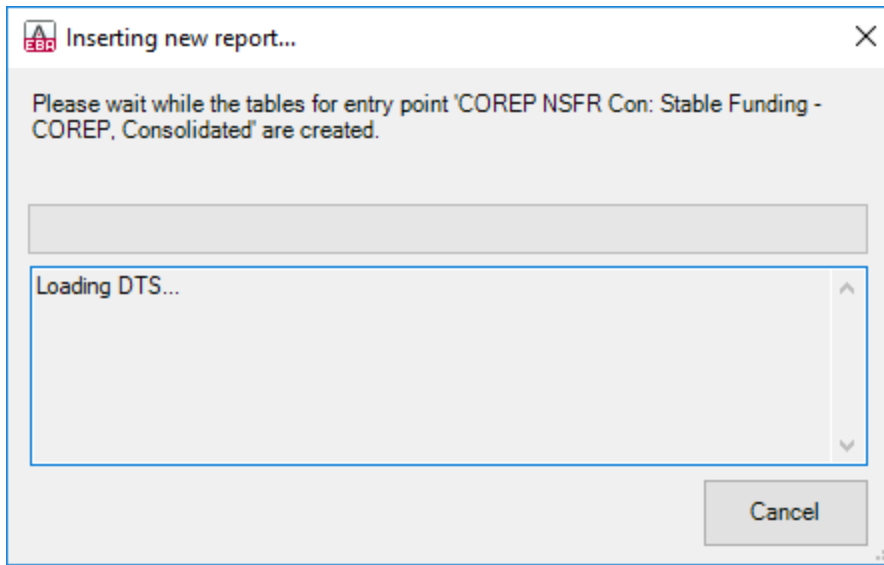


3. When prompted, select the taxonomy entry point corresponding to the report that you wish to create. Use the filter at the top of the dialog box to filter entry points by keywords such as "IFRS", "Funding", and so on. Note that, by default, only the most current entry points for the current version of the add-in are shown. To show all versions, clear the **Show only most current entry points** check box.

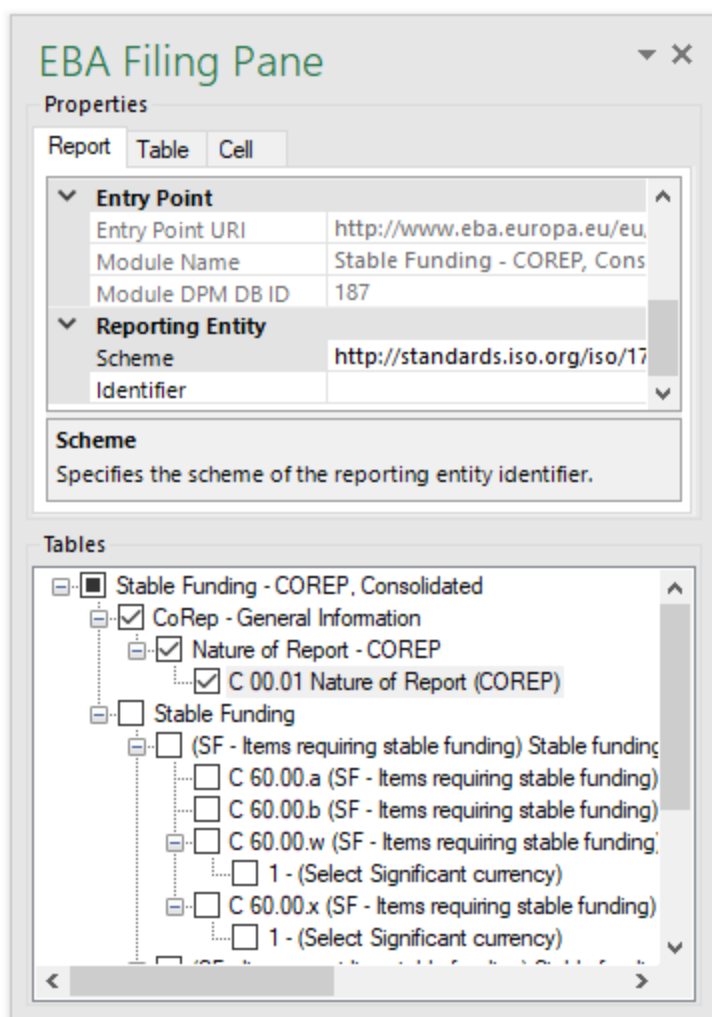


The entry points available by default on the dialog box above are those applicable to the EBA Taxonomy. For more information, refer to the supporting documents of the EBA taxonomy (specifically, see the "EBA Architecture for XBRL representation of DPM"). Additional entry points are available for selection if you downloaded and installed additional XBRL taxonomies from the Altova Download Center (<https://www.altova.com/eba-xbrl/download>).

4. Be patient while the report tables are loaded into Excel. During this operation, a dialog box informs you of the progress, for example:



Once the report tables have finished loading, notice the **Tables** section in the **EBA Filing Pane**.



5. Select the check boxes next to tables that you want to include in the report. Notice that each included table appears on a new sheet in the Excel book.

You can now start entering data in tables, validate it, and export it to XBRL format. See the following topics for more information:

- [Entering Data](#) ¹⁶
- [Controlling Accuracy of Cells](#) ²⁵
- [Validating Data](#) ²⁸
- [Exporting Data to XBRL](#) ³¹

EBA Filing Pane

EBA Filing Pane is the area where you can include or exclude tables from the report, view information about each cell, and view or set various report properties. By default, this pane is visible; you can show or hide it by clicking **Toggle EBA Filing Pane** command in the ribbon. As illustrated above, the **EBA Filing Pane** consists of two main sections: **Properties** and **Tables**.

Properties

The properties displayed in the **EBA Filing Pane** directly affect the content of the XBRL instance file that will be created when you export the XBRL instance. To view what each property does, click it and observe the description displayed in the gray box under the grid. Properties that are grayed out are read-only; otherwise, you can edit a property by typing text or selecting a value as applicable.

The **Scheme** and **Identifier** properties under "Reporting Entity" are typically provided by the relevant competent authorities.

Also note that, even though some property values begin with "http" (for example, **Entry Point URI, Scheme**), they do not necessarily point to live web resources and thus should not be considered dead links. To resolve entry point URIs, the add-in uses a catalog mechanism that maps URIs to files on the local system. This is in large part due to the size of the taxonomies and the fact that they contain thousands of files. Accessing the taxonomy files over the Internet would result in extremely slow performance, even if their issuing organizations served them that way.

Properties are grouped into the following three tabs:

- **Report** - This tab displays properties applicable to the entire report (one report corresponds to one Excel workbook).
- **Table** - This tab displays only properties of the currently selected table. A table normally corresponds to a single Excel worksheet. Therefore, whenever you click inside a new Excel sheet, the properties are re-drawn to reflect the new worksheet.
- **Cell** - This tab displays only properties of the currently selected cell. Whenever you click a new cell, the cell properties are re-drawn accordingly.

You can set the accuracy-related properties at report, table, or cell level. For more information, see [Controlling Accuracy of Cells](#) ²⁵.

Tables

To include a table in the report at any time, select its corresponding check box in the **EBA Filing Pane**. Each included table appears on a new sheet in the Excel book. To go to a specific sheet, either navigate to it using the standard Excel way, or click the corresponding table in the **EBA Filing Pane**. To remove a particular table from the report, clear the check box next to it. Tables that are not selected will not be included in the report.

Some tables support a Z-Axis (a third dimension). For information about adding a Z-Axis to a table, see [Entering Data into Three-Dimensional Tables \(Z-Axis\)](#) ²¹.

Each report table displayed in the **EBA Filing Pane** is XBRL-bound, meaning that data you enter directly in the table cells will be reflected in the XBRL instance file when the report is ready, see [Exporting Data to XBRL](#) ³¹. While the report data is work in progress, you can save the Excel workbook and reopen it at any time later, just like a standard Excel workbook.

Any sheets that contain tables are bound to the XBRL taxonomy, so they must not be deleted. It is also not recommended to rename such sheets. If necessary, you can add new sheets to the workbook; however, such sheets would not be bound to the XBRL taxonomy and consequently be ignored when you

generate the XBRL instance file.

2.1 Entering Data

You can populate a report with data either by entering data into cells manually or by pasting values. With some cells, you can select a value from a predefined list (such as countries or currencies). Also, in some report tables, you may need to add new rows or columns. The following is a list of tips and best practices for entering data.

Editable versus non-editable cells

As a general rule, gray cells must not be edited. Only cells that are included in the XBRL-bound area (delimited by the table boundaries) are to be edited. For guidance with respect to the purpose of the cell, and data expected to be entered, consult the cell information (properties) displayed in the **EBA Filing Pane**, in the **Cell** tab.

Pasting data

If you paste data from multiple columns, the number of pasted columns should correspond to the number of columns in the predefined sheet. If you accidentally paste a larger number of columns, or if you type text outside the default table, unwanted columns may appear outside the XBRL-bound area. To delete unwanted columns, right-click the cell and select **Delete > Table Columns**. To prevent Excel from adding new columns and rows automatically, go to **File > Options > Proofing > AutoCorrect Options > AutoFormat As You Type > Apply as you work**, and click to clear the **Include new rows and columns in table** check box.

When pasting data, it is recommended to keep only the values (and not the formatting). Namely, select the **Paste Values** option when pasting cells or rows.

Actual versus displayed cell value

While generating the XBRL instance file, the add-in ignores any cell formatting information and exports the *actual* value of the Excel cell. However, bear in mind that, as part of Excel functionality, the *actual* value may be different from the value displayed in the cell, because of the cell formatting information. You can view at any time the actual value (the one that will be written to the XBRL instance) in the formula bar of Excel. Consider this example:

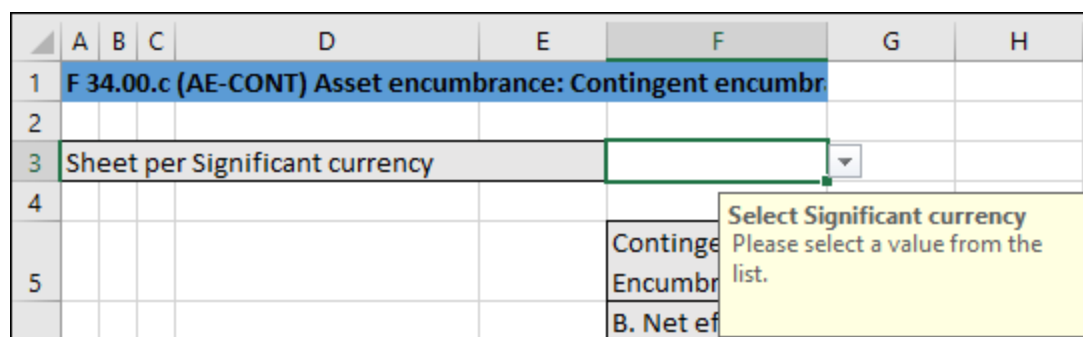
	A	B	C	D	E	F	G
1	C 66.01.a Maturity ladder. Total. Overnight and higher maturity						
2							
3							Overnight
4							0020
5	Liabilities resulting from					0010	12.345,00

In the example above, the value that will be written to the XBRL instance is *12345*. Note that the number accuracy reported in the XBRL instance file also depends on the value you selected for the "Accuracy"

properties (see [Controlling Accuracy of Cells](#) ²⁵).

Enumeration values

Some cells expect a fixed predetermined value (for example, cells that represent currencies or countries). In this case, the add-in displays a small tooltip when you click the cell. You can pick up the required value from the drop-down list:



To view the full list of all possible values, click the cell and observe the cell properties in the **Cell** tab of the **EBA Filing Pane**.

Conditional cells

In some tables, you must first fill out a cell value in order to make other cells of the table editable. For example, in the table below, the cell F3 must be first be filled out before all other cells in the same column become editable.

	A	B	C	D	E	F	G	H	
1	F 34.00.c (AE-CONT) Asset encumbrance: Contingent encumbr								
2									
3	Sheet per Significant currency								
4									
5						Continge Encumbr			
6						B. Net ef 10% depreciation of significant currencies			
7						Additional amount of encumbered assets			
8						026			
9	Carrying amount of selected				010				
10	Derivatives				020				
11	of which: Over-The-				030				
12	Deposits				040				
13	Repurchase agreements				050				
14	of which: central banks				060				
15	Collateralised deposits				070				
16	of which: central banks				080				
17	Debt securities issued				090				

Select Significant currency
 Please select a value from the list.

... | F 33.00.b | F 34.00.a | F 34.00.b | **F 34.00.c (1)** | +

Cells with multiple values

Depending on the XBRL taxonomy, some reports might have facts that represent an arbitrary list of comma-separated multiple values. Consequently, in Excel, the corresponding cell also requires multiple values to be entered in the same cell.

To enter data for such cells, first expand the drop-down list, and then click all items that qualify.

	A	B	C
1	S.25.01.21.03 Basic Solvency Capital Requirement (U		
2			
3			USP
4			C0090
5	Life underwriting risk	R0030	
6	Health underwriting risk	R0040	
7	Non-life underwriting risk	R0050	1 - Increase in the amt 2 - Standard deviation 3 - Standard deviation 4 - Adjustment factor f 5 - Standard deviation 9 - None
8			
9			
10			
11			

Alternatively, you can type all the numeric values, separated by a comma, as shown below. Remember that you can view all possible values of a cell in the **Cell** tab of the **EBA FilingPane**.

	A	B	C
1	S.25.01.21.03 Basic Solvency Capital Requirement (U		
2			
3			USP
4			C0090
5	Life underwriting risk	R0030	
6	Health underwriting risk	R0040	2,3
7	Non-life underwriting risk	R0050	

After you exit a multi-valued cell, it is automatically re-drawn to display all selected values in a readable form (even though you may have entered only numbers).

Adding new rows

With some tables, you may need to create new rows. For example, this is the case for table "C 10.02" available through the entry point **EBA 2.6 COREP CON**. You can add new rows either using the standard Excel commands or shortcuts, or by clicking the **Add Row** button in the ribbon. For example, to add a new row to the table "C 10.02" of the entry point mentioned above, do one of the following:

- In the Excel ribbon, click the **EBA** tab, and then click **Add Row | Insert Row Below**. Note that the commands to insert or delete rows are enabled only if the table supports adding new rows.
- Click the rightmost cell of the last row in the table, and press **Tab**.
- Right-click a cell in the empty row, and select **Insert | Table Row Below** from the context menu.

Note: Any newly added rows must be within the XBRL-bound area of the table, clearly delimited by black lines.

Adding new columns

Some tables may need extra columns to be added. In other words, they can grow horizontally. You can add new columns to such tables in one of the following ways:

- In the Excel ribbon, click the **EBA** tab, and then click **Add Column**. Note that this command is enabled only if the table supports adding new columns according to the XBRL taxonomy.
- Click the **Add** button that appears next to the rightmost column of a table.

	A	B	C	D
1	S.04.01.01.03 By EEA member (localization of activity)			
2				
3	Sheet per Sheets			
4				
5			By EEA member	
6			Business underwritten in the considered country through FPS, by the undertaking or any EEA branch	
7				
8			C0100	Add
9	Premiums written	R0020		
10	Claims incurred	R0030		
11	Commissions	R0040		

- Right-click a table cell, and then select **Insert | Table Columns to the Right** from the context menu.

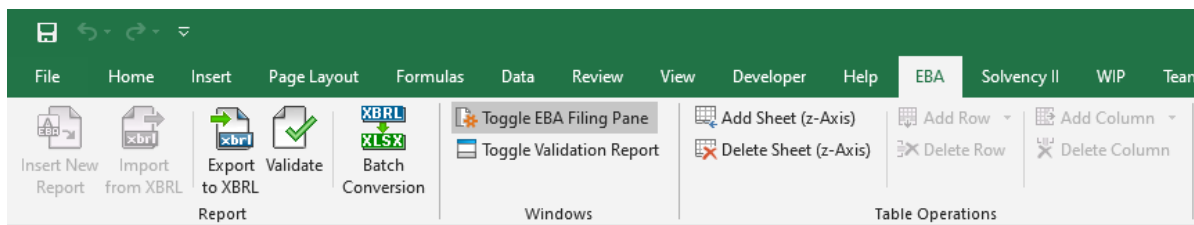
2.2 Entering Data into Three-Dimensional Tables (Z-Axis)

Most of the report tables have only two dimensions: the x-Axis (columns) and the y-Axis (rows). However, there are some tables where you may need to enter data into a third dimension (the z-Axis). An example of such a table is the "F 34.00.c (AE CON)" table available through the entry point **AE CON: Asset Encumbrance, Consolidated**. This table may need an additional sheet for each currency. As shown below, cell F3 is a drop-down list from where you can select a currency.

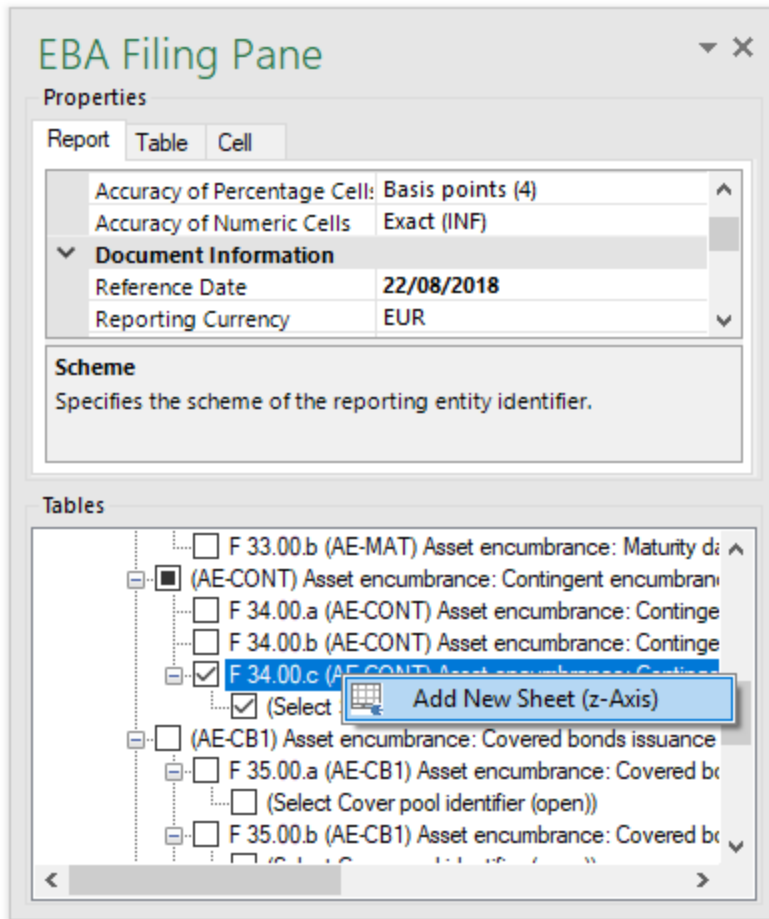
	A	B	C	D	E	F	G	H
1	F 34.00.c (AE-CONT) Asset encumbrance: Contingent encumbr							
2								
3	Sheet per Significant currency							
4								
5								

In cases such as the one above, you can add a new sheet along the z-Axis (third dimension) of the table, as follows:

1. In the Excel ribbon, click the **EBA** tab.
2. Click the **Add New Sheet (z-Axis)** button. Note that the commands to insert or delete new z-Axis sheets are enabled only if the table supports adding a z-Axis.



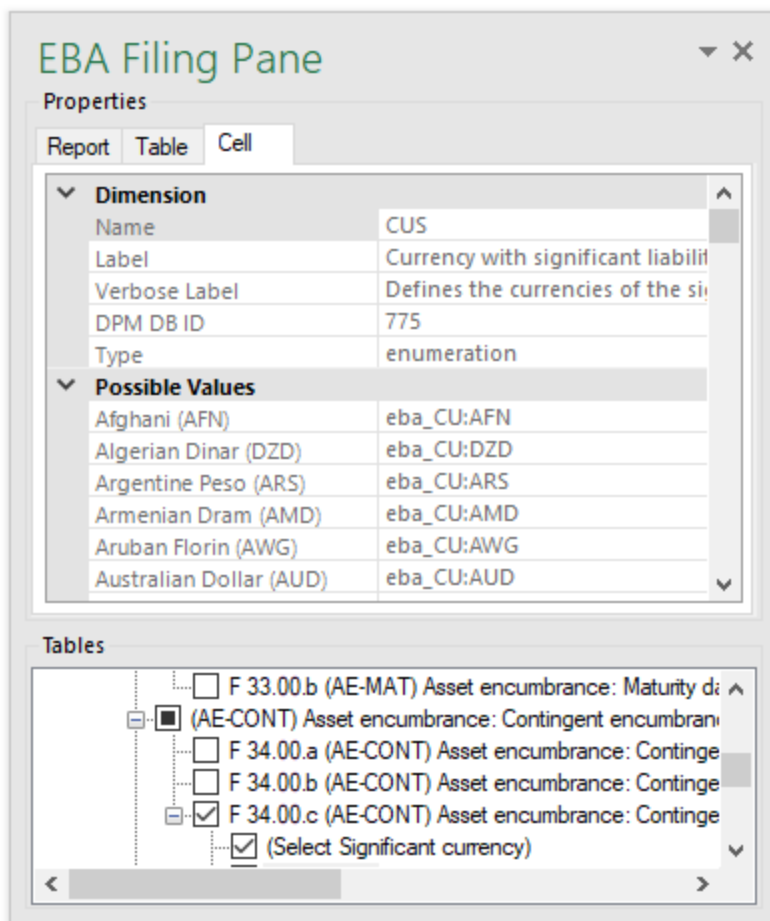
Alternatively, right-click the table in the **EBA Filing Pane** and select **Add New Sheet (z-Axis)** from the context menu, for example:



This creates a new sheet that displays the third dimension of the table (z-Axis). The sheet representing the z-Axis always has an indicative name that resembles the original table. In this example, if you select "Euro" as currency, the new sheet is called "F 34.00c (EUR)".

Data from the third dimension (z-Axis) of a table are displayed as new sheets in Excel. Therefore, three-dimensional tables span across more than one sheet. This is an exception to the rule that one Excel sheet corresponds to one table in the XBRL report. In the XBRL instance, data that belongs to the z-Axis will be, however, correctly reported as part of the same table.

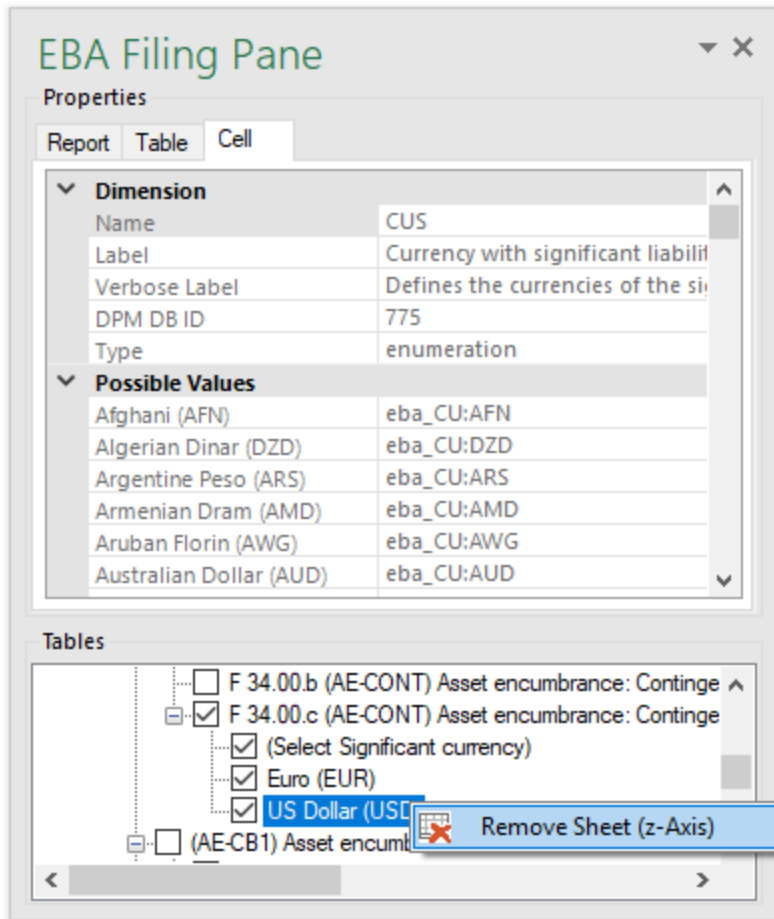
When you click a cell that represents z-Axis, all the possible values for the drop-down list are displayed in the **Cell** tab of the **EBA Filing Pane**, for example:



Deleting z-Axis sheets

You can delete sheets that contain data from the third dimension (z-Axis) in more than one way.

1. Select the required sheet (or click the corresponding entry in the "Tables" section of **EBA Filing Pane**).
2. Do one of the following:
 - In the **EBA** tab, click **Remove Sheet (z-Axis)**, or
 - Right-click the table in the **EBA Filing Pane** and select **Remove Sheet (z-Axis)** from the context menu.

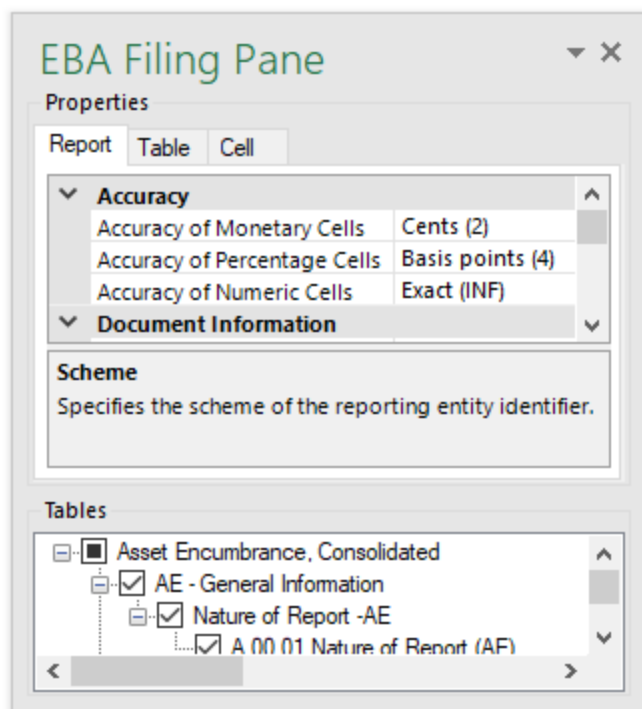


2.3 Controlling Accuracy of Cells

The accuracy of monetary and other numeric values in the XBRL report can be controlled by setting the following report properties:

1. Accuracy of monetary cells
2. Accuracy of numeric cells
3. Accuracy of percentage cells

These properties are available in the **EBA Filing Pane**, in the **Properties** group.

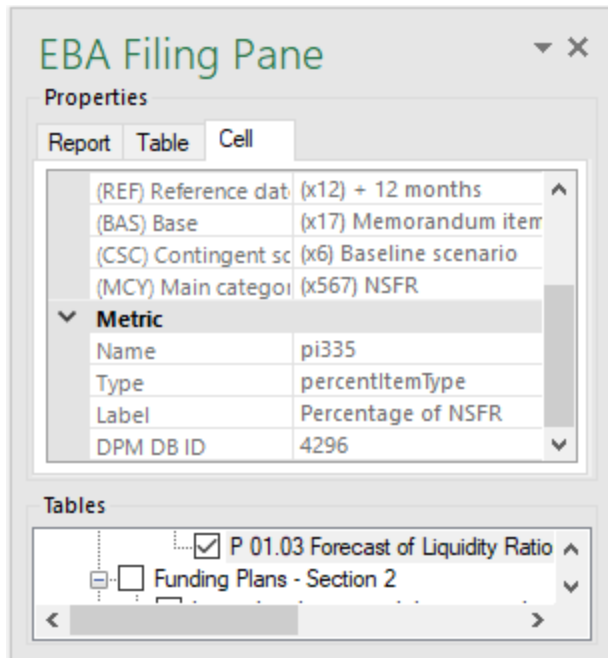


The property **Accuracy of Monetary Cells** applies to numeric cells in the report that represent a monetary value. For example, if you select entry point **FP IND: Funding Plans, Individual**, Table "P 01.03", this property affects all cells of type `monetaryItemType`.

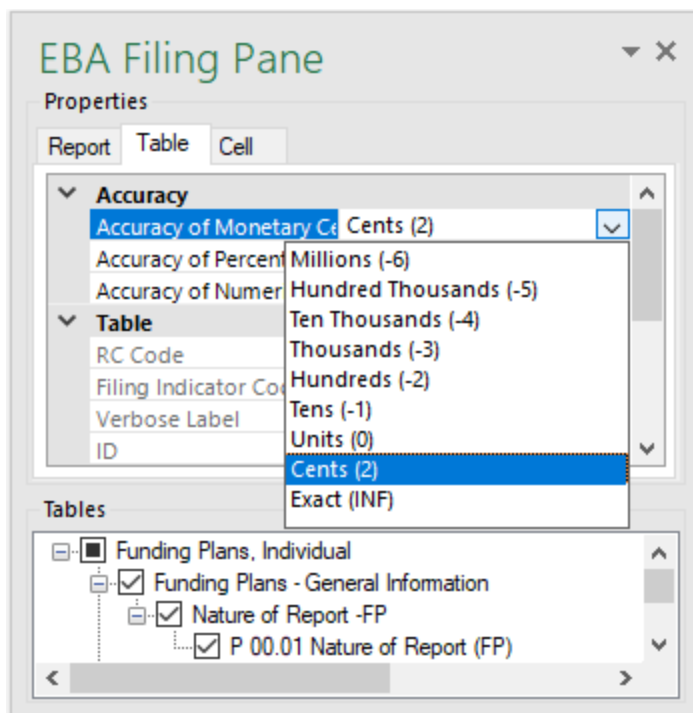
The property **Accuracy of Numeric Cells** applies to numeric values that have no unit (excluding percentage values, which have their own data type). For example, if you select entry point **SPB CON**, Table "C 101.00", this property affects the column "Maturity (160)".

The property **Accuracy of Percentage Cells** applies to values that represent a percentage. For example, if you select entry point **FP IND: Funding Plans, Individual**, Table "P 01.03", this property affects all cells of type `percentItemType`.

To view the type of a cell, first select a cell, and then click the **Cell** tab in the **EBA Filing Pane** and observe the **Type** property, for example:



The accuracy you select from the **EBA Filing Pane** controls the accuracy that will be written for this fact in the XBRL instance file. More specifically, the "Accuracy" properties are bound to the **decimals** attribute in the XBRL instance file. For example, in the image below, the accuracy value indicated in the brackets corresponds to the value of the **decimals** attribute in the XBRL instance file.



By default, accuracy is set as follows:

- The **Accuracy of Monetary Cells** is set to **Cents (2)**, which sets the value of the **decimals** attribute in the XBRL instance to "2"
- The **Accuracy of Percentage Cells** is set to **Basis Points (4)**, which sets the value of the **decimals** attribute in the XBRL instance to "4".
- The **Accuracy of Numeric Cells** is set to **Exact (INF)**, which sets the value of the **decimals** attribute in the XBRL instance to "INF".

You can set the accuracy-related properties at report, table, or cell level (see the corresponding tabs in the image above). If you set accuracy at multiple levels, keep in mind that the more specific property always overrides the more generic one. For example, the accuracy set at cell level takes priority over the one set at table level. Likewise, the accuracy set at table level takes priority over the one set at report level.

For monetary and numeric cells, the accuracy value can be either positive or negative.

A positive value N specifies the accuracy of up to N digits to the right of the decimal place. For example, the value 2 specifies the accuracy to be in cents, while the value 3 specifies the accuracy to be up to mills.

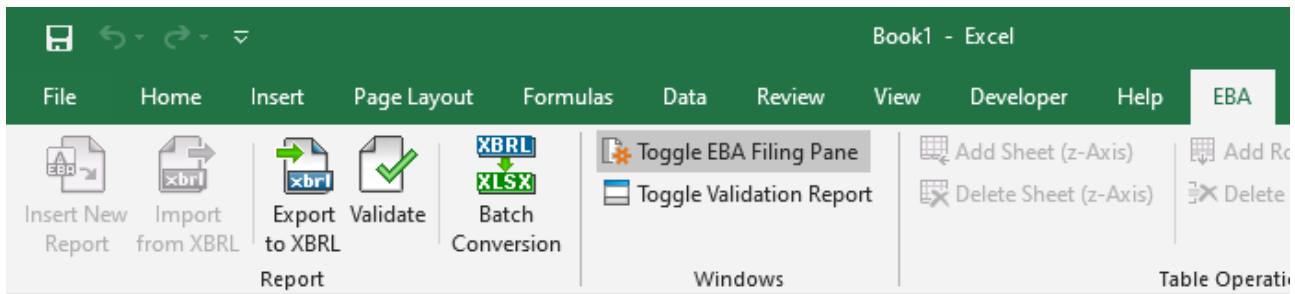
A negative value N specifies the accuracy of up to N digits to the left of the decimal place. For example, the value -3 specifies the accuracy to be up to thousands, while the value -6 specifies the accuracy to be up to millions.

3 Common Tasks

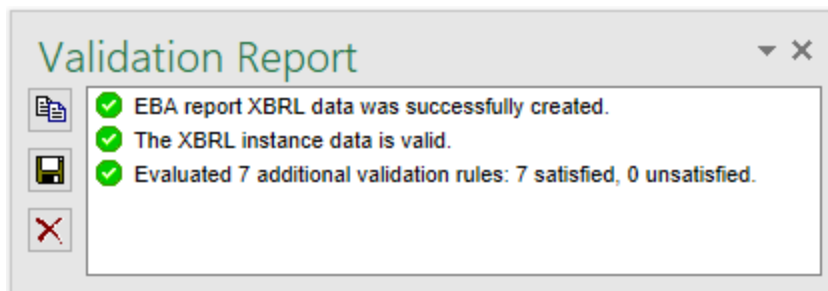
3.1 Validating Data

Validation ensures that the XBRL data you are filing conforms to the XBRL specification. The report data should be validated before you export it to XBRL. You might also want to validate data progressively, after each action that could potentially render it invalid (for example, after pasting new rows into the spreadsheet).

To validate data, click the **Validate** button in the **EBA** tab of the Excel ribbon.






Be patient while **Altova® European Banking Authority (EBA) XBRL add-in for Excel** performs the validation process. To validate XBRL data, the add-in creates a temporary in-memory XBRL instance. When validation of the in-memory instance completes, a validation report similar to the one below is displayed.



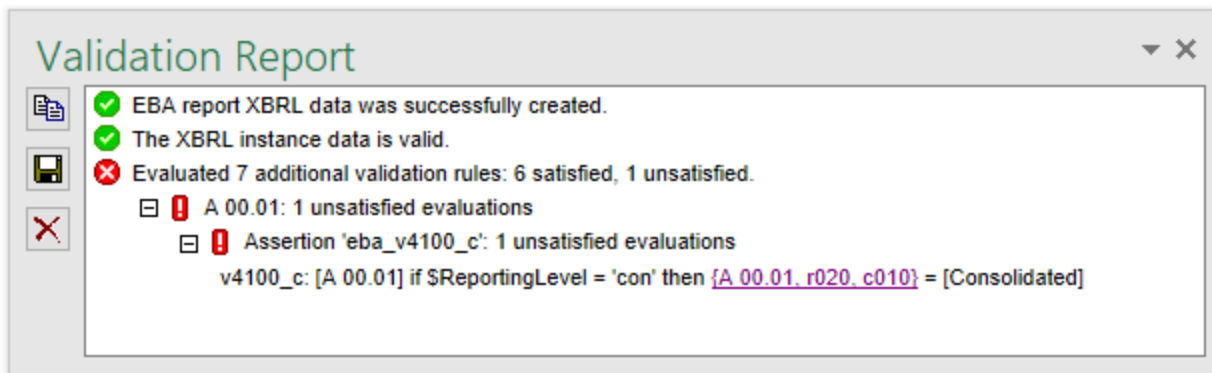
The validation result can be any of the following:

Message type	Meaning
	The instance data is valid.
	The instance data is valid, but has inconsistencies or warnings.
	The instance data is not valid.


The Validation Report dialog box may additionally display any of the following message types: information messages, warnings, and errors.

Message type	Meaning
	Denotes an information message. Information messages do not make the XBRL instance invalid.
	Denotes a warning message, or an inconsistency. Warnings and inconsistencies do not make the XBRL instance invalid.
	Denotes an error. If there are validation errors, the XBRL instance is not valid, and you will need to edit the report data so as to resolve each error before proceeding with the export to XBRL. Note: During validation, the add-in checks XBRL formula assertions and reports them as errors. If you are using the Altova RaptorXML+XBRL Server for validation (https://www.altova.com/raptorxml), XBRL formula assertions may be optionally configured not to be reported as errors.


When a report fails successful validation, the Validation Report window may display links to the cell where the error originates. To quickly find a cell where the error originates, click the underlined text and the cursor will be positioned automatically on the required cell. Note that there are cases where multiple cells are involved in a single validation check; in such cases, clicking on the error link will select just one of the affected cells.




To copy the contents of the validation report to clipboard:

- Click  **Copy**, and then paste into a target file (for example, an email). Alternatively, right-click inside the Validation Report window and select **Copy All Messages** from the context menu.

To save the validation report as text or HTML:

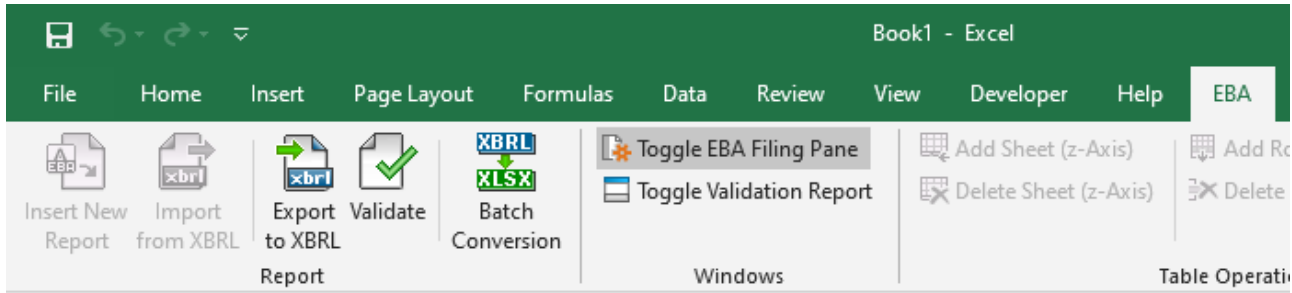
- Click  **Save**. Alternatively, right-click inside the Validation Report window and select **Save Validation Report** from the context menu.

To clear the validation report:

- Click  **Clear**. Alternatively, right-click inside the Validation Report window and select **Clear** from the context menu.

3.2 Exporting Data to XBRL

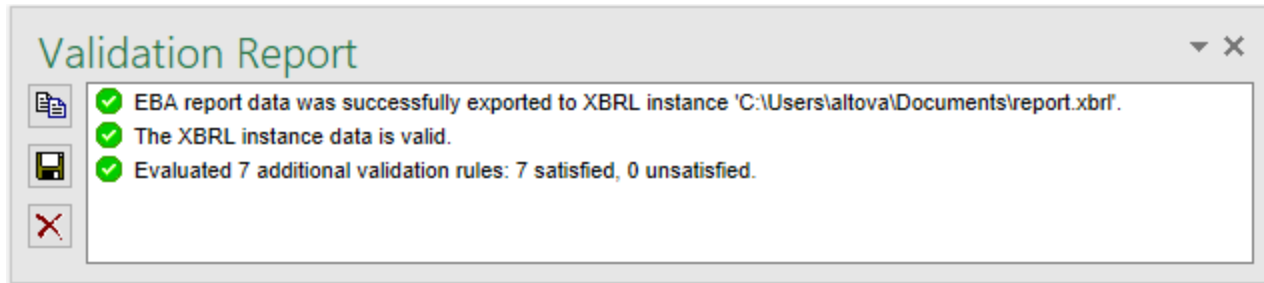
Once your report is ready and valid (see [Validating Data](#)²⁸), you can generate the XBRL instance file. To do this, click the **EBA** tab, and then click **Export to XBRL**.



By default, instance files are saved as files with .xbrl extension. If you need the exported file to have another extension (for example, .xml), type the file extension in the Export dialog box.

While the XBRL instance is being created, a dialog box which informs you about the progress may be displayed for a short time.

During the export operation, data is automatically validated. Any errors, inconsistencies and warnings are reported on the screen after the export finishes.



Note: Cell values that are not valid (that is, cells that don't conform to the data type of the underlying XBRL concept) prevent the report from being exported.

For tips on how to avoid data formatting errors, see [Entering Data](#)¹⁶. Note, however, that not all XBRL validation errors might be related to incorrect formatting. Some errors might occur because entered data does not meet the XBRL validation rules applicable to the report you are filing.

3.3 Importing Data from XBRL

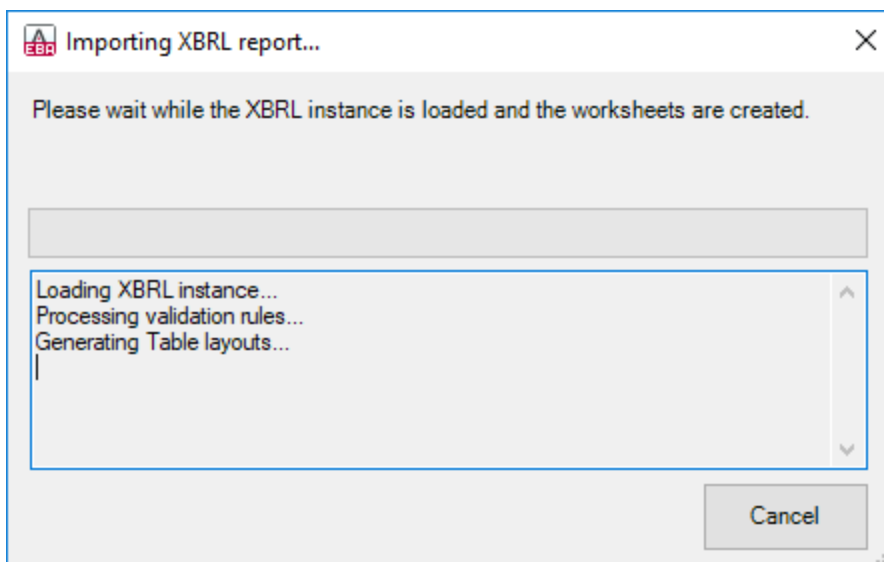
You can import data from existing instances of XBRL reports into Excel (typically, files with .xbrl extension). For the import to be successful, the imported instances must be valid XBRL reports. They may be either reports you have previously generated using the **Altova® European Banking Authority (EBA) XBRL add-in for Excel**, or reports that you received from other parties.

To import an existing XBRL instance file into Excel:

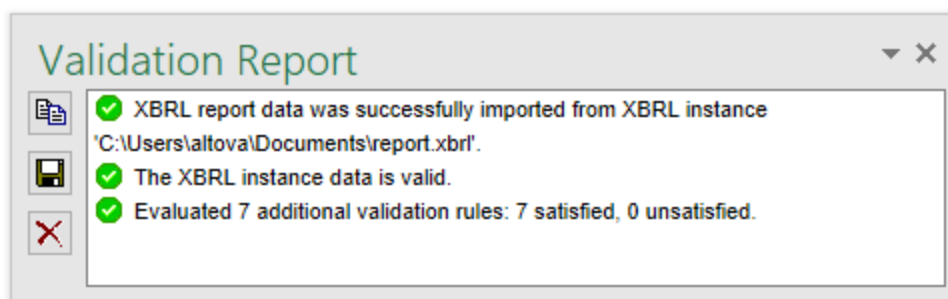
1. In the Excel ribbon, click the **EBA** tab.
2. Click **Import from XBRL**, and browse for the XBRL instance file.

Note: If a report is already open in Excel, the **Import from XBRL** button is disabled. To enable the command, save and close the current report (workbook) and create a new workbook.

During the import operation, a dialog box informs you about the progress:



While the report data is loaded into Excel, it is automatically validated. A dialog box notifies you about potential warnings, inconsistencies, or errors (see also [Validating Data](#)²⁸).



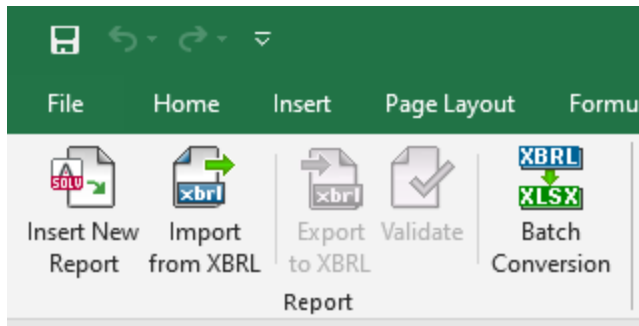
Note: During import, the add-in validates XBRL formula assertions. A report will be imported even if it contains unsatisfied assertions.

3.4 Batch Conversion from XBRL to Excel

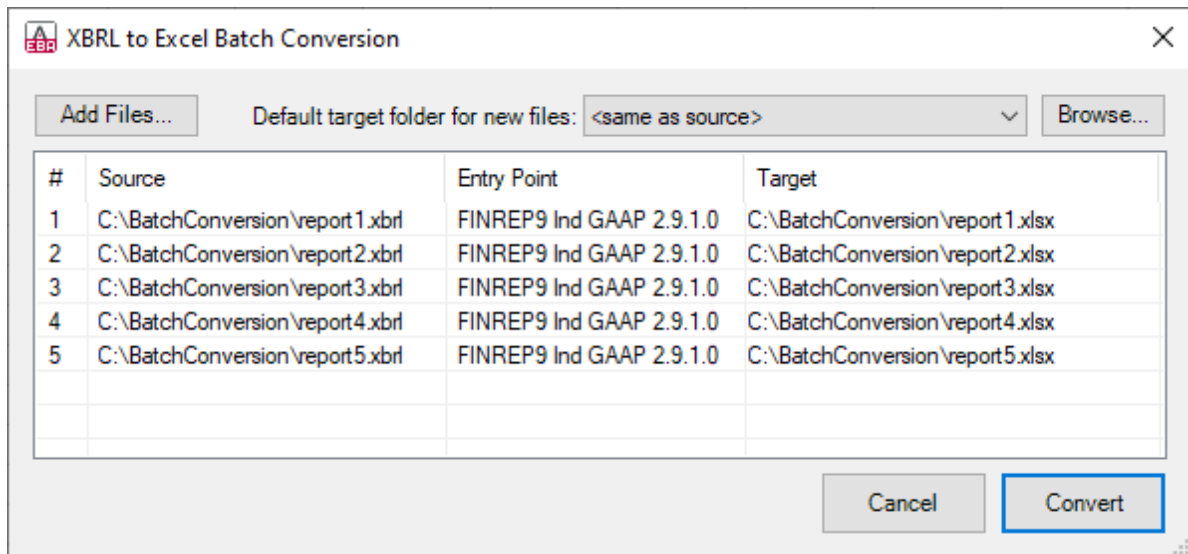
The **Batch Conversion** command in the Excel ribbon enables you to convert multiple XBRL instance files to Excel format. The result is the same as if you [imported](#)³² multiple XBRL instance files and then saved them to Excel format—while having the advantage that conversion takes place in batch.

In order to perform a batch conversion, you must first add all the required files to a batch, as follows:

1. In the Excel ribbon, click the **EBA** tab.



2. Click the **Batch Conversion** button.
3. Click **Add Files**. (Alternatively, right-click the grid, and select **Add Files** from the context menu.)

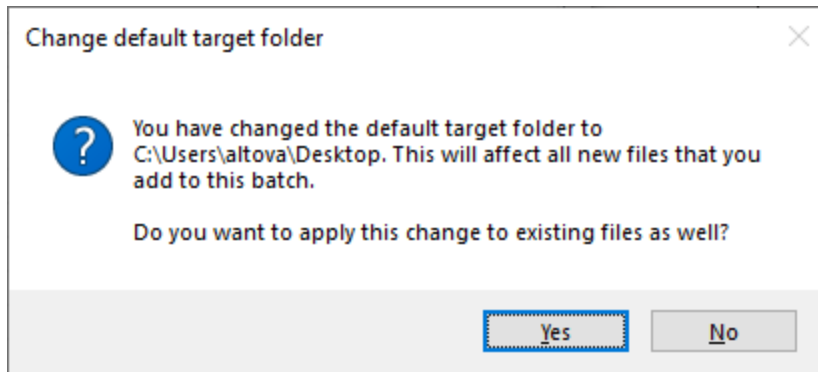


From the conversion dialog box, you can perform the following additional tasks:

1. To add files from multiple source folders to the same batch, click **Add Files** and select the desired source folder.
2. Whenever you add new files to the batch, their default target folder is the same as the source folder. If you want to assign a specific target folder to all new files by default, select it from the **Default target folder for new files** list. To add new entries to the list, click **Browse** and choose a folder.

Note: By default, the option **Default target folder for new files** affects new files that you add to the batch. However, if you change this option and files already exist in the batch (on the grid), a dialog box like

the one below appears. Click **Yes** if the target folder of existing files should be changed as well.



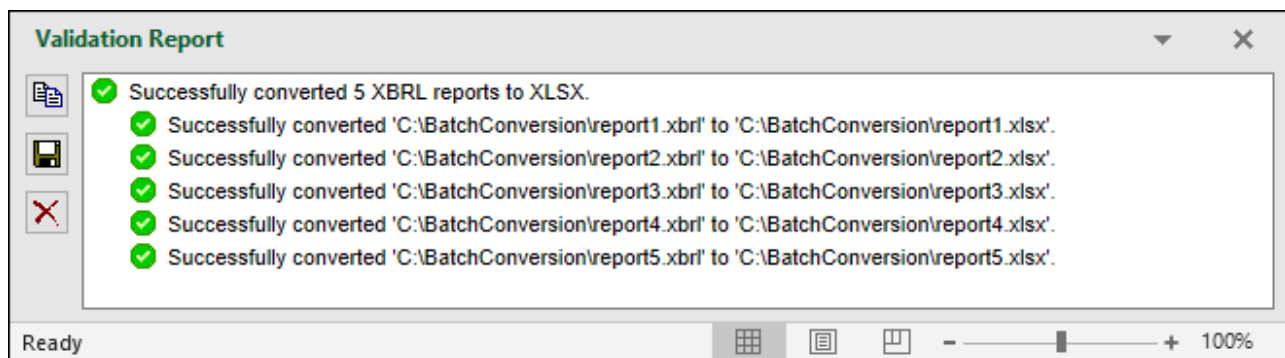
3. You can choose to save all converted files to the same target folder, or set a different target folder for each file. To change the target folder of specific files, first select the files on the grid, and then right-click the grid and select **Set target folder** from the context menu.
4. You can rename the target files. First select the files on the grid, and then right-click the grid and select **Rename** from the context menu (or press **F2**). You can change both the file name and the file path. However, if you change the path, make sure that it exists.
5. To remove files from the batch, first select them, and then right-click the grid and select **Remove** from the context menu (or press the **Del** key).

Tip: To select multiple files from the grid, the standard Windows key combinations apply, for example:

- While holding the **Ctrl** key pressed, click to select the files of interest.
- Click an empty area in the dialog box and then drag the cursor over the files to be selected (rectangular selection)
- Press **Ctrl+A** to select all files in the grid.

Once the batch is ready, click **Convert** to process all files in it. If any files with the same name already exist at the target folder, a message box appears asking your confirmation to overwrite them.

While a batch conversion is running, a dialog box appears that informs you about the progress. The outcome of the conversion operation is reported in the Validation Report window, for example:



When performing batch conversion, the same validation takes place as when importing a single XBRL instance file. If validation is successful, a message similar to "Successfully converted file.xlsx" is reported in the Validation Report for each file. If there are validation errors or warnings, they are reported in the Validation Report as well.

4 Command Reference

The add-in commands available in the **EBA** tab of the Excel ribbon are listed below.

Insert New Report

Creates a new EBA filing report, see [Creating a New Report](#)¹⁰. This command is disabled if the report sheet has already been inserted into the workbook.

Import from XBRL

Imports an XBRL instance file into the current Excel spreadsheet (see [Importing Data from XBRL](#)³²). This command is disabled if a report has already been inserted into the workbook. To enable the command, save and close the current report (workbook), and create a new workbook.

Export to XBRL

Exports data from all currently active sheets to an XBRL instance file (see [Exporting Data to XBRL](#)³¹).

Validate

Performs a validation of report data against the underlying XBRL taxonomy and displays the validation results in a dialog box (see [Validating Data](#)²⁸).

Batch Conversion

Converts multiple XBRL instance files to Excel, see [Batch Conversion fom XBRL to Excel](#)³⁴.

Toggle EBA Filing Pane

Toggles the **EBA Filing Pane** on or off. By default, this pane is visible.

Toggle Validation Report

Shows or hides the "Validation Report" window, see [Validating Data](#)²⁸.

Add Sheet (z-Axis)

Adds a new sheet which provides the ability to enter data in a third dimension. This button is enabled only if the table supports a third dimension according to XBRL taxonomy. For more information, see [Entering Data into Three-Dimensional Tables \(Z-Axis\)](#)²¹.

Remove Sheet (z-Axis)

Removes a previously added Z-axis sheet.

Add Row

Adds a new row to the currently selected table. This button is enabled only if the table supports growing vertically.

Delete Row

Deletes an existing row. This button is enabled only if the table supports growing vertically.

Add Column

Adds a new column to the currently selected table. This button is enabled only if the table supports growing horizontally.

Remove Column

Removes an existing column. This button is enabled only if the table supports growing horizontally.

Help

Opens this help file, in CHM (Microsoft Compiled HTML Help) format.

Add-In Activation

Displays the activation status of the add-in, or provides options to enter or purchase a license key code.

About EBA Add-In

Displays version information about the add-in.

Altova on the Web

Provides links to the Altova website (including Online Support Center, components download page, training and tutorials).

5 COM API

The add-in provides a COM API that can be used from programming languages that support interacting with Excel and accessing COM objects programmatically, such as VBA or .NET languages. Specifically, the API provides the means to create, import and export XBRL reports, and also to read and write form data.

If you intend to distribute the API to other clients, note the following minimum prerequisites:

1. Excel and **Altova® European Banking Authority (EBA) XBRL add-in for Excel** must be installed on each client machine.
2. Each API client that consumes your custom code or application must hold a valid **Altova® European Banking Authority (EBA) XBRL add-in for Excel** license.
3. There may be other prerequisites that depend on the platform that you used for your custom application.

5.1 Accessing the API

You can access programmatically the add-in's COM API in one of the following ways:

- from within Excel, by using Visual Basic for Applications (see also <https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office>)
- from your custom program, by using the Office Interop API from any .NET language.

The main interface is the `IAutomationAPI` interface. The following code listings illustrate how to create a new instance of the automation object in VBA.

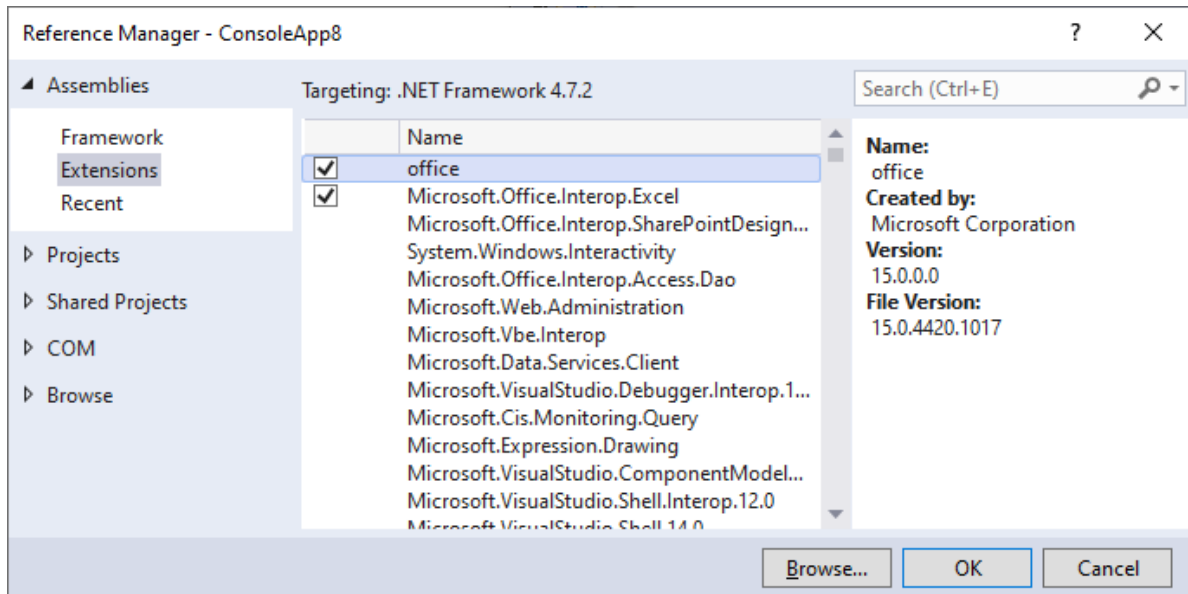
VBA

```
Dim automationObject As Object
Set automationObject = Application.COMAddIns.Item("Altova.EBAAddIn").Object
```

Accessing the COM API from a .NET project

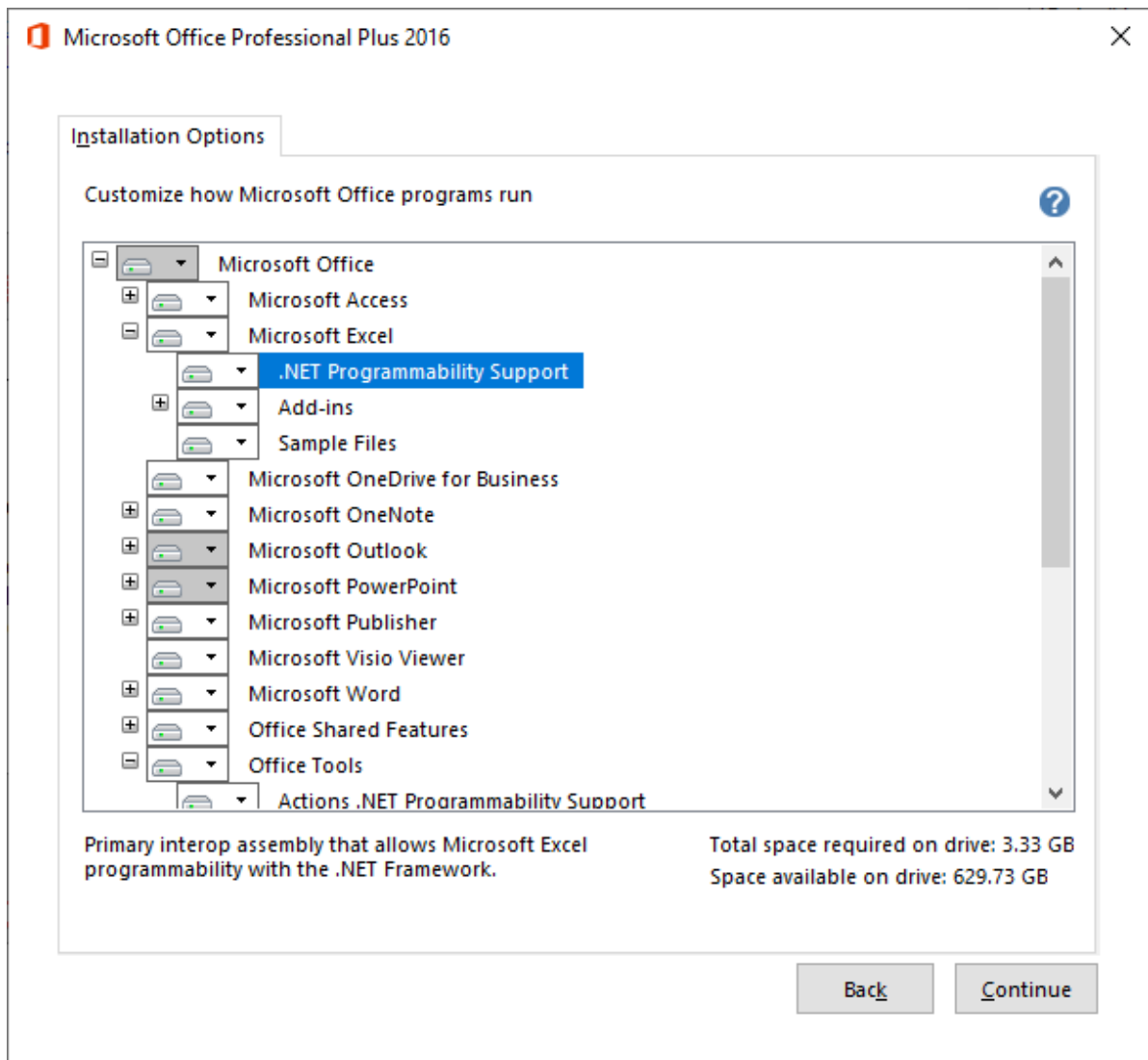
To access the COM API from a Visual Studio .NET project, add a reference to the **Microsoft Office Object Library (office.dll)** and **Microsoft.Office.Interop.Excel** assemblies:

1. In **Solution Explorer**, right-click your project's name and then click **Add Reference**. The **Add Reference** dialog box appears.
2. On the **Assemblies** page, select **office** and **Microsoft.Office.Interop.Excel** from the component list, and click **OK**.



If you do not see the assemblies above:

1. Make sure that you have installed Microsoft Office and that you have selected the **.NET Programmability Support** feature for Excel, for example:



2. Run the Visual Studio setup and make sure that you choose **Office/SharePoint development** workload (or the **Microsoft Office Developer Tools**, if applicable).

For more information about accessing Office interop assemblies from .NET projects, see <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interop/how-to-access-office-interop-objects>.

After adding the assembly references, you can create a new add-in instance as shown below.

C#

```
// Make sure that your project references the following two assemblies:
// * Microsoft.Office.Interop.Excel
// * Microsoft Office Object Library (office.dll)
var app = new Microsoft.Office.Interop.Excel.Application();
dynamic automationObject = app.COMAddIns.Item("Altova.EBAAddIn").Object;
```


5.2 C# Example

The following code listing illustrates how to convert an XBRL report to an Excel file, using the C# API.

```
// Converts an EBA XBRL report to an Excel document.
using Excel = Microsoft.Office.Interop.Excel;
namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var app = new Excel.Application();
            try
            {
                app.DisplayAlerts = false;

                var wb = (Excel._Workbook)(app.Workbooks.Add());

                // Get the Automation API object
                dynamic addIn = app.COMAddIns.Item("Altova.EBAAddIn");
                dynamic automationObject = addIn.Object;

                // Import EBA report eba_example.xbrl
                automationObject.ImportXBRL(@"C:\test\eba_example.xbrl");

                // Save as xlsx
                wb.SaveAs(@"C:\test\eba_example.xlsx");
                wb.Close();
            }
            finally
            {
                app.DisplayAlerts = true;
                app.Quit();
            }
        }
    }
}
```

5.3 VBA Example

The following code listing illustrates how to convert an XBRL report to an Excel file, using VBA.

```
' VBA Example 1:
' Creates a new EBA COREP ALM report with form 'C 68.00.a' and sets the value of the
' first cell
Sub Example1()
    Dim addIn As COMAddIn
    Dim automationObject As Object
    Dim Workbook As Object
    Dim tableTree As Object
    Dim tableNode As Object

    ' Get the Automation API object
    Set addIn = Application.COMAddIns.Item("Altova.EBAAddIn")
    Set automationObject = addIn.Object

    ' Insert a new EBA 2.9.1 COREP ALM report
    Set Workbook =
automationObject.InsertNewReport("http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/cir-
680-2014/2019-11-15/mod/corep_alm_con.xsd")
    Set tableTree = automationObject.GetTableTree(Workbook)

    ' Find table tree node for form 'C 68.00.a'
    Set tableNode = tableTree.FindTableByRCCode("C 68.00.a")

    ' Include this table in the filing (this will also create the respective Excel
worksheet)
    tableNode.IncludeInFiling = True

    ' Get the Data range of this form and set the value of the first cell to 42
    tableNode.Forms.Item(0).DataRange.Item(1).Value = "42"
End Sub
```

5.4 API Reference

This section provides reference to the objects of the **Altova® European Banking Authority (EBA) XBRL add-in for Excel** COM API. The objects are described in a generic manner, since the API may be used with virtually any language that supports calling a COM object.

5.4.1 Interfaces

5.4.1.1 IAutomationAPI

The `IAutomationAPI` interface is the main automation interface of the **Altova® European Banking Authority (EBA) XBRL add-in for Excel**. This interface is the starting point to do any further operations with the add-in or to retrieve or create other related automation objects. It allows you to create, import and export reports, and also to read and write form data. For information about creating an instance of this interface, see see [Accessing the API](#) ³⁹.

Methods

Name	Description
InsertNewReport ⁴⁴	Use this method to insert a new report of the respective taxonomy entry point.
ImportXBRL ⁴⁴	Imports an XBRL report. Returns the Excel workbook that contains the imported XBRL report.
ExportXBRL ⁴⁵	Exports the report in the respective Excel workbook to XBRL.
GetEntryPointTree ⁴⁵	Returns an <code>IEntryPointTree</code> object representing a tree of the available taxonomy entry points.
GetTableTree ⁴⁵	Returns an <code>ITableTree</code> object representing the tree of the available tables in the report opened in the specified Excel workbook.
GetReportProperties ⁴⁶	Returns an <code>IReportProperties</code> object providing the properties of the XBRL report.
GetFormProperties ⁴⁶	Returns an <code>IFormProperties</code> object providing the properties of the XBRL form in the specified Excel worksheet.
GetCellProperties ⁴⁶	Returns an <code>ICellProperties</code> object providing the properties of the fact in the specified Excel range.

5.4.1.1.1 Methods

5.4.1.1.1.1 *InsertNewReport*

Use this method to insert a new report of the respective taxonomy entry point.

Signature

```
InsertNewReport (in entryPointUrl:String) -> Microsoft.Office.Interop.Excel.Workbook
```

Parameters

Name	Type	Description
entryPointUrl	<i>String</i>	The URI of the taxonomy a report should be created for. Use the <code>GetEntryPointTree</code> method to get the available entry points.

5.4.1.1.1.2 *ImportXBRL*

Imports an XBRL report. Returns the Excel workbook that contains the imported XBRL report.

Signature

```
ImportXBRL (in inputFile:String) -> Microsoft.Office.Interop.Excel.Workbook
```

Parameters

Name	Type	Description
inputFile	<i>String</i>	The path to the XBRL report which should be imported.

5.4.1.1.1.3 *ExportXBRL*

Exports the report in the respective Excel workbook to XBRL.

Signature

```
ExportXBRL(in workbook:Microsoft.Office.Interop.Excel.Workbook, in outputFile:String) ->
Void
```

Parameters

Name	Type	Description
workbook	Microsoft.Office.Interop.Excel.Workbook	The Excel workbook to be exported.
outputFile	String	The path to the output XBRL file.

5.4.1.1.1.4 *GetEntryPointTree*

Returns an IEntryPointTree object representing a tree of the available taxonomy entry points.

Signature

```
GetEntryPointTree() -> IEntryPointTree
```

5.4.1.1.1.5 *GetTableTree*

Returns an ITableTree object representing the tree of the available tables in the report opened in the specified Excel workbook.

Signature

```
GetTableTree(in workbook:Microsoft.Office.Interop.Excel.Workbook) -> ITableTree
```

Parameters

Name	Type	Description
workbook	Microsoft.Office.Interop.Excel.Workbook	The Excel workbook containing the XBRL report.

5.4.1.1.1.6 *GetReportProperties*

Returns an `IReportProperties` object providing the properties of the XBRL report.

Signature

```
GetReportProperties(in workbook:Microsoft.Office.Interop.Excel.Workbook) ->
IReportProperties
```

Parameters

Name	Type	Description
workbook	<code>Microsoft.Office.Interop.Excel.Workbook</code>	The workbook containing the XBRL report.

5.4.1.1.1.7 *GetFormProperties*

Returns an `IFormProperties` object providing the properties of the XBRL form in the specified Excel worksheet.

Signature

```
GetFormProperties(in worksheet:Microsoft.Office.Interop.Excel.Worksheet) ->
IFormProperties
```

Parameters

Name	Type	Description
worksheet	<code>Microsoft.Office.Interop.Excel.Worksheet</code>	An Excel worksheet containing an XBRL form.

5.4.1.1.1.8 *GetCellProperties*

Returns an `ICellProperties` object providing the properties of the fact in the specified Excel range.

Signature

```
GetCellProperties(in cell:Microsoft.Office.Interop.Excel.Range) -> ICellProperties
```

Parameters

Name	Type	Description
cell	Microsoft.Office.Interop.Excel.Range	An Excel cell containing an XBRL fact (must be within the data range of a form).

5.4.1.2 IEntryPointTree

The `IEntryPointTree` interface provides information about the available taxonomy entry points in a structured way.

Properties

Name	Description
Groups ⁴⁷	Read-only. Returns a collection of <code>IEntryPointGroup</code> representing the available taxonomy groups, for example, "2.9" or "Bank of England Banking Taxonomy 3.1.1".

5.4.1.2.1 Properties

5.4.1.2.1.1 Groups

Returns a collection of `IEntryPointGroup` representing the available taxonomy groups, for example, "2.9" or "Bank of England Banking Taxonomy 3.1.1".

Signature

Groups : [Collection](#)

5.4.1.3 IEntryPointGroup

The `IEntryPointGroup` interface provides information about a group of taxonomy entry points.

Properties

Name	Description
Name ⁴⁸	Read-only. The Name of the group. For example, "EBA 2.9".

Name	Description
Country ⁴⁸	Read-only. The country code for which this entry point group is relevant. This is set only for country-specific taxonomies such as "Bank of England Banking Taxonomy".
Version ⁴⁹	Read-only. The version of the taxonomy. May be empty for sub-groups.
IsCurrentVersion ⁴⁹	Read-only. True if this group contains the current version of the taxonomy, false for older versions.
Groups ⁴⁹	Read-only. A collection of <code>IEntryPointGroup</code> representing the sub-groups of this entry point group.
EntryPoints ⁴⁹	Read-only. A collection of <code>IEntryPoint</code> representing the specific taxonomy entry points of this group.

5.4.1.3.1 Properties

5.4.1.3.1.1 Name

The Name of the group. For example, "EBA 2.9".

Signature

Name : `String`

5.4.1.3.1.2 Country

The country code for which this entry point group is relevant. This is set only for country-specific taxonomies such as "Bank of England Banking Taxonomy".

Signature

Country : `String`

5.4.1.3.1.3 *Version*

The version of the taxonomy. May be empty for sub-groups.

Signature

```
Version : String
```

5.4.1.3.1.4 *IsCurrentVersion*

True if this group contains the current version of the taxonomy, **false** for older versions.

Signature

```
IsCurrentVersion : Boolean
```

5.4.1.3.1.5 *Groups*

A collection of `IEntryPointGroup` representing the sub-groups of this entry point group.

Signature

```
Groups : Collection
```

5.4.1.3.1.6 *EntryPoints*

A collection of `IEntryPoint` representing the specific taxonomy entry points of this group.

Signature

```
EntryPoints : Collection
```

5.4.1.4 `IEntryPoint`

The `IEntryPoint` interface provides information about a specific taxonomy entry point.

Properties

Name	Description
Name ⁵⁰	Read-only.

Name	Description
	The name of the entry point. For example, "Additional Liquidity Monitoring - COREP, Consolidated"
ShortName ⁵⁰	Read-only. The abbreviated form of the entry points name. For example, "COREP ALM Con".
Version ⁵¹	Read-only. The version of the entry point. This may be empty (use the parents group version in this case).
URI ⁵¹	Read-only. The URI of the taxonomy entry point, for example, "http://www.eba.europa.eu/eu/fr/xbml/crr/fws/corep/cir-680-2014/2019-04-30/mod/corep_alm_con.xsd". Pass this to the <code>IAutomationAPI.InsertNewReport</code> method.
Needs64Bit ⁵¹	Read-only. True if the entry point requires the 64-bit version of Excel, false otherwise.

5.4.1.4.1 Properties

5.4.1.4.1.1 Name

The name of the entry point. For example, "Additional Liquidity Monitoring - COREP, Consolidated"

Signature

Name : `String`

5.4.1.4.1.2 ShortName

The abbreviated form of the entry points name. For example, "COREP ALM Con".

Signature

ShortName : `String`

5.4.1.4.1.3 Version

The version of the entry point. This may be empty (use the parents group version in this case).

Signature

Version : **String**

5.4.1.4.1.4 URI

The URI of the taxonomy entry point, for example, "http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/cir-680-2014/2019-04-30/mod/corep_alm_con.xsd". Pass this to the `IAutomationAPI.InsertNewReport` method.

Signature

URI : **String**

5.4.1.4.1.5 Needs64Bit

True if the entry point requires the 64-bit version of Excel, false otherwise.

Signature

Needs64Bit : **Boolean**

5.4.1.5 ITableTree

The `ITableTree` interface provides structured information about the available tables and forms in an XBRL report.

Properties

Name	Description
Nodes ⁵²	Read-only. Returns a collection of <code>IGroupNode</code> and <code>ITableNode</code> objects, which represent groups of tables and tables, respectively.

Methods

Name	Description
FindTableByRCCode ⁵²	Returns the table node with the specified RC Code.

5.4.1.5.1 Properties

5.4.1.5.1.1 Nodes

Returns a collection of `IGroupNode` and `ITableNode` objects, which represent groups of tables and tables, respectively.

Signature

```
Nodes : Collection
```

5.4.1.5.2 Methods

5.4.1.5.2.1 FindTableByRCCode

Returns the table node with the specified RC Code.

Signature

```
FindTableByRCCode(in rcCode:string) -> ITableNode
```

Parameters

Name	Type	Description
rcCode	<code>string</code>	The RC Code of the desired table node.

5.4.1.6 IGroupNode

The `IGroupNode` interface provides information about a group of tables.

Properties

Name	Description
Text ⁵³	Read-only. The name of the group of tables as displayed in the EBA Filing Pane.
IsGroup ⁵³	Read-only.

Name	Description
	This is always true for group nodes. Use this to distinguish between <code>IGroupNode</code> and <code>ITableNode</code> members of the <code>Nodes</code> collection.
IsTable ⁵³	Read-only. This is always false for group nodes. Use this to distinguish between <code>IGroupNode</code> and <code>ITableNode</code> members of the <code>Nodes</code> collection.
Nodes ⁵⁴	Read-only. Returns a collection of <code>IGroupNode</code> and <code>ITableNode</code> objects, which represent groups of tables and tables, respectively.

5.4.1.6.1 Properties

5.4.1.6.1.1 Text

The name of the group of tables as displayed in the EBA Filing Pane.

Signature

Text : `String`

5.4.1.6.1.2 IsGroup

This is always **true** for group nodes. Use this to distinguish between `IGroupNode` and `ITableNode` members of the `Nodes` collection.

Signature

IsGroup : `Boolean`

5.4.1.6.1.3 IsTable

This is always **false** for group nodes. Use this to distinguish between `IGroupNode` and `ITableNode` members of the `Nodes` collection.

Signature

IsTable : `Boolean`

5.4.1.6.1.4 Nodes

Returns a collection of `IGroupNode` and `ITableNode` objects, which represent groups of tables and tables, respectively.

Signature

Nodes : [Collection](#)

5.4.1.7 ITableNode

The `ITableNode` interface provides information about a table.

Properties

Name	Description
Text ⁵⁶	Read-only. The name of the table as displayed in the EBA Filing Pane.
IsGroup ⁵⁶	Read-only. This is always false for table nodes. Use this to distinguish between <code>IGroupNode</code> and <code>ITableNode</code> members of the <code>Nodes</code> collection.
IsTable ⁵⁶	Read-only. This is always true for table nodes. Use this to distinguish between <code>IGroupNode</code> and <code>ITableNode</code> members of the <code>Nodes</code> collection.
Forms ⁵⁶	Read-only. Returns a collection of <code>IForm</code> objects, which represent concrete forms that can be displayed as Excel worksheet.
CanAddSubForm ⁵⁶	Read-only. This is true if additional sub forms can be added to the table, for example, if the table has open aspects on the z-Axis. In most cases, this is a form for each country or currency.
FilingIndicator ⁵⁷	Read-only. The filing indicator code of the table.
RCCode ⁵⁷	Read-only. The RC code of the table.
IncludeInFiling ⁵⁷	True if the table should be part of the report, false otherwise. If you set this property to true for the first time, a new worksheet for this table will be created.

Methods

Name	Description
AddSubForm ⁵⁷	Creates a new sub-form of this table and returns the respective <code>IForm</code> object. This method returns null if no sub-form can be added.

5.4.1.7.1 Properties

5.4.1.7.1.1 *Text*

The name of the table as displayed in the EBA Filing Pane.

Signature

```
Text : String
```

5.4.1.7.1.2 *IsGroup*

This is always **false** for table nodes. Use this to distinguish between `IGroupNode` and `ITableNode` members of the `Nodes` collection.

Signature

```
IsGroup : Boolean
```

5.4.1.7.1.3 *IsTable*

This is always **true** for table nodes. Use this to distinguish between `IGroupNode` and `ITableNode` members of the `Nodes` collection.

Signature

```
IsTable : Boolean
```

5.4.1.7.1.4 *Forms*

Returns a collection of `IForm` objects, which represent concrete forms that can be displayed as Excel worksheet.

Signature

```
Forms : Collection
```

5.4.1.7.1.5 *CanAddSubForm*

This is **true** if additional sub forms can be added to the table, for example, if the table has open aspects on the z-Axis. In most cases, this is a form for each country or currency.

Signature

```
CanAddSubForm : Boolean
```

5.4.1.7.1.6 *FilingIndicator*

The filing indicator code of the table.

Signature

```
FilingIndicator : String
```

5.4.1.7.1.7 *RCCode*

The RC code of the table.

Signature

```
RCCode : String
```

5.4.1.7.1.8 *IncludeInFiling*

True if the table should be part of the report, **false** otherwise. If you set this property to **true** for the first time, a new worksheet for this table will be created.

Signature

```
IncludeInFiling : Boolean
```

5.4.1.7.2 Methods

5.4.1.7.2.1 *AddSubForm*

Creates a new sub-form of this table and returns the respective `IForm` object. This method returns null if no sub-form can be added.

Signature

```
AddSubForm() -> IForm
```

5.4.1.8 IForm

The `IForm` interface provides information about a form.

Properties

Name	Description
Text ⁵⁸	Read-only. The name of the form as displayed in the EBA Filing Pane.
DataRange ⁵⁹	Read-only. The Excel range containing the data of this form.
FormSelectorRange ⁵⁹	Read-only. The Excel range containing the form selector of this form. Namely, the cells that contain the data that distinguishes this form from the other forms of the same table. This returns null if the table may not consist of multiple forms.
Worksheet ⁵⁹	Read-only. The Excel worksheet containing this form. This may be null if <code>IncludeInFiling</code> is false.
IncludeInFiling ⁵⁹	True if this form should be part of the report, false otherwise. This shows/hides the respective Excel worksheet.

Methods

Name	Description
Remove ⁶⁰	Removes this form and deletes the respective Excel worksheet.

5.4.1.8.1 Properties

5.4.1.8.1.1 Text

The name of the form as displayed in the EBA Filing Pane.

Signature

Text : `String`

5.4.1.8.1.2 *DataRange*

The Excel range containing the data of this form.

Signature

```
DataRange : Microsoft.Office.Interop.Excel.Range
```

5.4.1.8.1.3 *FormSelectorRange*

The Excel range containing the form selector of this form. Namely, the cells that contain the data that distinguishes this form from the other forms of the same table. This returns null if the table may not consist of multiple forms.

Signature

```
FormSelectorRange : Microsoft.Office.Interop.Excel.Range
```

5.4.1.8.1.4 *Worksheet*

The Excel worksheet containing this form. This may be null if `IncludeInFiling` is false.

Signature

```
Worksheet : Microsoft.Office.Interop.Excel.Worksheet
```

5.4.1.8.1.5 *IncludeInFiling*

True if this form should be part of the report, **false** otherwise. This shows/hides the respective Excel worksheet.

Signature

```
IncludeInFiling : Boolean
```

5.4.1.8.2 Methods

5.4.1.8.2.1 Remove

Removes this form and deletes the respective Excel worksheet.

Signature

```
Remove() -> Void
```

5.4.1.9 IReportProperties

The `IReportProperties` interface provides properties of the whole XBRL report.

Properties

Name	Description
EntryPointURI ⁶¹	Read-only. The URI of the taxonomy entry point. For example, "Additional Liquidity Monitoring - COREP, Consolidated".
EntryPointModuleName ⁶¹	Read-only. The module name of the taxonomy entry point. For example, "http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/cir-680-2014/2019-04-30/mod/corep_alm_con.xsd".
EntryPointModuleDPMID ⁶¹	Read-only. The Data Point Model Database ID of the module. For example, "218".
ReportingEntityScheme ⁶¹	The scheme of the reporting entity. For example, "http://standards.iso.org/iso/17442".
ReportingEntityIdentifier ⁶²	The identifier of the reporting entity.
ReferenceDate ⁶²	The reference date of the report.
MonetaryCellsAccuracy ⁶²	The accuracy of monetary facts in this report. Applies to each monetary fact for which no separate accuracy was specified (at cell or table level).
PercentageCellsAccuracy ⁶²	The accuracy of percentage facts in this report. Applies to each percentage fact for which no separate accuracy was specified (at cell or table level).
PureCellsAccuracy ⁶³	The accuracy of pure facts in this report. Applies to each pure fact for which no separate accuracy was specified (at cell or

Name	Description
	table level).
ReportingCurrency ⁶³	The reporting currency used in the XBRL report as an ISO 4217 currency code.
ReportingLanguage ⁶³	The language of footnotes in the XBRL report as BCP-47 language tag. For example, "en-US".

5.4.1.9.1 Properties

5.4.1.9.1.1 *EntryPointURI*

The URI of the taxonomy entry point. For example, "Additional Liquidity Monitoring - COREP, Consolidated".

Signature

```
EntryPointURI : String
```

5.4.1.9.1.2 *EntryPointModuleName*

The module name of the taxonomy entry point. For example, "http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/cir-680-2014/2019-04-30/mod/corep_alm_con.xsd".

Signature

```
EntryPointModuleName : String
```

5.4.1.9.1.3 *EntryPointModuleDPMID*

The Data Point Model Database ID of the module. For example, "218".

Signature

```
EntryPointModuleDPMID : String
```

5.4.1.9.1.4 *ReportingEntityScheme*

The scheme of the reporting entity. For example, "http://standards.iso.org/iso/17442".

Signature

```
ReportingEntityScheme : String
```

5.4.1.9.1.5 *ReportingEntityIdentifier*

The identifier of the reporting entity.

Signature

```
ReportingEntityIdentifier : String
```

5.4.1.9.1.6 *ReferenceDate*

The reference date of the report.

Signature

```
ReferenceDate : DateTime
```

5.4.1.9.1.7 *MonetaryCellsAccuracy*

The accuracy of monetary facts in this report. Applies to each monetary fact for which no separate accuracy was specified (at cell or table level).

Signature

```
MonetaryCellsAccuracy : String
```

5.4.1.9.1.8 *PercentageCellsAccuracy*

The accuracy of percentage facts in this report. Applies to each percentage fact for which no separate accuracy was specified (at cell or table level).

Signature

```
PercentageCellsAccuracy : String
```

5.4.1.9.1.9 *PureCellsAccuracy*

The accuracy of pure facts in this report. Applies to each pure fact for which no separate accuracy was specified (at cell or table level).

Signature

```
PureCellsAccuracy : String
```

5.4.1.9.1.10 *ReportingCurrency*

The reporting currency used in the XBRL report as an ISO 4217 currency code.

Signature

```
ReportingCurrency : String
```

5.4.1.9.1.11 *ReportingLanguage*

The language of footnotes in the XBRL report as BCP-47 language tag. For example, "en-US".

Signature

```
ReportingLanguage : String
```

5.4.1.10 IFormProperties

The `IFormProperties` interface provides properties of one form of the report.

Properties

Name	Description
MonetaryCellsAccuracy ⁶⁴	The accuracy of monetary facts in this form. Applies to each monetary fact for which no separate accuracy was specified.
PercentageCellsAccuracy ⁶⁴	The accuracy of percentage facts in this form. Applies to each percentage fact for which no separate accuracy was specified.
PureCellsAccuracy ⁶⁴	The accuracy of pure facts in this form. Applies to each pure fact for which no separate accuracy was specified.
TableRCCode ⁶⁵	Read-only. The RC Code of the table. For example, "C 66.01.a".

Name	Description
FilingIndicatorCode ⁶⁵	Read-only. The filing indicator code of the table. For example, "C_66.01".
Label ⁶⁵	Read-only. The label of the table. For example, "C 66.01.a".
VerboseLabel ⁶⁵	Read-only. The verbose label of the table. For example, "C 66.01.a Maturity ladder. Total. Overnight and higher maturity".
TableID ⁶⁶	Read-only. The id of the table resource. For example, "eba_tC_66.01.a".
TableDPMID ⁶⁶	Read-only. The Data Point Model Database ID of the table. For example, "429.1431".
ValidationRules ⁶⁶	Read-only. The collection of validation rules that apply to this table.

5.4.1.10.1 Properties

5.4.1.10.1.1 *MonetaryCellsAccuracy*

The accuracy of monetary facts in this form. Applies to each monetary fact for which no separate accuracy was specified.

Signature

```
MonetaryCellsAccuracy : String
```

5.4.1.10.1.2 *PercentageCellsAccuracy*

The accuracy of percentage facts in this form. Applies to each percentage fact for which no separate accuracy was specified.

Signature

```
PercentageCellsAccuracy : String
```

5.4.1.10.1.3 *PureCellsAccuracy*

The accuracy of pure facts in this form. Applies to each pure fact for which no separate accuracy was specified.

Signature

```
PureCellsAccuracy : String
```

5.4.1.10.1.4 *TableRCCode*

The RC Code of the table. For example, "C 66.01.a".

Signature

```
TableRCCode : String
```

5.4.1.10.1.5 *FilingIndicatorCode*

The filing indicator code of the table. For example, "C_66.01".

Signature

```
FilingIndicatorCode : String
```

5.4.1.10.1.6 *Label*

The label of the table. For example, "C 66.01.a".

Signature

```
Label : String
```

5.4.1.10.1.7 *VerboseLabel*

The verbose label of the table. For example, "C 66.01.a Maturity ladder. Total. Overnight and higher maturity".

Signature

```
VerboseLabel : String
```

5.4.1.10.1.8 *TableID*

The id of the table resource. For example, "eba_tC_66.01.a".

Signature

```
TableID : String
```

5.4.1.10.1.9 *TableDPMID*

The Data Point Model Database ID of the table. For example, "429.1431".

Signature

```
TableDPMID : String
```

5.4.1.10.1.10 *ValidationRules*

The collection of validation rules that apply to this table.

Signature

```
ValidationRules : Collection
```

5.4.1.11 *ICellProperties*

The *ICellProperties* interface provides properties of one fact of the report.

Properties

Name	Description
Accuracy ⁶⁷	The accuracy of the numeric item as string. This may be "INF" or a number representing the number of decimal places to which this fact is accurate. This is null for non-numeric items.
Footnote ⁶⁷	The footnote of the fact.
Name ⁶⁷	Read-only. The concept name of the fact. For example, "eba_met:mi256".

Name	Description
Type ⁶⁸	Read-only. The type of the fact. For example, "xbrli:monetaryItemType".
Label ⁶⁸	Read-only. The label of the fact. For example, "Cash value".
DPMID ⁶⁸	Read-only. The Data Point Model Database ID of this fact. For example, "6062".
Dimensions ⁶⁸	Read-only. The collection of <code>IDimension</code> objects representing the dimensions for which this fact is reported.

5.4.1.11.1 Properties

5.4.1.11.1.1 Accuracy

The accuracy of the numeric item as string. This may be "INF" or a number representing the number of decimal places to which this fact is accurate.

This is null for non-numeric items.

Signature

Accuracy : `String`

5.4.1.11.1.2 Footnote

The footnote of the fact.

Signature

Footnote : `String`

5.4.1.11.1.3 Name

The concept name of the fact. For example, "eba_met:mi256".

Signature

```
Name : String
```

5.4.1.11.1.4 *Type*

The type of the fact. For example, "xbri:monetaryItemType".

Signature

```
Type : String
```

5.4.1.11.1.5 *Label*

The label of the fact. For example, "Cash value".

Signature

```
Label : String
```

5.4.1.11.1.6 *DPMID*

The Data Point Model Database ID of this fact. For example, "6062".

Signature

```
DPMID : String
```

5.4.1.11.1.7 *Dimensions*

The collection of `IDimension` objects representing the dimensions for which this fact is reported.

Signature

```
Dimensions : Collection
```

5.4.1.12 IDimension

The `IDimension` interface provides basic information of a `Dimension` and its value.

Properties

Name	Description
Name ⁶⁹	Read-only. The name of the dimension.
Value ⁶⁹	Read-only. The value of the dimension as <code>String</code> .

5.4.1.12.1 Properties

5.4.1.12.1.1 Name

The name of the dimension.

Signature

Name : `String`

5.4.1.12.1.2 Value

The value of the dimension as `String`.

Signature

Value : `String`

6 License Information

This section contains information about:

- the distribution of this software product
- software activation and license metering
- the license agreement governing the use of this product

Please read this information carefully. It is binding upon you since you agreed to these terms when you installed this software product.

To view the terms of any Altova license, go to the [Altova Legal Information page](#) at the [Altova website](#).

6.1 Electronic Software Distribution

This product is available through electronic software distribution, a distribution method that provides the following unique benefits:

- You can evaluate the software free-of-charge for 30 days before making a purchasing decision. (*Note: Altova MobileTogether Designer is licensed free of charge.*)
- Once you decide to buy the software, you can place your order online at the [Altova website](#) and get a fully licensed product within minutes.
- When you place an online order, you always get the latest version of our software.
- The product package includes an onscreen help system that can be accessed from within the application interface. The latest version of the user manual is available at www.altova.com in (i) HTML format for online browsing, and (ii) PDF format for download (and to print if you prefer to have the documentation on paper).

30-day evaluation period

After downloading this product, you can evaluate it for a period of up to 30 days free of charge. About 20 days into the evaluation period, the software will start to remind you that it has not yet been licensed. The reminder message will be displayed once each time you start the application. If you would like to continue using the program after the 30-day evaluation period, you must purchase a product license, which is delivered in the form of a license file containing a key code. Unlock the product by entering this key code in the Software Activation dialog of your product.

You can purchase product licenses at the [Altova online shop](#).

Helping Others within Your Organization to Evaluate the Software

If you wish to distribute the evaluation version within your company network, or if you plan to use it on a PC that is not connected to the Internet, you may distribute only the installer file, provided that this file is not modified in any way. Any person who accesses the software installer that you have provided must request their own 30-day evaluation license key code and after expiration of their evaluation period, must also purchase a license in order to be able to continue using the product.

6.2 Software Activation and License Metering

As part of Altova's Software Activation, the software may use your internal network and Internet connection for the purpose of transmitting license-related data at the time of installation, registration, use, or update to an Altova-operated license server and validating the authenticity of the license-related data in order to protect Altova against unlicensed or illegal use of the software and to improve customer service. Activation is based on the exchange of license related data such as operating system, IP address, date/time, software version, and computer name, along with other information between your computer and an Altova license server.

Your Altova product has a built-in license metering module that further helps you avoid any unintentional violation of the End User License Agreement. Your product is licensed either as a single-user or multi-user installation, and the license-metering module makes sure that no more than the licensed number of users use the application concurrently.

This license-metering technology uses your local area network (LAN) to communicate between instances of the application running on different computers.

Single license

When the application starts up, as part of the license metering process, the software sends a short broadcast datagram to find any other instance of the product running on another computer in the same network segment. If it doesn't get any response, it will open a port for listening to other instances of the application.

Multi-user license

If more than one instance of the application is used within the same LAN, these instances will briefly communicate with each other on startup. These instances exchange key-codes in order to help you to better determine that the number of concurrent licenses purchased is not accidentally violated. This is the same kind of license metering technology that is common in the Unix world and with a number of database development tools. It allows Altova customers to purchase reasonably-priced concurrent-use multi-user licenses.

We have also designed the applications so that they send few and small network packets so as to not put a burden on your network. The TCP/IP ports (2799) used by your Altova product are officially registered with the IANA (see the [IANA Service Name Registry](#) for details) and our license-metering module is tested and proven technology.

If you are using a firewall, you may notice communications on port 2799 between the computers that are running Altova products. You are, of course, free to block such traffic between different groups in your organization, as long as you can ensure by other means, that your license agreement is not violated.

If you are online, you will also notice that your Altova software provides many useful functions. These are unrelated to the license-metering technology.

Note about certificates

Your Altova application contacts the Altova licensing server (link.altova.com) via HTTPS. For this communication, Altova uses a registered SSL certificate. If this certificate is replaced (for example, by your IT department or an external agency), then your Altova application will warn you about the connection being insecure. You could use the replacement certificate to start your Altova application, but you would be doing this at your own risk. If you see a *Non-secure connection* warning message, check the origin of the certificate and

consult your IT team (who would be able to decide whether the interception and replacement of the Altova certificate should continue or not).

If your organization needs to use its own certificate (for example, to monitor communication to and from client machines), then we recommend that you install Altova's free license management software, [Altova LicenseServer](#), on your network. Under this setup, client machines can continue to use your organization's certificates, while Altova LicenseServer can be allowed to use the Altova certificate for communication with Altova.

6.3 Altova XBRL Add-in Software License Agreement

- The Altova XBRL Add-in Software License Agreement is available here: <https://www.altova.com/legal/xbrl-add-in-eula>
- Altova's Privacy Policy is available here: <https://www.altova.com/privacy>

Index

6

64-bit Excel,
using the add-in on, 7

A

Accuracy,
as property in Document Actions pane, 25

ACPR Taxonomy,
installing, 9

Altova® European Banking Authority (EBA) XBRL add-in for Excel,
about, 5
command reference, 36
installation, 7
licensing, 7
limitations, 5
system requirements, 5
viewing the current version, 7

B

BDP Taxonomy,
installing, 9

BOE Taxonomy,
installing, 9

C

C#,
API, 38, 41

COM API,
accessing the, 39
examples, 41, 42
using, 38

Copyright information, 70

D

Distribution,
of Altova's software products, 70, 71

DNB FTK Taxonomy,
installing, 9

E

EBA data,
exporting to XBRL instance, 31
importing from XBRL instance, 32
validating, 28
viewing cell formatting, 31

End User License Agreement, 70

Evaluation period,
of Altova's software products, 70, 71

Excel .xltx template,
opening, 10

Export,
data to XBRL, 31

I

Import,
XBRL instance into Excel, 32

Installation, 7

L

Legal information, 70

License,
information about, 70

License metering,
in Altova products, 72

R

Report data,

Report data,

entering, 16

pasting, 16

S**SRB Taxonomy,**

installing, 9

System requirements, 5**V****Validation, 28****VBA,**

API, 38, 42