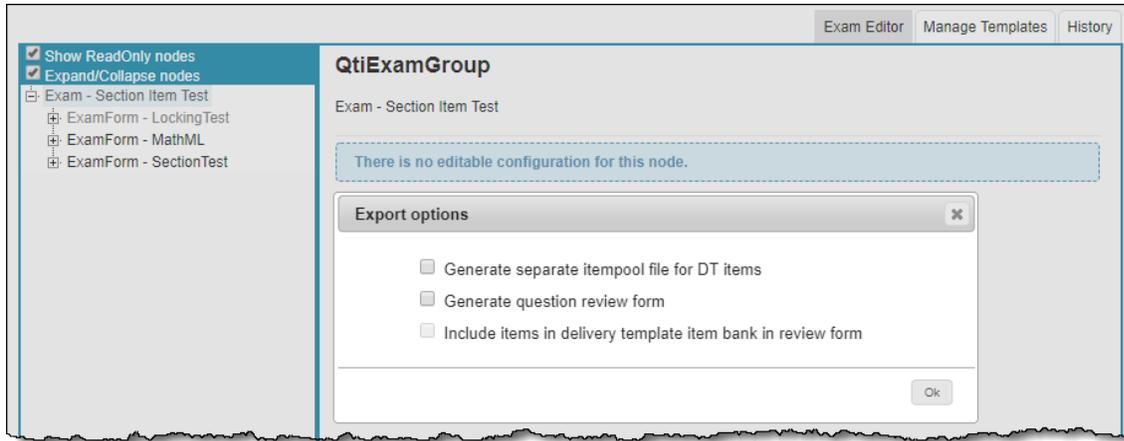


4.1.11.1 Export QTI Package

Once all the required modifications have been made and saved to the exam form configuration, you are ready to export the QTI Package file to be played in the **Pearson VUE Player** for the candidate to take the exam. To generate this ZIP archive package file, click the **Export** button at the bottom. If you have made changes, you must click **Save** before clicking **Export**.

An **Export options** pop-up appears where you can decide how to generate the Question Review Form, which is a file named "*Review.xml*", that is also generated along with the other files in the export package. This file contains information about the list of items from the exam form.



The following selections are available in the **Export options** pop-up:

- **Generate separate itempool file for DT items:** The items that are part of the Delivery Template (DT) itembank are exported in a separate Itempool file. When this option is selected, the **ItemPool-DT.xml** file only contains items that belong to the DT itembank and the **ItemPool.xml** file only contains the items that belong to the standard itembank. If this option is not selected, the ItemPool.xml file contains all standard itembank and DT itembank items.
- **Generate question review form:** The Question Review Form **Review.xml** is generated containing all the items from the standard Item Bank Project that are on the exam form.
- **Include items in delivery template item bank in review form:** The Question Review Form **Review.xml** is generated containing all the items from the standard Item Bank Project that are on the exam form and also all the items that may be from the Delivery Template Item Bank via template applied to the exam form.



Items in the Question Review Form are listed in the ascending Question ID order, regardless of whether the item is from the standard Item Bank Project or from the Delivery Template Item Bank via template applied to the exam form. This is true even when the Question Review Form is divided into multiple files to include specific number of items per file, with the first file containing the start of the list of items.

The export begins when you click the **Ok** button. A progress indicator circular animation is shown while the QTI Export is generating.

This may not be visible for smaller exams because the QTI Export may happen very quickly, but it provides a good measure of the progress of the process for large exam forms.



The ZIP file contains all the information about the examination along with the exam forms, items, and assets. Shown below is an example of the contents of the ZIP archive.

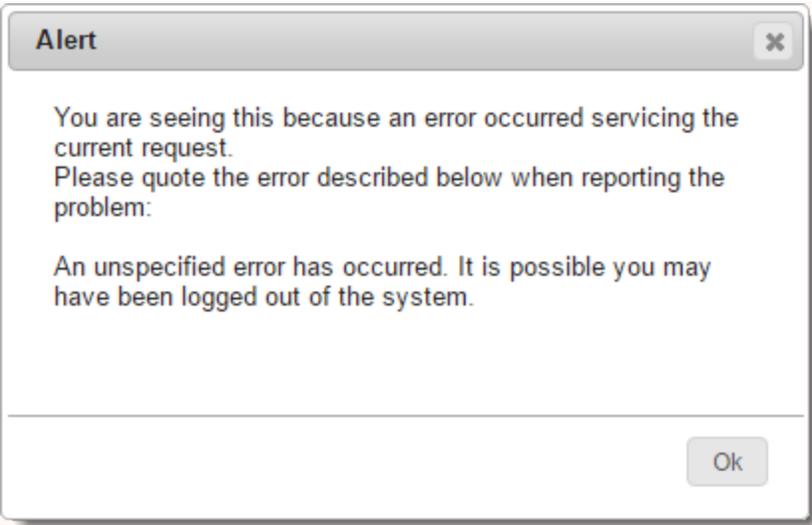
Name	Date modified	Type	Size
Assessment.xml	3/11/2019 2:27 PM	XML Document	1 KB
ItemPool.xml	3/11/2019 2:27 PM	XML Document	36 KB
manifest.xml	3/11/2019 2:27 PM	XML Document	1 KB
MCQ_Only.cpf	3/11/2019 2:27 PM	CPF File	1 KB
MyImage1.jpg	3/11/2019 2:27 PM	JPG File	52 KB
MyImage2.jpg	3/11/2019 2:27 PM	JPG File	52 KB
QAReview.xml	3/11/2019 2:27 PM	XML Document	2 KB
ReportingGroup.xml	3/11/2019 2:27 PM	XML Document	21 KB
Section.xml	3/11/2019 2:27 PM	XML Document	2 KB

Here is a brief explanation of the files that may be present within the zip file:

1. **Assessment.xml**: represents all the ExamForm nodes and the list of all sections in all the Exams.
2. **ItemPool.xml**: XML file containing information related to all the Questions in the standard itembank. If "Generate separate itempool file for DT items" is not selected, it also contains the items from the DT itembank.
3. **ItemPool-DT.xml**: XML file containing only the items from the DT itembank. This file is only present if "Generate separate itempool file for DT items" is selected.
4. **Manifest.xml**: Manifest file. Identifies exam Language, and all the contents of the exam.
5. **QAReview.xml**: Group all items on one exam form for review purposes.
6. **<Exam Name>.cpf**: Pearson Compiler Project file, which can run in the Pearson VUE Player and play the examination.
7. **Section.xml**: represents all the Sections the user created and added to all the Exams, and all items included in the sections.
8. **Image files** (if any): Assets.
9. **ReportingGroup.xml**: This file is generated when Reporting Groups are added to an assessment and items are present in the Reporting Group.
10. **ResourceMap.zip**: This file is generated when a Resource Container is added to an assessment. The ResourceMap.xml file used to define various display settings for assets such as the scroll position and split screen state in the Pearson VUE Player.
11. **Review.xml**: This file contains the items from the Item Bank Project and from the Delivery Template Item Bank if selected.



If you encounter this error while compiling your QTI, please consult with your application consultant.



**Naming conventions followed by QTI Package files for idents with \$ and #
For Section nodes:**

When an exam form is exported to a QTI Package from the **Delivery Configuration Tool** page, all the Section nodes are combined into a single file called **Section.xml**.

For ExamForm nodes:

All ExamForm nodes (Assessment nodes) are combined into a single file called **Assessment.xml**.

Under PearsonvuePrecondition (under ExamForm/Section nodes)

The \$ and # replacement also happens on the following nodes LinkRefID attributes:

Variable Comparison (VARIABLE_COMPARISON)

Response Comparison (RESPONSE_COMPARISON)

Item State (ITEM_STATE)

The behavior is also seen for **PearsonvueApplication - Calculator**.



When an "&" (ampersand) symbol is used in a Question Stem or Answer Option, it is first carried to the "Itempool.xml" file.

Therefore, an item with a Stem having an "&" symbol as below,

Examine Questions Project: Huss's Training (New BP) [Help](#)

Question Status: Review: Default

[Return to Author](#) [Modify](#) [Clone](#) [Preview](#) [Print](#) [< Previous](#) [Next >](#)
Jump to 1 of 1
1 (157053)

Question Summary

Type: MCQ, One Correct Option State: Review: Default Weight: 1

Stem:

Romeo & Juliet come from what two families?

Options:

- *A. Montague and Capulet
- B. Monterey and Capitan

will have a Stem entry in the "Itempool.xml" file as below,

```

<flow>
  <flow>
    <material>
      <mattext><>Romeo &amp; Juliet come from what two families?</p>
    </material>
  </flow>
</flow>

```

where the "&" is replaced by "&", which is correct per HTML notation.

However, if the "Itempool.xml" is manually edited to change this entry to "&", which is not its correct HTML notation, then it results in a compilation error in the Pearson VUE QTI Compiler and the compilation fails.



When any font attribute is modified for a MathML expression in the MathML window of any item, an error message similar to the one shown below is generated in the Pearson VUE QTI Compiler and the compilation fails.

```

2017-05-17 13.18.44.073 : ERROR : Attribute "mathsize" must be declared for element type "m:mrow". [Current tag: m:math line: 9 column: 119]
2017-05-17 13.18.44.073 : ERROR : Attribute "mathvariant" must be declared for element type "m:mrow". [Current tag: m:math line: 9 column: 119]
2017-05-17 13.18.44.074 : ERROR : Attribute "mathbackground" must be declared for element type "m:mrow". [Current tag: m:math line: 9 column: 119]
2017-05-17 13.18.44.074 : ERROR : Attribute "mathcolor" must be declared for element type "m:mrow". [Current tag: m:math line: 9 column: 119]

```

This can be fixed by editing the "Itempool.xml" with a text editor such as Notepad. Locate the affected code at the mentioned line number(s) and move the font attributes block (including mathsize, mathvariant, mathbackground, and mathcolor) from the <m:mrow> tag, into the <m:mtext> tag as shown in the example below

```

Before:
<material>
  <mattext><p><m:math><m:mrow mathsize="12" mathvariant="sans-serif" mathbackground="#ffff" mathcolor="#ff0066"><m:fenced open="|" close="|"><m:mtext>0/1</m:mtext></m:fenced></m:mrow></m:math></p></mattext>
</material>

After:
<material>
  <mattext><p><m:math><m:mrow><m:fenced open="|" close="|"><m:mtext mathsize="12" mathvariant="sans-serif" mathbackground="#ffff" mathcolor="#ff0066">0/1</m:mtext></m:fenced></m:mrow></m:math></p></mattext>
</material>

```