Overview

When performing batch conversion of assets using the RC (see parent article) you can choose to perform XML conversion to binary format.

Binary XML conversion takes an input XML file, parses it, filters the XML elements and attributes, and stores the node tree in a binary format.

Advantages

- Smaller file size.
- Faster to read at run-time.
- Can be filtered to contain only specific data.

Disadvantages

- Not human-readable.
- Not human-editable.

Conversion

In your Job XML file (see parent article), add a **<Job>** XML block in the <ConvertJob> group like follows:

```
<Properties source_folder="..."/>
<Properties target_folder="..."/>
<Properties xml_types="..."/>
<!-- For SDK conversion, we use xml_types="*.animevents;*.animsettings;*.adb;*.bspace;*.cdf;*.chrparams;*.comb;
*.dlg;*.ent;*.fsq;*.fxl;*.ik;*.json;*.lmg;*.mtl;*.setup;*.xml;*.node;*.veg" -->

<ConvertJob>
        <!-- Add a job similar to this -->
        <Job sourceroot="${source_folder}" input="${xml_types}" targetroot="${target_folder}"
overwriteextension="xml" xmlfilter="xmlfilter.txt" />
</ConvertJob>
```

In this example, the file 'xmlfilter.txt' is referenced, this file is located in `Bin64\RC\xmlfilter.txt`. This file controls which files are converted, and how.

The 'overwriteextension' property is used to force the files matching the input-set to be evaluated as if they were XML files (see parent document, on 'overwriteextension').

Because the output file can be filtered, the conversion is potentially lossy. Like most asset conversions, you should always keep the original file in another location as the converted file to prevent loss of data.

Filter file

The file specified through the 'xmlfilter' property is formatted as flat text, with each filter instruction on a separate line in the file.

While parsing this file, empty lines are ignored, all other lines must be formatted like one of the following:

```
a <modifier> <attribute name>
e <modifier> <element name>
f table <file mask>
```

- A line starting with 'a' declares a filter on a XML attribute. Use the '*' wildcard to match multiple attribute names.
- A line starting with 'e' declares a filter on a XML element. Use the '*' wildcard to match multiple element names.
- The modifier for 'a' or 'e' can be either '+' (to include the matching XML data in the binary data) or '-' (to exclude the matching XML data)
- A line starting with 'f table' declares a filter on XML files that will be subject to table-conversion from Microsoft Excel exported xml table format. Specify a file-mask to apply the filter to the files matching the mask. Use the '*' wildcard to match multiple files.

The filter file's element and attribute filters are evaluated top-down. The first matching rule for each XML item is selected to determine if it will be filtered out or not. If not element of attribute filter matches an XML item, it will not be filtered out.