

Lessons Learned from importing data for Parts with xClassification data and a thumbnail images

1. Documentation

This is just a record of my experience (as a non-programmer) trying to work out how to import the data for a demo system just using AML and Batch Loader. A lot of it was picking clues out from various sources plus a lot of educated guesses. The documentation is insufficient. What I did find was as below:

- A nice introduction, but just an [introduction by Christopher Gillis](#) on the Aras Forum
- The **Programmer's Guide** in the Innovator Documentation. This includes a three-page reference, which feels a bit short.
- The documentation for **Batch Loader** includes some examples at the end

I would love to see further documentation and references, in particular:

- The full extent of what AML can and can't do. Specifically:
 - Can it include logic such as IF statements that depend on the content of the file being imported (i.e. an SQL where condition doesn't help here)
 - Can one concatenate input data. See below where this would have been useful for loading the thumbnail images.
 - Can one import files (e.g. image files) along with the data by using pointers to the file locations with the import sheet?
- A full description of the error messages returned within Batch Loader. These are mostly very obscure and singularly unhelpful.

2. Software used

- **Excel** for
 - Data
 - Lookup for GUIDs to enable importing of Thumbnails
 - Generating AML for Batch Loader
- **Batch Loader**. I assume that the reader has a basic understanding of using this program. If not
- **MS SQL Studio**. Other options also available. Used to get a list of files in the vault along with their GUID

3. Dealing with import errors

The error messages are almost of no use at all.

- The messages themselves are obscure. For example I worked out that “[CDATA[Arithmetic overflow error converting numeric to data type numeric.” means that you are trying to import a decimal number that does not match the Precision and Scale of the field. Other messages are far less useful.
- The row in the input file can (usually) be identified, but the field or column cannot. This leads to a lot of guessing, trial and error.

4. Importing thumbnail images

This is a three-step process. First the images are loaded into the vault, then the file names are exported along with the GUIDs of the image files from which I made a lookup table for the final import, then the images are associated with the new items as part of the import.

Uploading images to the vault

See **4.3.3 Uploading a File Item** in the Batch Loader documentation.

The key fields for the import file look something like this:

FileType	Filename	actual_filename
PNG Image	Part_1_image.png	C:\Images to upload\Part_1_image.png
PNG Image	Part_2_image.png	C:\Images to upload\Part_2_image.png

Avoiding commas and other “exotic” characters in the filenames can be helpful.

Exporting a list of the image files with their GUIDs

Listing Files from within Innovator did not work as I was unable to get it to display the GUIDs correctly. I therefore used MS SQL Server Management Studio for this.

Importing the images along with the new Items

It is sadly not enough to simply run the Batch Loader with the GUID to identify the image.

The AML itself is simple: `<thumbnail type="image">@7</thumbnail>`

However, the contents of that column must be in the form:

vault:///?fileId=F67BC3B4C38E4D57B64AE1BC4BBB8492

Because of the need for the extra **vault:///?fileId=** at the beginning, I used a VLOOKUP to get the GUID from a second sheet based on the file name and `=vault:///?fileId="&RC[-1]` to generate the text for the import. This is converted to straight text when exporting the sheet to the tab-delimited .txt file.

5. Preparing the system for importing items with xProperties

I created a table of all of the new xProperties along with any parameters as needed (e.g. Precision and Scale for Decimal fields) and imported these using Batch Loader

I was then able to build the xClassification tree:

Level	Name	Label	Data Type	Data Source [...]	Default Value
1 KSP Base	xp-ksp_radial_size	x Radial Size	List	KSP Radial Size	
2 KSP Base	xp-ksp_manufacturer	x Manufacturer	Item	Manufacturer	
3 KSP Base	xp-ksp_reliability	x KSP Reliability	String		
4 KSP Base	xp-ksp_mass_empty	x Mass Empty (t)	Decimal		
5 KSP Base	xp-ksp_mass_full	x Mass Full (t)	Decimal		
6 KSP Base	xp-ksp_speed_tolerance	x Speed Tolerance (m/s)	Decimal		
7 KSP Base	xp-ksp_acceleration_tolerance	x Tolerance (g)	Decimal		
8 KSP Base	xp-ksp_max_temp	x Max Temp (K)	Integer		
9 KSP Base/Engine	xp-ksp_burn	Burn (s)	Decimal		
10 KSP Base/Engine	xp-ksp_engine_type	Engine Type	List	KSP Engine Types	
11 KSP Base/Engine	xp-ksp_intake_per_sec	Intake per second	Decimal		
12 KSP Base/Engine	xp-ksp_max_fuel_cons_elec	Max fuel consumption (elec/s)	Decimal		
13 KSP Base/Engine	xp-ksp_max_fuel_cons_units	Max fuel consumption (units/s)	Decimal		
14 KSP Base/Engine	xp-ksp_thrust_atm	Thrust (atm, kN)	Decimal		
15 KSP Base/Engine	xp-ksp_thrust_vac	Thrust (vac, kN)	Decimal		
16 KSP Base/Engine	xp-ksp_thrust_vectoring	Thrust Vectoring (deg)	Decimal		
17 KSP Base/Engine	xp-ksp_twr_empty_atm	Thrust to Weight Ratio Empty atm	Decimal		
18 KSP Base/Engine	xp-ksp_twr_empty_vac	Thrust to Weight Ratio Empty vac	Decimal		

This included some xProperties that belonged to the root of the tree as well as some for each branch.

The Export to Excel function for this table was useful for getting an error-free list of property names to use in the import files.

6. Importing the item data

I imported the data one xClassification branch at a time, mostly to make it easier to identify any errors.

An example of the import script (saves as a .xml file from Batch Loader:

```
<Item type="Part" action="add">
  <item_number>@1</item_number>
  <name>@3</name>
  <classification>@8</classification>
  <thumbnail type="image">@7</thumbnail>
  <Relationships>
    <!--Part Goal is the relationship item that links to item type Goal-->
    <Item type="Part Goal" action="add">
      <goal>@10</goal>
      <actual_value>@11</actual_value>
    </Item>
    <Item type="Part_xClass" action="add">
      <related_id>
        <Item type="xClass" action="get" select="id">
          <name>@12</name>
        </Item>
      </related_id>
    </Item>
  </Relationships>
```

```

<!--The properties below can only be loaded once the xClass relationship has been
established as above-->
<xp-ksp_mass_full set="value">@14</xp-ksp_mass_full>
<xp-ksp_mass_empty set="value">@15</xp-ksp_mass_empty>
<xp-ksp_max_temp set="value">@16</xp-ksp_max_temp>
<xp-ksp_scanning_mode set="value">@90</xp-ksp_scanning_mode>
<xp-ksp_max_scanning_altitude set="value">@91</xp-ksp_max_scanning_altitude>
</Item>

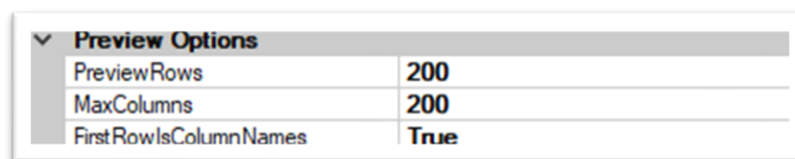
```

Tip: Set Excel to use numbers for column headers instead of letters (R1C1 format) as this makes it much easier to identify the column numbers used (@1, etc.) To do this, select File/Options/Formulas then select **"R1C1 reference style"** in the Working with formulas section

7. Other tips and lessons learned

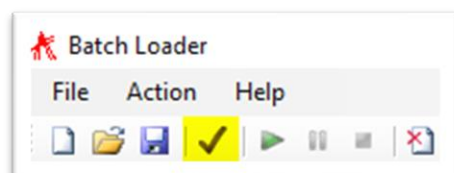
Through much trial and error

- Make sure enough columns are included in the Batch Loader preview pane. This ensures that you can see all of the fields from your import file in the preview.



Preview Options	
PreviewRows	200
MaxColumns	200
FirstRowIsColumnName	True

- Use tab-delimited text files. csv files cause problems if there are commas in any text fields
- Text fields with a " in them get quotes appended and prepended. Replace with ' if possible
- Fields whose contents should match values in an Aras list (data type of the field is List), but do not, are still imported, but surrounded by quotes. This can also be seen in the preview window of Batch Loader. For example, if a list includes Red, Blue, Yellow, but you try to import an invalid value of Green, this field will display as 'Green' in Innovator. This also applies to incorrect capitalisation.
- Booleans must be converted to 0 or 1 in the input file
- Using the verify button not only checks the validity of the AML code, but also checks that the fields are valid, e.g. there were several times that I had not added an xProperty to a branch in the xClassification tree.



- If an import was only partially successful in that Items were created, but not all properties were imported, a second run with the script above will fail. In this case, replace the first line of the AML script above with:
 - <Item type="Part" where="item_number='@1'" action="merge">

- You could also use action="edit"
 - And remove the second line, <item_number>@1</item_number>
 - Note that all "add" commands will generate errors if the field being added already exists.
- As the error messages are so unhelpful, I often used the edit option above, deleted most of the script and added fields back in again a few at a time to identify which field was causing the error.
- An error of **SOAP-ENV:Server.MissingCriteriaException** might mean that Edit is being used instead of Add
- A branch in an xClassification tree may not be named the same as another xClassification tree. Note that there is no error generated when the branch is named, but you get an error when trying to import the xClassification as the field only takes the name of the leaf of the xClassification tree, not the whole path. This generates the error:
 - [CDATA[A single related item has to be specified.]]