

Translation management import/export

Introduction

Translation management export/import is an alternative export/import, with a different format and algorithm, for the purpose of exchanging documents with external translators (translation agencies).

This alternative export/import is implemented by the same tools as the normal export/import (activated by the "-t tm" switch). It supports most of the same configuration options.

The alternative export/import not only differs in format, but also in behavior. Here are some of the important features and differences:

- the format consists of a flat set of ".xml" files. The problem with the default export format is that the documents are identified by directory names, and the content of documents is spread out over multiple files: one for the non-binary data, and one for each part. These files don't have extensions either. In the TM format, there is only one file per document: the content of the parts (as long as it's XML) is embedded within the main ".xml" file. For fields, only string fields (without selection list) are exported. It does not make much sense to translate date fields etc.
- when importing, the data should not be loaded into the same language version as exported, but in another language version
- when importing, documents are only updated with the data present in the import files. Fields, parts and other items that are missing in the import data are not assumed to be deleted (as in the normal import), but only to be missing.
- some fields and parts should be the same in all language variants. Since the concept of "cross-language" fields and parts does not exist in Daisy, these currently have to be synced manually. The TM import tool allows to let these fields be synced upon import.
- on import, the "synced with" version attribute (this is part of the standard Daisy document model) is set to the exported language version.

Example scenario

Performing a 'translation management' export or import is very similar to the normal export and import. Let's look at an example scenario. Suppose we have a number of documents in language 'en' which we want to translate to language 'fr'.

Getting started

Often your Daisy installation will be situated on some server, but you will be calling the tools from your desktop. If you find yourself in this situation, just do the following:

- download a Daisy distribution (not the Windows installer). Use the same version as installed on the server.
- extract the download
- do **not** perform the Daisy installation procedure
- make sure you have the Sun Java runtime (version 1.5 or higher) installed, have JAVA_HOME point to it.
- set DAISY_HOME to point to the directory created when extracting the download
- and for convenience, add Daisy's bin directory to the PATH so that the tools can be called easily

The last two steps are done as follows:

```
[For Windows]
set DAISY_HOME=c:\daisy-{version}
set PATH=%PATH%;%DAISY_HOME%\bin

[For Linux]
set DAISY_HOME=/home/me/daisy-{version}
export PATH=$PATH:$DAISY_HOME/bin
```

Exporting the content to be translated

To create an export, first decide on the documents to export.

Let's suppose the reference variant is 'en' and we want to export the documents for which the 'fr' translation is not in sync or is missing. The query below is the canonical example of how to select those documents.

Create a file called exportset.xml and put the snippet below in it. Of course, you can use any query you like, as long as the resultset only contains documents belonging to the same language variant.

```
<documents>
  <query>
    select
      id
    where
      branch = 'main'
      and language = 'en'
      and referenceLanguage = 'en'
      and ( ReverseLangNotInSync('fr', 'last') or variants has none('main:fr') )
      and documentType = 'SimpleDocument'
      and InCollection('my collection')
    option
      search_last_version = 'true'
  </query>
</documents>
```

Invoke the export tool like this:

```
daisy-export -t tm
              -f exportset.xml
              -l http://localhost:9263
              -u testuser
              -e exportdata.zip
```

The -t switch is for selecting the tm export format (tm stands for translation management).

The -l switch indicates the location of the server, change “localhost” by the appropriate host name. If it is localhost, you can drop the -l option altogether.

The -u option specifies the Daisy user to connect to the repository (you will be prompted for the password).

The -e option tells to do an export to the given location. If this location ends with “.zip”, a zip file will be created, otherwise a directory will be created.

To see all available options, execute:

```
daisy-export -h
```

The translation

Normally you will now send this `exportdata.zip` file to your translation agency, which will update the files and send you back the `.zip` file. The structure of the file should still be the same, but the content will have been replaced by the translated content.

Importing the translated content

Invoke the import tool like this:

```
daisy-import -t tm
              -u testuser
              -i exportdata.zip
              --target-language fr
              --htmlcleaner-config c:\daisy\daisywiki\webapp\daisy\resources\conf
\htmlcleaner.xml
```

The target language `fr` should be an existing language variant. You can define these via the Administration console of the Daisy Wiki.

Specifying the HTML cleaner configuration file will make sure the HTML content gets formatted in the same way as when editing through the Daisy Wiki. This is optional, but highly recommended.

Non-translatable (= language-independent) fields and parts

It might be that your documents contain parts and fields which should not be translated, but rather be the same for all language variants. At the time of this writing, Daisy did not have a feature to support this directly. However, the import tool allows to sync these fields and parts while performing the import.

To specify which parts and fields these are, you can create a little configuration file. This file can be specified both while exporting and while importing. When exporting, it will cause the specified fields and parts not to be part of the export. When importing, it will cause the content for these fields and parts to be taken from the original exported variant version.

Next to this, the document type of the document and the collections are also synced automatically, as described further on.

The configuration file to define the language independent parts and fields has the following format:

```
<tm-config>
  <language-independent>

    ... zero or more part elements ...
    <part type="{...}" documentType="{...}" />

    ... zero or more field elements ...
    <field type="{...}" documentType="{...}" />

  </language-independent>
</tm-config>
```

The `documentType` attribute on the part and field elements is optional. If not specified, the field or part will be considered language-independent for all document types, otherwise only for the specified document type.

The configuration file can be supplied to the `daisy-import` and `daisy-export` tool with the option `-g` or `--tm-config`. For example:

```
daisy-import -t tm --tm-config my-tm-config.xml {... more options ...}
```

WARNING

The functionality described above is only of limited help for language-independent content. For example, suppose a reference variant contains only (major) changes to the language independent fields. In that case, it makes no sense to re-export the document for translation, however the changes to the language-independent fields will still have to be applied in the translated variants, and the synced-with pointer updated correspondingly. The best solution would be to have real language-independent content, managed separately from the language-specific content. See [this wiki page](#)¹ for some thoughts on this subject.

Miscellaneous

Automatic synchronisation of document type and collections

When importing, the document type and the collections are automatically synchronized with those of the exported variant. This behaviour is currently not configurable.

Setting the 'synced with' pointer

Upon import, if a new version has been created, the 'synced with' pointer of this new version will point to the export language variant version.

If no new version has been created because the document already contained the same data as in the export, than the 'synced with' pointer of the last version will be updated to point to the exported version, unless the 'synced with' would already point to a newer version.

No creation of new documents

The TM import never creates new documents, it only updates variants or adds variants.

Since the import tool reads the exported document upon import (e.g. for syncing the language-independent properties), it is required the document already exists anyway.

An export can only contain documents from one language variant

Yes.

Subset imports

If you want to specify a subset of documents to import, you can do this in the same way as for the normal import tool, but you need to know that the set of documents specified should be the target variants of the import.

Export format

The export consists of a set of XML files following this naming pattern:

1. daisy:547-cd@main (Language-independent content)

```
{docid}~{branch}.xml
```

There is no language in the file name, as all exported documents are of the same language.

```
<ie:document xmlns:ie="http://outerx.org/daisy/1.0#tm-impexp" exportedLanguage="{...}"
  exportedVersion="{...}">

  <ie:title>{the document name}</ie:title>

  ... zero or more ie:part elements ...

  <ie:part name="{part type name}" mimeType="{...}">
    {XML content of the part inlined}
  </ie:part>

  ... zero or more ie:link elements ...

  <ie:link>
    <ie:title>{title}</ie:title>
    <ie:target>{target}</ie:target>
  </ie:link>

  ... zero or more ie:field elements ...

  Syntax for single-value fields:

  <ie:field name="{field type name}">{field value}</ie:field>

  Syntax for multi-value fields:

  <ie:field name="{field type name}">
    <ie:value>{value 1}</ie:value>
    <ie:value>{value 2}</ie:value>
  </ie:field>

</ie:document>
```

The export also contains an info subdirectory containing the files meta.xml and namespaces.xml. These files have the same format as the [normal export format](#)².

2. daisy:332-cd (Import/export format)