

Modules ExportCSV, ImportCSV

V1.0 December 2012

Utilities for MODX Evolution (tested with Evo 1.0.6, PHP 5.3.17)

These modules are intended to assist when copying sets of resources from one MODX site to another, e.g. when you wish to create a standard install. ExportCSV creates a CSV file containing all the fields from the `site_content` table for the specified root document and its children (maximum depth of ten). The template ID is stored as the template name rather than its ID to make importation into an existing site easier. Template Variables associated with the resources are also stored in the CSV file, along with their current value.

ImportCSV reads the CSV file and creates a set of new resources with the same structure in the receiving site, as children of the root resource. The imported root is inserted into the document tree under the site root.

ImportCSV makes use of the Document class developed for use with the PubKit snippet. This class does not handle every field. See the appendix for a list of the fields that will be imported.

Installation

The zip package contains all the necessary files in a structure that can be merged with the standard MODX 1.0.6 installer, in case you are making a customized installer.

To install manually:

- Copy the `assets/modules/impex` folder to your site's `assets/modules` folder
- Create a new module (Modules > Manage Modules > New Module)
- Open `install/assets/modules/ImportCSV.tpl` in a text editor and copy contents to the clipboard
- Paste module code into the module
- Copy the text after **@properties** in the installer file, go to the module's Configuration and paste it into the Module configuration input field, then click on Update parameter display
- Return to the main module editing screen Save the module. Prune the comments if it suits you.
- Repeat the above steps to create **ExportCSV**.

Note that new modules do not show in the Modules menu bar until you log out and log back in to the Manager.

The forms for entering parameters for these modules are stored in **import.html** and **export.html**. These contain placeholders in the usual format to preserve data. You can use chunks instead of the files if preferred. Simply copy the content of the HTML file into a chunk, and name the chunk as a module parameter called **&formTpl**, e.g. **&formTpl=Form template;string;import.form.tpl**

Operation

Export

To export a set of resources, go to **Modules > ExportCSV** and specify the root resource if the set, and the directory and filename to store the data. You can specify directories under assets without a full path, e.g. **assets/export**. Otherwise you must specify the full path to the directory. Click on **Proceed**, and you should quickly see a list of the exported documents.

To export everything under the site root, enter the root ID as **-1**.

Import

To import resources from a CSV file, you have a choice of locations:

- any CSV files placed in assets/import are listed as choices when you run the module
- any files listed in the module configuration with a full file path are also listed
- you can enter a full path to any file in the input field in the module's parameter entry screen. NB make sure you also select the radio button next to the input field – selection is not as yet automatic.
- There is also a choice of “No resource files”, so you can exit without importing anything, or just import tables and no resources.

To import a custom table into the MODX database, the options are pretty much the same as above, looking for SQL files in the import folder. This feature is intended only for custom tables that have been exported using phpMyAdmin or the Manager's backup command. Use for MODX system tables would be decidedly risky.

Templates, TVs and snippet parameters

If the receiving site has templates and TVs with names that match those associated with the resources in the exporting site, the corresponding values should be transferred along with the resources.

IDs specified in basic Ditto and PubKit parameters, e.g. **&parents='12'**, are converted to use the new IDs generated by ImportCSV. All the same, check carefully and adjust as required all snippet calls that IDs in parameters.

Alias

The Alias field is exported and imported “as is”. **It is not checked for duplication on import.** This means you may have to change the alias of an imported resource if it clashes with an existing one when you later **Save** the resource.

Appendix

Parameters

Export

&root=Root;int; *<ID of root for set of exported pages, or -1 for all>*

&exportDir=Export to;string; *<path to directory for CSV file. Trailing slash optional>*

&exportFile=File;string; *<output filename>*

&formTpl=Form template chunk;string; *<name of form template chunk>*

Import

&dataFile=Data file;string; *<full path to CSV file>*

&formTpl=Form template chunk;string; *<name of form template chunk>*

Imported fields

The Document class handles the following fields in version 1.0 (maybe the list will be expanded in future versions):

content, parent, pagetitle, longtitle, alias, published, pub_date, unpub_date, isfolder, introtext, richtext, template (by name, not ID), menuindex, menutitle, hidemenu