

Cloning Column Properties

It is not uncommon for a research project to measure many of the same parameters at different locations, say, or in different contexts. In such cases, the same parameters can end up as data table columns in different tables and even different datasets. It is useful, then, to “clone” column properties from one table to another. ezEML provides such a capability. This is a convenience, but it is also a small step toward metadata standardization.

The ability to clone column properties from another table is also very useful in scenarios where you’ve modified a table by adding, deleting, or reordering columns, for example. When a table’s *structure* has changed the **Re-upload** feature cannot be used, but you would like to be able to upload the modified table without having to re-enter the metadata for the columns whose properties are unchanged. Cloning the properties makes that possible.

Suppose, for example, we have created a dataset with a data table called Data_1, and we have entered all of its metadata. Then we upload another data table, Data_2, with most of the same columns. We want to clone a number of column properties from Data_1 to Data_2. Suppose our list of data tables looks like this:

The screenshot shows the ezEML web interface. At the top is a navigation bar with links: ezEML, EML Documents, Import/Export, EDI Info, User Guide, About, News, and a Logout button. Below the navigation bar, it says "Welcome Back EDI" and "Active EML Document: sample.1.1".

The main content area is titled "Data Tables" and contains a table with the following data:

Data Table Name					
Data_1	▼	Edit	Remove	Re-upload	?
Data_2	▲	▼	Edit	Remove	Re-upload
Nitrogen data	▲	Edit	Remove	Re-upload	

Below the table are two buttons: "Load Data Table from CSV File" and "Add Data Table from Scratch". At the bottom of the main content area is a "Save and Continue" button.

On the left side, there is a "Contents" sidebar with a list of metadata fields: Title, Data Tables (selected), Creators, Contacts, Associated Parties, Metadata Providers, Abstract, Keywords, Intellectual Rights, Geographic Coverage, Temporal Coverage, Taxonomic Coverage, Maintenance, Publisher, Publication Info, Methods, Project, Other Entities, and Data Package ID. At the bottom of the sidebar are links: "Check Metadata", "Send to EDI", and "Send to Colleague".

We’ve already filled out all the metadata for Data_1, and we want to reuse its column properties in table Data_2. In this scenario, we will refer to Data_1 as the **source** data table and Data_2 as the **target** data table.

We click **Edit** next to Data_2, opening a page that contains the following (only part of the page is shown):

Columns:

VEGETATION_TYPE, YEAR_PLACED_IN_FIELD, NTRT, YEAR, PERCENT_LOSS, PLOT_ARM

Edit Column Properties

Clone Column Properties from Another Data Table



Save and Continue

Cancel

We click **Clone Column Properties from Another Data Table**. This brings up a page where we select the EML document containing the source data table:

Clone Column Properties from Another Data Table

Select a data package from which to clone:

Document Name

sample.1.1 (current data package) 

Open for Cloning

Cancel

For this example, we're assuming Data_1 and Data_2 are both in the current data package, which we've selected, but Data_1 is allowed to be in a different data package. Click **Open for Cloning**. This opens a page that contains:

Clone Column Properties from Another Data Table

Select a data table from which to clone column properties:

☒ Data_1

☐ Nitrogen data

Select Data Table

Cancel

We need to select which table will be the source table. (Table Date_2 is not listed as an option because it wouldn't make sense to clone a table onto itself.) We select Data_1 and click **Select Data Table**. This brings up a page that contains:

Clone Column Properties ?

From source data table: Data_1

Select the columns whose properties you wish to clone:

- ☒ TYPE
- ☒ YEAR_PLACED_IN_FIELD
- ☒ ARM
- ☒ NTRT
- ☒ YEAR
- ☐ PERCENT_LOSS

Select Columns

Cancel

We need to select the source data table's columns we want to clone. For the purposes of this example, let's suppose we want to clone all but PERCENT_LOSS. We select all of the other columns and click **Select Columns**. This brings up a page that contains:

Clone Column Properties ?

From source data table: Data_1

To target data table: Data_2

Source Column	→ Target Column
TYPE	<input type="text" value="- select target column -"/>
YEAR_PLACED_IN_FIELD	<input type="text" value="- select target column -"/>
ARM	<input type="text" value="- select target column -"/>
NTRT	<input type="text" value="- select target column -"/>
YEAR	<input type="text" value="- select target column -"/>

Clone Column Properties

Cancel

For each of the source columns, we need to select the corresponding column in the target data table. Note that the columns don't have to be named identically between the two tables, nor do they need to appear in the same order. We select the target columns, giving something like:

Clone Column Properties ?

From source data table: Data_1

To target data table: Data_2

Source Column	→ Target Column
TYPE	VEGETATION_TYPE ▾
YEAR_PLACED_IN_FIELD	YEAR_PLACED_IN_FIELD ▾
ARM	PLOT_ARM ▾
NTRT	NTRT ▾
YEAR	YEAR ▾

Clone Column Properties

Cancel

We then click **Clone Column Properties** to cause the properties to be cloned. This is potentially a significant change, so you are asked to confirm the action before cloning takes place.

After cloning, the selected columns in the target data table will have the same properties as the corresponding columns in the source data table with two exceptions:

1. The column names are not changed. I.e., the target data table columns keep the names they have in the target data table's CSV file.
2. For Categorical columns, the *codes* in the target data table are not changed, but their *definitions* are filled in by using the definitions for the corresponding codes in the source data table. This way, the lists of codes will match what's in the target data table's CSV file, but the code definitions won't all have to be re-entered for codes that are in common between the two tables.

Note: For tables with many columns, you may prefer to do this cloning a few columns at a time, just so the process feels more manageable. This is fine to do. You can work on a given table a few columns at a time by running through the **Clone Column Properties from Another Data Table** process repeatedly, selecting different columns on each pass until you've completed all the columns you wish to clone.