

LG Data Matrix v1.0.0

Create a table of data using one custom field

Features

LG Data Matrix is an [ExpressionEngine](#) custom field extension that allows site editors to publish multiple records using a single custom field. It has been developed as a replacement for Mark Huots [Multi-Text extension](#) first published in 2006.

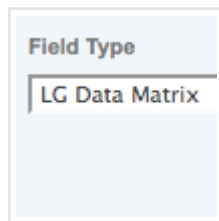
LG Data Matrix allows you to sort "rows" of data using simple drag and drop and add / delete rows using javascript. Cells values can be created as either a text, textarea or select data type. Each custom field has its own configuration making this extension extremely flexible.

LG Data Matrix is also MSM compatible and integrates with [LG Addon Updater](#).

Screenshots



A configured custom field



Custom field settings



Extension Settings

Requirements

LG Data Matrix requires [ExpressionEngine 1.6+](#). Update notifications will only be displayed if [LG Addon Updater](#) is installed.

Installation

The LG Data Matrix extension contains an extension and language file. To install LG Data Matrix follow the instructions below:

- [Download](#) the latest version of LG Data Matrix
- Extract the .zip file to your desktop
- Copy extensions/ext.lg_data_matrix_ext.php file to your /system/extensions directory
- Copy extensions/lg_data_matrix_ext directory to your /system/extensions directory
- Copy the language/english/lang.lg_data_matrix_ext.php file to your /system/languages/english directory

- Open the [Extension Manager](#)
- Enable Extensions if not already enabled
- Enable the extension
- Configure the extension settings

Configuration

LG Data Matrix has the following extension settings **which need to be entered separately for each Multi-Site Manager site**:

- [Enable](#)
- [jQuery Core](#)
- [jQuery UI](#)
- [Check for updates](#)

Enable LG Data Matrix extension for this site? [required]

Yes | No

Enabling the extension will allow site administrators to create a LG Data Matrix custom field.

jQuery Core [required]

`http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js`

LG Data Matrix requires [jQuery 1.2.6+](#) to function correctly. By default this is set to use Googles AJAX API.

jQuery UI [required]

`/system/extensions/lg_data_matrix/scripts/jquery.ui.core+interactions.1.6.min.js`

LG Data Matrix also requires [jQuery UI 1.6+](#) with "Interactions" to function correctly. This file is included in `/extensions/lg_data_matrix/scripts` folder however if you use more jQuery UI features you may [consider building your own](#).

Check for updates?

Yes | No

LG Data Matrix can call home and check for recent extension updates. **This feature requires [LG Addon Updater](#).**

Usage

Creating a custom field

Before any content can be added the site editor must create a new LG Data Matrix [custom field](#).

1. Create a new custom field.

```
Admin > Weblog Administration > Custom Weblog Fields > Add/Edit
```

2. Add a field name (I will use my_matrix in the demo below), label and instructions where appropriate
3. Choose LG Data Matrix as the custom field type
4. Add the [field configuration](#)
5. Save the new custom field

Field Configuration

Each custom field has its own field configuration that allows you to control the number and type of column. The three available types are text, textarea and select. Default values are added when a new field is created.

Probably best if we start with the default example:

```
short_name = col_1  
title = Column 1  
type = text  
  
short_name = col_2  
title = Column 2  
type = textarea  
  
short_name = col_3  
title = Column 3  
type = select  
options = One|Two|Three
```

In the example above the custom field has three columns separated by a blank line. Each of the columns attributes is separated by a single line break. Each attribute is separated into its key and value using an equals "=" sign.

Every column must include the following keys:

short_name= required

```
short_name = col_1
```

The column short name will be used as the template tag in your ExpressionEngine templates.

title= required

```
title = Column 1
```

The title is used in the publish / edit form and will also be available as a template tag.

type= required

```
type = text|textarea|select
```

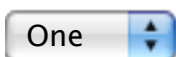
The type determines the possible html elements in the publish / edit form. Only one of the three types can be chosen.

options="" [required for select]

Additionally the `select` field type requires an options key in the following format.

```
options = One|Two|Three
```

In the example above the select box will have three options; one, two and three similar to the select element below.



In some cases you may want the value of the option to be different to the label. If this is the case separate the value and the label like so:

```
options = raspberrylease Choose|one:One|two:Two|three:Three
```

In the example above the "Please Choose" option will not have a value. If your field is required a non-value label must be used. The above example will create a select like the one below:



Outputting data in templates

Now that the custom field has been setup (I used `my_matrix` as the custom field name) create a new entry after entering data in the new custom field.

In your template the custom field short name (`my_matrix`) will be used as a tag pair to loop over each of the matrix rows. Two simple examples are provided below:

```

{exp:weblog:entries}
  <!--
  lg_data_matrix is our custom field tag
  limit = "" limits the number of rows
  flip = "y" Flips the row order so that the last row shows first
  disable = "cells|headers|rows" limits the tags that are parsed inside your custom field.
  I highly recommend using the disable paramter if you are not using the tags in the loop.
  It will speed up performance
  -->
  Example 1: A basic Table
  {my_matrix limit="2"}
  <table border="1">
    <thead>
      <!--
      Loop over the column headers using the {header} tag
      Two tags are available:
      {col_title} - The column title
      {col_short_name} - The column short name (good for css classes)
      -->
      <tr>{headers}<td>Column Short Name: {col_short_name} Title: {col_title}</td>{/headers}</tr>
    </thead>
    <tbody>
      <!--
      Loop over each of the rows using the rows tag
      -->
      {rows}
      <tr>
        <!--
        Loop over each cell in the row
        Three tags are available:
        {col_title} - The column title
        {col_short_name} - The column short name (good for css classes)
        {col_data} - The cell data
        -->
        {cell}<td>{col_short_name} {col_title}: {col_data}</td>{/cell}
      </tr>
    </tbody>
  </table>
  {/my_matrix}

  Example 2: Simple output of rows
  {my_matrix limit="2" disable="headers|cells" }
  <!-- Loop over our rows -->
  {rows}
  <ul>
    <!-- Output the data -->
    <li>Column 1: {col_1}</li>
    <li>Column 2: {col_2}</li>
    <li>Column 3: {col_3}</li>
  </ul>
  {/rows}
  {/my_matrix}

{/exp:weblog:entries}

```

Showing the first row only

There may be times when you only want to show the first row of the matrix, for instance only display the latest download version of a piece of software. In this case you can just use the column short names as the cell values.

An example:

```
{lg_data_matrix disable="headers|rows|cells"}
  {col_1} <--- The first row value for col_1
{/lg_data_matrix}
```

Tag Reference

LG Data Matrix uses your custom fields short name as the primary tag pair.

Primary Tag Parameters

The primary tag pair has the following parameters:

- [disable="cells|headers|rows"](#)
- [flip="y"](#)
- [limit="10"](#)

disable= optional

```
disable="cells|headers|rows"
```

This will disable rendering of the cells, headers or rows nested tag pair. Using this parameter will result in processing speed improvements.

flip= optional

```
flip="y"
```

Setting this parameter to 'y' will cause the data matrix to reverse the sorted rows.

limit= optional

```
limit="10"
```

Limits the number of rows rendered to the template.

Single Variables

The following single variables are also available inside the primary tag pair:

- [{total_row_count}](#)

[{total_row_count}](#)

The total number of rows being shown.

Variable Pairs

LG Data Matrix also has the following nested tag pairs:

- [{headers}](#)
- [{rows}](#)
- [{cells}](#)

[{headers}](#) ... [{/headers}](#) [nested in primary tag pair]

```
{my_matrix}
<table border="1">
  <thead>
    <!--
      Loop over the column headers using the {header} tag
      Two tags are available:
      {col_title} - The column title
      {col_short_name} - The column short name (good for css classes)
    -->
    <tr>{headers}<td>Column Short Name: {col_short_name} Title: {col_title}</td>{/headers}</tr>
  </thead>
</table>
{/my_matrix}
```

The headers tag will loop over each of your matrix columns. The primary use for this tag is adding headers to table output.

The following tags will be replaced with their corresponding values inside this tag pair:

- col_title
- {col_short_name}

[{rows}](#) ... [{/rows}](#) [nested in primary tag pair]

```

{my_matrix}
<table border="1">
<!--
The following tags are available in the {rows} tag pair:
{row_count} - The current row number
-->
{rows}
<tr>
<!--
Loop over each cell in the row
Three tags are available:
{col_title} - The column title
{col_short_name} - The column short name (good for css classes)
{col_data} - The cell data
-->
{cell}<td>{col_short_name} {col_title}: {col_data}</td>{/cell}
</tr>
{/rows}
</table>
{/my_matrix}

```

The {rows} and {cells} tag pairs work closely together to render the actual content of the data matrix. They should be pretty self explanatory from the example above.

The following single tag is available in the {rows} tag pair:

- [{row_count}](#)

{row_count}

The current row number being displayed.

{cells} ... {/cells} [nested in {rows}]

The {cells} tag pair must be nested inside the {rows} tag pair as it's purpose is to render each cell of the row.

The following single tags are available inside the {cells} tag pair and all are available as conditionals.

- [{col_title}](#)
- [{col_short_name}](#)
- [{col_data}](#)

{col_title}

The column title for this cell.

{col_short_name}

The column short name for this cell. Good for use in CSS classes.

{col_data}

The cell value.

Conditionals

Each of the single tags can also be used as conditionals.

Change Log

1.0.0

- Initial release with documentation.

License

LG Data Matrix is a free commercial product and therefore its usage is subject to the [commercial license agreement](#).