Home → CMS Customisation → ExpressionEngine → LG Data Matrix

Do you need a custom CMS addon or theme? - Contact Me

Looking for ExpressionEngine v2 addons? Visit my new site: expressionengine-addons.com

LG Data Matrix v1.1.1
Highly configurable data matrix custom field ExpressionEngine extension.

⊙ DOWNLOAD LG DATA MATRIX 1.1.1

Using the donationware version?
Consider buying me coffee.

PayPal DONATIONS = ☕

Contributors

1. Luke Stevens | Design + Consulting

Enjoy LG Data Matrix v1.1.1?
Bookmark and share it with others.

💬 ▪ 📇 f f G ● 🌠
🐵 🌀 ⤴ ⤵

Publish PLUS

Collaborative publishing for *ExpressionEngine*

Improve your workflow with drafts, revisions, notes, and member-based privileges.

Buy now ☆

1. Overview   Top

LG Data Matrix is a Multi-Site Manager compatible ExpressionEngine custom field extension that allows site editors to create multiple records using a single custom field.

It was developed as a replacement for Mark Huots Multi-Text extension first published in 2006.
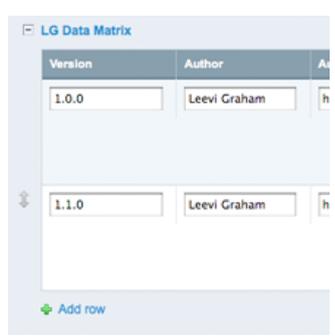
2. Features   Top

LG Data Matrix allows you to dynamically add / delete and sort rows of data similar to a spreadsheet. Each data row can have one or more cells which are assigned either a text, textarea, select, checkbox or date data type.
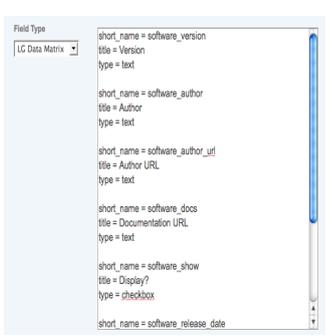
Each custom field is configured independently allowing you to create different cell combinations making this extension extremely flexible.

- Independent field configuration
- Custom template tags w/ date formatting options
- Multiple cell types:
  - Text input
  - Textarea
  - Select box
  - Checkbox
  - Dates
- Quicksave, preview compatible
- Multi-Site Manager compatible
- Newism Publish Plus compatible
- CP jQuery compatible

3. Screenshots   Top

Custom Field - Entry Form   Custom Field - Configuration

4. User guide   Top

4.1. Requirements

LG Data Matrix is an ExpressionEngine extension that has been tested on version 1.6.4+

To function correctly javascript must be enabled and the jQuery javascript library must be loaded in the control panel. jQuery can easily be included using the CP jQuery extension which is packaged with ExpressionEngine 1.6.6+. CP jQuery must also be configured to use jQuery 1.2.6 and jQuery UI 1.5.3 with "Interactions".

The user must also be using a modern web browser (any thing better than IE6).

4.2. Download

Download the latest version of LG Data Matrix or view and download previous versions.

4.3. Installation

To install LG Data Matrix follow the instructions below:

1. Download the latest version of LG Data Matrix

2. Extract the .zip to your desktop or similar folder where you can find it easily

3. Copy `/system/extensions/ext.lg_data_matrix_ext.php` file to your ExpressionEngine `/system/extensions/` directory

4. Copy `/system/languages/english/lang.lg_data_matrix_ext.php` file to your ExpressionEngine `/system/languages/english/` directory

5. Copy the contents of `/themes/site_themes/cp_themes/default/` to your ExpressionEngine `/themes/site_themes/cp_themes/default/` directory

6. Copy the contents of `/themes/site_themes/cp_global_images/` to your ExpressionEngine `/themes/site_themes/cp_global_images/` directory

4.4. Activation

1. Open the Extensions Manager in your sites control panel

2. Enable Extensions if not already enabled

3. Enable LG Data Matrix

4. Configure and save the extension settings for each of your sites

Page created on: Aug 26, 2008
Last updated on: Feb 03, 2009

4.5. Configuration

4.5.1. Extension settings

Extension access

Enable LG Data Matrix for this site? [required]

Enabling LG Data Matrix will add a new custom field option in the publish / edit form.

Check for updates

Would you like this extension to check for updates and display them on your CP homepage? [required]

If you allow it LG Data Matrix can call home (http://leevigraham.com) and check for recent extension updates. This feature requires LG Addon Updater.

4.6. Creating a new 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. *Control Panel > 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. Configure the field

5. Save the new custom field

Each LG Data Matrix custom field has its own field configuration that allows you to control the number of cells and type of data stored in the cell.

There are five available cell types: text, textarea, select, date and checkbox.

The example below demonstrates all five cell types:

```
short_name = software_version
title = Version
type = text

short_name = software_name
title = Name
type = text

short_name = software_release_date
title = Release Date
type = date

short_name = software_release_notes
title = Release Notes
type = textarea

short_name = software_show
title = Display?
type = checkbox
```

In the example above the custom field cell has three parameters separated by a blank line. Each of the cells 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 = software_version
```

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

title= required

```
title = Version
```

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

type= required

```
type = text|textarea|select|date|checkbox
```

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

type = select

The select cell type also requires the "options" parameter.

type = date

The date cell with store values as a UTC timestamp which can be formatted using standard ExpressionEngine date variable formating.

Dates must be entered in the following format: `2009-02-02 05:58 PM`.

type = checkbox

Checkbox cells will always have the value of "y" when checked.

options= [required for select]

Additionally the *select* field type requires an options key in the following format.
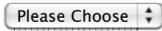
```
options = One|Two|Three
```

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

One ⬍

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 = Please Choose|one:One|two:Two|three:Three
```

In the example above the "Please Choose" option will not have a value. It will create a select like the one below:

Please Choose ⬍

width= optional

```
width = 20
```

The width of the cell (percentage)

4.7. 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 examples are provided below:

4.7.1. Simple example

The following example loops over the matrix rows using the cell short name as the template tags:

```
{exp:weblog:entries}
  <!--
    Example 1: Simple output of known cells
  -->
  {my_matrix disable="headers|cells"}
    <!-- Loop over our rows -->
    {rows}
      {if software_show == "y"}
        <ul>
          <!-- Output the data -->
          <li>Name: {title}</li>
          <li>Version: {software_version}</li>
          <li>Release Date: {software_release_date format="%Y %m %d"}</li>
          <li>Release Notes: {software_release_notes}</li>
        </ul>
      {/if}
    {/rows}
  {/my_matrix }
{/exp:weblog:entries}
```

4.7.2. Full featured example

The following example loops over the matrix adding a table header and outputting 2 matrix rows:

```
{exp:weblog:entries}
  <!--
    Example 2: Looping over an unknown number of cells with table headers
    my_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
  -->
  {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>
        <td>row_count</td>
        {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
          Four tags are available:
          {col_title} - The column title
          {col_short_name} - The column short name (good for css classes)
          {cell_data} - The cell data
          {cell_type} - The cell type
        -->
        <td>{row_count}</td>
        {cells}
        <td>
          {col_short_name} {col_title}:
          {if cell_type == "date"}
            {cell_data format="%Y %m %d"}
          {if:else}
            {cell_data}
          {/if}
        </td>
        {/cells}
      </tr>
      {/rows}
    </tbody>
  </table>
  <p><strong>Total Rows</strong>: {total_row_count}</p>
  {/my_matrix }
{/exp:weblog:entries}
```

---

5. Reference Guide

5.1. Primary variable pair

LG Data Matrix uses your custom fields short name as the primary variable pair. The custom field variable pair must be placed inside your {weblog:entries} tag pair.

Example: If your LG Data Matrix custom field short_name was "my_matrix" the primary variable tag pair would be *{my_matrix} … {/my_matrix}*

5.1.1. 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="10"
```

Limits the number of rows rendered to the template.

5.1.2. 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.

5.1.3. Nested variable pairs

The following variable pairs can be nested inside the primary variable pair.

- *{headers}*
- *{rows}*
- *{cells}*

{headers} … *{/headers}* [nested in primary variable pair]

```
{exp:weblog:entries}
  {my_matrix}
    {headers}
        Column Short Name: {col_short_name}
        Title: {col_title}
    {/headers}
  {/my_matrix}
{/exp:weblog:entries}
```

The headers tag will loop over each of your matrix columns outputting the column title and column short name. 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}* - The column title
- *{col_short_name}* - The column short name (good for css classes)

{rows} … *{/rows}* [nested in primary variable pair]

```
{exp:weblog:entries}
  {my_matrix}
    {rows}
      <p>Row number: {row_count}</p>
    {/rows}
  {/my_matrix}
{/exp:weblog:entries}
```

The *{rows}* tag pair loops over each matrix row. It is generally used in conjunction with the *{cells}* variable pair.

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

- *{row_count}* - The current row count

{cells} … *{/cells}* [nested in rows variable pair]

```
{exp:weblog:entries}
  {my_matrix}
    {rows}
    <ul>
      <li>Row number: {row_count}</li>
      {cells}
      <li>
        {col_short_name} {col_title}:
        {if cell_type == "date"}
          {cell_data format="%Y %m %d"}
        {if:else}
          {cell_data}
        {/if}
```

```
        </li>
      {/cells}
    </ul>
  {/rows}
{/my_matrix}
{/exp:weblog:entries}
```

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}* - The column title
- *{col_short_name}* - The column short name (good for css classes)
- *{cell_data}* - The cell data
- *{cell_type}* - The cell type

₆. Release notes

₆.₁. Version 1.1.1

Download:
http://leevigraham.com/downloads/lg-data-matrix.1.1.1.zip

₆.₁.₁. Changelog

- Added date and checkbox field types (Thanks Brandon Kelly)
- Fixed most data serialisation issues
- Tweaked template tags

₆.₁.₂. Updgrade Notes

- Changed *{cell}* to *{cells}*
- Changed *{col_data}* to *{cell_data}*

₆.₂. Version 1.0.0

Download:
http://leevigraham.com/downloads/lg-data-matrix.1.0.0.zip
Documentation:
http://leevigraham.com/downloads/docs/lg-data-matrix-1.0.0.pdf

₆.₂.₁. Changelog

- Initial Release w/ documentation

₇. Other topics

₇.₁. Support

Technical support is available primarily through the ExpressionEngine forums. Leevi Graham and Newism do not provide direct phone support. No representations or guarantees are made regarding the response time in which support questions are answered.

₇.₂. Upgrades

Although we are actively developing LG Data Matrix, Newism makes no guarantees that this extension will be upgraded within any specific timeframe.

{cell_data}

8. License

8. **License** Top

Available For Hire

Need a website, blog or CMS customisation?
[Contact me and I will be happy to discuss your project.](#)

Powered by: [ExpressionEngine](#)
Hosted with [Dreamhost](#)
Save $80 on hosting | Promo code: **saveheaps**.

My mum: [Newcastle massage - Remedial, sports, relaxation & Bowen therapist - BodyKneads.com.au](#)