

# Digital Hill Multimedia, Inc.

229 S. Main St.  
Goshen, IN 46526

888-537-0703

[www.digitalhill.com](http://www.digitalhill.com)

---

This document assumes that you have created an account with SSH access, a MySQL database, and a MySQL user with permissions to work with the database.

---

## Core System Install

If your server does not have the core system installed, clone the core repository. Only one copy of the core should be installed per server so we can update the core system with one git pull. As of this writing the repo is 168M. The SSH password for coredigitalhill is located in the office system under core.digitalhill.com.

```
git clone coredigitalhill@core.digitalhill.com:/home/coredigitalhill/repo digitalhill
```

The content in the core folder should be readable by all users.

```
chmod 755 digitalhill/  
find digitalhill/ -type d -exec chmod 755 {} +  
find digitalhill/ -type f -exec chmod 644 {} +
```

## Site Install

Copy the contents of `digitalhill/site/` into your `~/public_html/` folder. Ensure that your user is not enabled in CageFS. If so, perform the following command as root:

```
/usr/sbin/cagefsctl --disable [username]
```

The site files must be owned by the user and the permissions must be set appropriately. Use something like the following:

```
cp -r digitalhill/site/. ~/public_html/
find ~/public_html/ -type d -exec chown [username]:[username] {} +
find ~/public_html/ -type f -exec chown [username]:[username] {} +
find ~/public_html/ -type d -exec chmod 755 {} +
find ~/public_html/ -type f -exec chmod 644 {} +
```

Install the database:

```
mysql -u mysql_user -p mysql_database < ~/public_html/site.sql
```

## Site Configuration

Modify the site `config.php` with the appropriate MySQL credentials. Modify the site `index.php` to reference the correct location for the core `config.php` and core `content.php`

Modify the contents of the `'dh_configuration'` table as necessary. The `'widgets'` option is a comma delimited list of the widget IDs the client has access to. The `'navigation_size'` is a number indicating how many navigation links are present in the header and footer. The `'modules'` option is a comma delimited list of the modules the client has access to. The modules should include HOME and may include any of WORDPRESS, MAGENTO, or PRODUCT if the client needs access to these modules. Update the `'preview_width'` to contain the full site, and set `'preview_responsive'` to 0 or 1 based on whether the site is responsive. The `'parallax'` option should be 0 or 1 based on whether the parallax template should be made available.

At this point you should have a working base site. The admin section for the site can be accessed at `www.yoursite.com/admin`.

There is a master login for Digital Hill employees that will work across all sites using the core system. The username is `digitalhill`, and the password can be found in `core/config.php` under `MASTER_PASSWORD`.

To add a client user to the admin section, add your email address to the `'dh_users'` table, and set the `'active'` flag to 1. You can now follow the "Forgot our password?" option on the site admin to set a password for this client account. Lastly, update the email address in the `'dh_users'` table to the client's email address.

You can add your site-specific styles to the `site/site.css` stylesheet. You might consider uploading any site assets to this site folder.

There are two files, `custom0.php` and `custom1.php`, showing a couple of ways to interface with the core system if more site customization is necessary.

## Widgets

todo

## Templates

todo

## Layout Engine

The core system processes layouts which may contain variables wrapped by braces, e.g. `{{YEAR}}` or `{{header_additions_{{TYPE}}}}`. In the case of `{{header_additions_{{TYPE}}}}` the first pass will replace `{{TYPE}}` with the page type: "index", "inner", or user-defined. This results in a tag such as `{{header_additions_index}}`. The 'dh\_data' table will then be queried for the 'header\_additions\_index' key.

Review the `replaceContent()` method in `core/includes/classes/Content/cContent.php` for a list of variables that are replaced. Values in 'dh\_data' may also use these variables. A first replace is performed before querying the database. After querying the database for values, a second replace is performed followed by a second query to replace any variables discovered during the first query. A final replace is then performed. Variables in the 'dh\_data' table that begin with "php" are evaluated using the PHP `eval()` function and must ultimately set a `$str` variable for replacement. Be aware of the scope for this PHP code. This code is evaluated in a method of the `cContent` class, so you have access to the methods and properties of `cContent`. Database methods can be accessed using `$this->cpdo->method()`.

## Wordpress

There is a base theme under `blog/wp-content/themes/digitalhill/` that should interface with the core system. Depending on the site design, this theme may need to be modified. You will likely need to add your blog override styles to `site/site.css`.

`blog/wp-config.php` has been modified to use the MySQL credentials in the `site config.php` file. `blog/wp-includes/template-loader.php` has been modified to generate a layout which is passed through the core system.

Modify the 'wp\_options' table, specifically the 'siteurl' and 'home' options.

## Magento

Rename the `magento/` subfolder with an appropriate name for the site or leave it as `magento/`. Visit the subfolder in a web browser to perform the magento install.

Replace line 87 in `index.php` with:

```
ob_start();
Mage::run($mageRunCode, $mageRunType);
$html = ob_get_contents();
ob_end_clean();
require_once(__DIR__."/../config.php");
require_once(__DIR__."/../core/config.php");
$content = new \Content\cContent($cpdo, null, "magento", 0, -1, "Magento Title", "Magento Keywords", "Magento Description", "", null, "inner");
```

```
$content->output($html);
```

This will force all Magento output through the core system allowing you to use core system variables. There are four example files: `1column.phtml`, `2columns-left.phtml`, `2columns-right.phtml`, `3columns.phtml`. These files show how to convert the Magento output into a layout for the core system. The original files are located under `magento/app/design/frontend/rwd/default/template/page/`.

To style the Magento admin using the Digital Hill theme, add the following code to `app/etc/local.xml`.

```
<stores>
  <admin>
    <design>
      <theme>
        <default>digitalhill</default>
      </theme>
    </design>
  </admin>
</stores>
```

The Digital Hill skin for the admin is located under `assets/magento/` in the repository. You will find that the Magento install includes a `skin/adminhtml/default/default/` folder. The Digital Hill theme should be located in `skin/adminhtml/default/digitalhill/`.

## Product Database

The following widgets are available within the product database:

- Type 0 - A basic input field.
- Type 1 - An upload field.
- Type 2 - A two-column simplified ckeditor.
- Type 3 - A gallery.
- Type 4 - A simplified ckeditor.
- Type 5 - An image uploader that retains a thumbnail of an image.
- Type 6 - A basic product tagging system.
- Type 7 - A system to assign related products.
- Type 8 - A one-column complete ckeditor.
- Type 9 - A grid editor with csv upload support.
- Type 10 - A system that supports multiple uploads.

The `'dh.configuration'` table includes two fields, `category_data` and `product_data`, that exploit these options. These are json-encoded fields such as:

```
[
  {
    "name": "title", "heading": "Category Title", "type": 0,
    "name": "image", "heading": "Category Image", "type": 5,
    "name": "full", "heading": "Description", "type": 4
  }
]
```

The `category_template`, `category_category_template`, `category_product_template`, `product_template`, `tag_template`, `tag_category_template`, `tag_product_template`, `search_category_template`, and `search_product_template` fields employ this data using the `name` attribute. An example `category_template` using the above `category_data` might look something like:

```
<div class="category-item">
  <div class="category-title">{{title}}</div>
  <div class="category-image">{{image}}</div>
  <div class="cb"></div>
  <div class="category-full">{{full}}</div>
  <div class="cb"></div>
  <div class="category-items">{{items}}</div>
  <div class="cb"></div>
</div>
```

where `{{items}}` is an element that employs `category_category_template` and `category_product_template` to list any subitems of the current category.

## Custom Admin Modules

todo