



## **Membership Website Platform**

### User Guide

## Table of Contents

Table of Contents.....	2
Overview .....	4
Welcome to Caribou CMS.....	4
Who uses Caribou? .....	5
Licensing Options.....	5
Receiving Software Updates.....	6
Transferring Licenses.....	7
Installation and Configuration .....	8
Installing Caribou CMS.....	8
Upgrading Caribou CMS .....	11
Server Migration.....	13
The Control Panel.....	15
Content Management .....	15
Subscription Management .....	19
Managing Your Storefront.....	22
Customizing Email Templates .....	24
Custom Fields.....	25
User Management .....	27
Affiliate System .....	29
Sending Newsletters .....	30
Your Design: Caribou & Templates .....	32

Templates Overview .....	32
Selecting a Theme .....	33
Modifying a Theme in the Theme Editor (Novice to Intermediate Users).....	33
Modifying a Theme via FTP (Intermediate to Advanced Users).....	34
Template Plugins.....	38
Caribou Themes and CSS Stylesheets .....	38
Creating, Managing, and Including Menus .....	39
Security.....	43
Integrations .....	45
Forum Integrations .....	46
Payment Gateway Integrations .....	48
Form Integrations .....	51
Shipping Integrations .....	52
Glossary .....	53

## Overview

### Welcome to Caribou CMS

Caribou CMS is a website publishing platform for revenue-generating businesses and organizations written in **PHP** and powered by a **MySQL** backend. At the heart of Caribou is a subscription membership website management content publishing system. However, subscription memberships are just one aspect of Caribou; the product has evolved to include other revenue-generating components such as online store functionality and affiliate system manager.

Like any innovation, Caribou's main purpose is to make things easier. Caribou makes things easier for businesses by:

- Accepting online subscription and store purchase payments via multiple payment **gateways** securely and automatically (see the Integrations section for a list of gateways),
- Managing all site content from one clean **control panel**,
- Managing site members, including seeing billing history, manually adding subscriptions to their accounts, searching your members database, etc., and
- Providing a great platform to be customized for any type of website – now made even easier with the Forms module, custom fields, integrations and, of course, easily editable and replaceable templates thanks to the Theme Editor.

Caribou's control panel is simple enough for non-technical users while providing the power and customizability that power users require.

For a more in-depth exploration of Caribou's features and uses, continue reading the Overview. For installation and configuration guides, technical help and introductions, skip ahead to the next section.

## Who uses Caribou?

Caribou CMS is used by a diverse selection of businesses and non-profit organizations. The list below highlights some of the more popular uses of Caribou:

- Selling and protecting members-only web content (articles, downloads, galleries, streaming videos, blogs, stores, etc.) with subscription access,
- Supplementing membership revenues with online store sales (even selling members-only products or giving members discounts),
- Managing public web content like blogs or tutorial pages,
- Earning revenue from web services through subscriptions,
- Selling downloadable products online and automatically (with email delivery and protected links), and
- Web designers providing clients with an online store and content management system all in one simple interface.

Caribou is essentially a collection of website components that can be combined and customized to create your website. Configuring Caribou may be done by the licensee or by a Caribou Professional. Before configuring Caribou, however, the user must select the type of license they would like to purchase.

## Licensing Options

Essentially, there are three licensing options available for getting started with Caribou CMS.

Don't think too hard about licensing options if you're just getting started. The free trial is automatically given to all users simply by installing the product. And, if you want to continue using Caribou after 30 days, upgrading is automatic. You won't even need to upload a license file – Caribou will do that for you.

First, the 30-day free trial option is available without any strings attached. You can either download the latest release of Caribou CMS as a .zip archive and install on

your own web server or shared hosting space or create a simple hosted free trial at <http://www.cariboucms.com/trial>.

The hosted free trial is best for non-technical users who want to try the control panel without worrying about breaking anything. Furthermore, you can see how managing your website's content, members, products, and settings is reflected in a default site design.

Second, the monthly paid license to Caribou is offered for a \$29.95 per month (USD) subscription. This option grants the licensee a single-domain license for as long as the license fee is paid. If cancelled, the license will expire 30 days after the last payment date. There are no license restrictions other than the single-domain access. Users are able to install multiple sites on one domain (does not include sub-domains). Monthly license holders are given free phone, email, and web support for as long as they remain subscribers.

Third, the lifetime paid license, is the most popular license for Caribou. A one-time fee of \$295 grants the licensee a single-domain license without any other restrictions. Support is available via phone, email, and the web for free for up to one-year after the purchase. After that, Caribou may charge for professional support though bug fixes will always be provided free-of-charge.

If you are experiencing a license error and the license error does not include the text, "ERROR-", but rather a code that looks like a license key, the problem may be a file permissions error. Delete `/misc/.license` and ensure that PHP has permissions to write to `/misc`. Reload your Caribou page (not the license error page) and see if it has been fixed. This problem often occurs after server migration.

## Receiving Software Updates

This software is actively developed and thus is often receiving product updates including bug fixes, feature add-ons, and usability enhancements. These updates are provided free to all licensees with certain restrictions:

1. If your version of Caribou CMS has been modified in a significant way (library modifications, core content modifications, core member modifications, etc.), updates may not be able to be applied as this will undo your customizations. A Caribou Professional can advise you as to whether your customizations will be upgrade-able prior to purchase.
2. Major upgrades (e.g. from 1.x to 2.x) are not guaranteed as often the software is so highly modified between versions that an upgrade process is not available out of the box.

These restrictions are blanket restrictions that are stated for complete transparency prior to investigating in a license. However, just as a Caribou Professional can do anything to customize your instance of Caribou to your needs, so too can a developer help you receive the updates that you want to your application. These updates may not come in a plug-and-play format, however, and may require paid development time.

### **Transferring Licenses**

Licenses can be transferred from one person to another once every 6 months at no charge. If a license is requested to be transferred more frequently than that, a fee may be incurred. Licenses may only be active on one domain at a time and so, immediately after transferring a license, the previous domain's instance of Caribou will expire.

In the event that the licensing server is unreachable by your server, your installation of Caribou will remain intact forever. Reliance on a licensing server in any other way would be irresponsible.

# Installation and Configuration

## Installing Caribou CMS

### **1) Requirements**

Before installing Caribou CMS, you will need to have a MySQL database created and privileges to access that database. This means that you need to know:

- Your MySQL username,
- Your MySQL password,
- Your MySQL hostname (often local host or the server IP address), and
- Your MySQL database name.

As for the web server in which you intend to upload Caribou CMS, it will need to have **IonCube** loaders installed. These can be downloaded and installed by visiting <http://www.ioncube.com/loaders.php>. If you do not know how to install these or are not a server administrator, contact your server administrator and request these. Installation is simple and secure – all web hosting providers should easily accommodate this.

Of course, your hosting must also include support for PHP. Caribou will make various checks on the server during installation and warn you if the environment is not sufficient (with specific details and suggestions to help).

Caribou also requires that your web server can read .htaccess file and has mod\_rewrite (or an equivalent) installed. This is how your URL's become search engine-friendly in Caribou. Apache is the web server of choice.

### **2) Download Caribou CMS**

New licensees should download the latest release of Caribou CMS from <http://www.cariboucms.com>. The download is a .zip archive containing a variety of folders, PHP files, CSS files, images, and HTML templates. In this download, you'll

find everything you need to get started with Caribou CMS (even a 30-day free trial license for a new domain).

Extract the contents of the .zip archive to a folder on your hard drive.

### ***3) Upload the files to your web server***

Connect to your server via FTP.

Most FTP clients are useable for uploading Caribou but, if you want a free option with the necessary options, try FileZilla at <http://www.filezilla-project.org>.

Set the file Transfer Mode to BINARY (Transfer > Transfer Type > Binary in FileZilla). Binary mode will keep the IonCube-encoded files from becoming corrupted during the upload.

Getting “file corrupted” errors when accessing your installation? This means that the files were not uploaded in BINARY mode. Re-upload /admin/index.php and /includes/library/main.php in BINARY mode to fix the problem.

### ***4) Access the Installation Wizard***

Once the files are uploaded, simply load up your web browser and access the /install directory at the location you installed Caribou. So, if you wanted to run Caribou at example.com, access <http://www.example.com/install>. If you wanted to run Caribou from a sub-directory of a domain, access <http://www.example.com/subdirectory/install>.

From here, Caribou will walk you through database configuration, site configuration, file permissions setup, and a server requirements check.

If you receive errors during the server requirements check that aren't fatal, they may be incorrect. Every server is different and, while Caribou does its best to sniff out versions and installed applications to check your environment, it can be wrong. The most common errors include Apache errors and mod\_rewrite errors.

### **5) Delete the Installation directory**

After the installation, all files in the `/install` directory *must* be deleted as they pose a serious security threat to your website and server. Caribou will allow you to access the control panel while these files still exist but an error will be displayed on your site's frontend until they are deleted.

### **6) Create the update cronjob**

Certain maintenance functions must be performed daily in order to keep your records accurate. The easiest way to perform these functions is to create a daily cronjob that executes `/updatecron.php`. This cronjob can be setup in any popular web hosting manager such as cPanel, Plesk, or Ensim. If a cronjob is not possible on your server, you can access Users > Update Subs in the control panel to perform the same maintenance functions.

Some people use a "php" call in the cronjob. This would look like this if it were to run at 3am every morning.

```
0 3 * * * php /home/example.com/public_html/updatecron.php
```

If that isn't working (and it doesn't always work), you can use the "wget" command to load the script as if someone accessed it in their browser, like so:

```
0 3 * * * wget -q http://www.example.com/updatecron.php > /dev/null
```

That code will retrieve the page (run the script) and place whatever output occurs from the script into the trash at `/dev/null`.

In the first example, an absolute path to the file is given. This is just an example path. Another common variation is `"/var/www/example.com/htdocs/updatecron.php"`. However, you should investigate with your web host what the absolute path to that file is. It will be the same absolute path as the `documentroot` setting for your Caribou installation.

## Upgrading Caribou CMS

### **1) Verify upgradability**

As mentioned in the previous section, Receiving Software Updates, there are technical reasons that may prevent (or suggest not) upgrading your site's version of Caribou CMS to the latest version. Please contact support if you are not sure about upgrading your instance of Caribou.

### **2) Backup files and database**

For peace of mind, safety, and a million other reasons, backup all files to your hard drive (and/or another drive) and download an export of your MySQL database prior to uploading new version files or running the upgrade wizard.

To backup your files, simply connect to your web server via FTP and download all of the files in your instance's folder to a new directory on your hard drive. This includes all uploaded files such as those in `/uploads`.

To backup your database, you have two choices. First, you can generate a backup SQL file through your Caribou Control Panel in Administration -> Backup/Restore. This file can then be downloaded via FTP from the `/misc/backups` directory. Second, you can login to phpMyAdmin or similar MySQL administration application and generate an Export of your entire database.

With these backups, the worst-case scenario for a botched upgrade is a few moments of downtime while you re-upload your old files and import the old database.

### ***3) Upload all files to your web server***

Connect to your server via FTP.

Set the file Transfer Mode to BINARY (Transfer > Transfer Type > Binary in FileZilla). Binary mode will keep the IonCube-encoded files from becoming corrupted during the upload.

Upload all files right over top of your existing files. Caribou will not overwrite your content.

### ***4) Access the Upgrade Wizard***

Access `/install/upgrade.php` in your web browser to begin the Upgrade Wizard. If your site is running at the root of a domain name, this folder will be at a location like <http://www.example.com/install/upgrade.php>. If it's in a subdirectory, this folder will be at a location such as <http://www.example.com/subdirectory/install/upgrade.php>.

### ***5) Delete the Installation directory***

After the installation, all files in the `/install` directory *must* be deleted as they pose a serious security threat to your website and server. Caribou will allow you to access the control panel while these files still exist but an error will be displayed on your site's frontend until they are deleted.

Upgrading your version of Caribou CMS will **not** overwrite your templates. New template files will be added to your install but changes made by the developers to the templates will not be transferred to your site. So, if you are experiencing a template-based problem, look at the new version's template and copy over the fixes from the new templates.

## Server Migration

Caribou is easily migrated from one web server environment to another. However, because the installation wizard cannot be run after the `/install` directory, certain procedures must be executed manually during a server migration and settings need to be modified. Follow the step-by-step guide below to easily migrate your instance of Caribou from one server to another:

1. Copy the entire Caribou folder to the new server. This can be done by download and uploading the files (in BINARY mode) via FTP or by zipping up the folder into an archive and using the \*nix command `wget` to download it to the new server where it can be extracted.
2. Export the Caribou database to a file and import this database structure and content into the new database at the new server. `phpMyAdmin` is good for both of these tasks.
3. If MySQL username, password, host, or database name have changed, modify `/includes/config.php` to reflect the new information. Note that all references to the connection information in this file must be changed.
4. If your document root location has changed (*and it probably has!*), run the following SQL command in `phpMyAdmin` or a MySQL command line interface to update it:

```
UPDATE `settings` SET
value='/your/new/absolute/path/to/your/caribou/directory/' WHERE
name='documentroot';
```

5. Delete `/misc/LICENSE` from the new server to prevent license conflicts.
6. Delete all sub-folders of `/includes/smarty/templates_c`.
7. Ensure that the following directories are writeable (CHMOD 0755 or 0777):
  - `/includes/smarty/templates_c`
  - `/includes/smarty/cache`
  - `/misc`
  - `/misc/backups`
  - `/uploads` and all sub-folders and files

- /templates
- 8.** Ensure that all server requirements are met by the new server (IonCube loaders, mod\_rewrite, cURL, allow\_url\_fopen = 'On' in php. ini , Apache, GD2 image library, etc.).
  - 9.** Create the /updatecron.php cron job on the new server to ensure your system is up-to-date.
  - 10.** Access your Caribou folder and test.

## The Control Panel

The control panel for your Caribou installation is located at `/admincp` relative to the directory you installed Caribou. It is secured by usernames and passwords. During the installation wizard, a default Super Administrator account is created and this will be your first login here. The Super Administrator can then add other staff accounts with varying levels of privileges.

Your entire website and business can be managed from this control panel. Here, you will:

- Add, edit, and delete content,
- Add new content types via backend forms,
- Manage your members (or add new ones), and
- Modify your Caribou settings, theme, etc.

Step-by-step guides for site management are not necessary as the control panel is very intuitive, includes contextual tips, and is hard to make irreversible errors in. However, this section will provide some insight into the basics of managing content and certain featured areas of the control panel that justify introduction and explanation.

## Content Management

### *Using Topics to Organize Your Site*

Topics are a simple way to organize your site in a hierarchal structure. Topics play a less important role with the Layout > Site Menus function now available (drag-and-drop your content into a site tree-style menu). However, they are important a) if you want to create topic homepages or blogs or, b) if you use breadcrumbs in your web design.

Essentially, topics serve four functions:

1. Generate breadcrumbs for content items automatically.

2. If created in Content > Site Topics & Blogs (and not from a content item), the topic home pages will serve as a blog of all of that topic's content postings.
3. Allow for inclusion of content into RSS feeds which include content based on topic.
4. Ease content management by showing more organizational data about each content item in the control panel.

If a topic is created from an Article / Page content item (by selecting *Create topic from article?*), the topic home page will be the page in which that option was checked. So, instead of seeing a blog or list of topic content, one would see that content page if they were to click the breadcrumb link.

### ***Page Information: Cross-Content Details***

When submitting any type of content, you will see a similar box in the form called "Page Information". This standardized box serves to unite all of the different content types you submit (articles, pages, forms, galleries, files, topics/blogs, and RSS feeds) in the database. Below is a summary of those fields and some noteworthy points regarding them:

- *Type* specifies a subtype of the individual content. For example, you may create a "Document" article sub-type which uses a different display template (a cleaner, lighter template, perhaps).
- *Title* specifies the page title.
- *URL String* is the location the content shall be placed at your website. This string must be URL friendly (and will be converted if unfriendly) so no spaces or special characters are allowed other than forward slashes (e.g. "/company/about-us/contact.html" is acceptable). Leave blank for auto-generation from *Title*.
- *Publish Date*, if modified, will postpone the publishing of the document to your public site. It will be available for editing/deletion in the control panel.
- *Summary* is a text description of the content. It is optional for articles but, if your article is being published to a blog topic, you should complete it as it

will be what's posted to the blog (the *Body* will be available when the user clicks the article link).

- *Feature Image* is an image that will be displayed in your content. It's rarely used in production sites but provides a quick and simple way to add an image to content.
- *Resize Feature Image* will resize the uploaded feature image to whatever dimensions you select. The dimension options are editable in Administration > Settings as the "imagesize" setting.

### ***Streaming Videos (FLV Format)***

You can add streaming videos to your website easily through the Content > Files section of the control panel. Caribou includes a Streaming Video (FLV) file format by default. Files submitted here will work with a special file download template that includes an FLV video player. No longer do you have to fumble around with JavaScript or Flash to get streaming videos on your website – just upload and go!

### ***Restricting Access to Usergroups***

Every piece of content, store item, and menu link has the ability to restrict its viewing/access to certain usergroups. This is the heart of Caribou's subscription system. You may select one or more usergroups to gain access to the content.

As will be repeated, usergroups do not equal subscription packages. Multiple subscription packages can upgrade a user to the same usergroup. This allows for greater flexibility in who can see the content.

### ***Forms: Manage Anything***

The Forms module in Caribou allows you to manage any type of content in the backend that you would like as well as receive form submissions in the frontend. The two types of forms are frontend forms and backend forms.

Frontend forms can perform simple tasks like Contact Forms or more complex tasks where users must submit data on your website. Form submissions can be emailed to an email address but, by default, they are stored in a database accessible via the control panel in Content > Forms. In the form's row, there is a link to view the current entries in the database. You can view and edit these entries at that link. To extend frontend forms into more complex things (such as having user's fill out profiles or submitting new site content) can be done with form plugins. Form plugins are discussed in the documentation and apply to backend forms as well.

Backend forms have one purpose and that is to create and store new content or database information. For example, you may want to create a form called "Books" and store a list of books there. With the database created in just moments and full create-update-delete functions handled automatically, publishing this database to your site is easy for a developer.

The simplest way to publish Forms-managed content is to create an article of a new article type with a *View Template* that points to a new template you create. In this template, call a template plugin which you must also create. This plugin can query the database and load the content to the article page's template. If you need to pass arguments to the plugin, have the template call the plugin and pass article variables like *Body* which will, in this case, include arguments (e.g. `{customplugin type=$body}`) will pass whatever you put in the article body to the plugin.

### ***Using the Content Queue***

If an administrator's current settings are to have all of their content queued before publishing, it will appear in the Content Queue. Only administrators who have privileges to review and accept/reject content from the queue will be able to make the content active on the site. These privileges are managed in Users > Staff.

## Subscription Management

At the heart of Caribou's membership website management capabilities lies subscription management. Subscription packages are recurring (either automatically or manually – see Gateway Integrations) fees that are charged at recurring intervals to a user until cancellation or expiry. Subscriptions can **promote** a user to a usergroup upon subscription and **demote** a user upon cancellation. For example, by promoting users to a group called "Subscribers", you can restrict content or products to subscribed users by making them only accessible to the Subscribers user group. Also, you could demote users to a group called "Expired Accounts" and target specific content or navigation links (such as a link that says "Renew Now!") to this particular usergroup.

### *Adding / Editing Subscription Packages*

Below are the options and fields one must consider when adding or editing a subscription package to their website:

- *Name* is the name of the subscription package. It's used in order forms, invoices, billing histories, etc., so keep it short and sweet.
- *Package Details* is a short description of the subscription package. Again, it's used all over your site so don't make this your primary sales text – just use it to give an overview of the benefits of subscribing.
- *Fee (in your Currency)* is the fee for the subscription.
- *Fee Recurrence (in Days)* is the time interval between recurring fee billings. For auto-recurring gateways, this will be billed automatically. For manual gateways, the user will have to have renewed their subscription manually some point between their last renewal and this upcoming expiration. If a user renews before the expiration date, Caribou will push back their expiration date by this number of days (basically, users can renew at any time).

Set this to “Package Never Expires” for a one-time payment subscription.

- *Automatically charge recurring subscription fee?*, when checked, will tell Caribou to use your payment gateways’ auto-recurring billing feature if it is available.
- *Recurrences* is the number of total payments to charge using auto-recurring billing (if available) before allowing the subscription to expire and require a manual renewal by the user. You may want to set this to 12 (for monthly subscriptions) to a) ensure your users still want the subscription and, b) get new credit card details once per year.
- *Free Trial Period* will make the package free for a set number of days. The user will not need to give billing details at the initial signup. The free trial is only available once per user per package.
- *Promotes User to Group* is the usergroup that you want to add to a user’s account upon subscription. They will be removed from this group at expiration. By using this feature, you can restrict access to certain site content and components by targeting a usergroup (or multiple usergroups).
- *Demotes User to Group* is the usergroup that you want to add to a user’s account upon cancellation/expiration.
- *Add-on Products* are products from your Store that will be suggested at checkout in a page like, “Would you like to add these to your order?”. These are a great way to cross-sell.
- The *Welcome Email* is an email that will be sent to your user upon initial subscription creation. They won’t receive this email for subscription renewals. You can use the site tags listed to personalize the email.

Each subscription package has its own link. If users access this link directly, they will not be prompted to select their subscription type in the checkout process. This link looks like `http://www.example.com/user/action/register/type/[typeid]`.

## ***Using Coupons to Provide Subscription Discounts***

Coupons are promotional codes that can be entered upon checkout for discounts on the subscription price. Caribou offers different types of coupons for subscription packages:

- *(Your Currency) Discount* is a straightforward coupon that subtracts a certain monetary amount off of the subscription fee for the first month.
- *% Discount* subtracts a percentage of the subscription off for the first month. This is a good code to use if you are offering a code for use across multiple subscription packages because the savings are relative to the total price of the package.
- *Free Trial (in Days)* will give a free trial to the subscription package for X days. Users will not be required to submit billing details at purchase – they will do this when the free trial expires.
- *One-Time Fee (in Your Currency) giving access to a specific date* will do exactly what it says. You can charge users a one-time fee for a package and all packages will expire on the same date. This is a rarely used coupon format.

Only one coupon code can be used per purchase. Furthermore, you can limit use of coupon codes when adding a coupon code to the site (restrict uses per person, per package, until a certain date, and to certain subscription packages).

If a user uses a coupon code, it will appear in their Billing History page in the control panel. You can also see a count of how many times each code was used in Subscriptions > Coupons.

## ***Adding Subscriptions to Accounts***

Most users will add a subscription to an account by accessing the subscribe component on your website and going through the checkout process. Once paid, the subscription is automatically added and you don't have to do any work.

You can also manually add subscriptions to accounts in the control panel. Using Member Search, find the target user. You can click the “Add Sub” icon for the user to add a subscription to their account. Or, if a user is active in the control panel, you can select “Add Subscription” from the drop-down menu. All subscriptions added in the control panel cost \$0.00. Users will receive notification of the subscription of the subscription and get a receipt for \$0.00.

## **Managing Your Storefront**

The storefront is a component of Caribou that allows you to sell non-subscription products online. These products can be downloadable products or shippable products. Downloadable products will be delivered via email after purchase. Caribou protects your downloadable products by sending a personal link to the purchaser. Furthermore, the user can only download the product twice with this link. Shippable products will, unlike downloadable products, require shipping information from the user upon checkout.

### ***Creating Your Store***

Before you start adding products and sub-categories, you must first create the initial store category. This category will be the parent category for the entire store. It determines the name to be used in your site’s navigation menu as well as the page that is displayed when someone accesses /store at your website. If you want to restrict access of your store to certain usergroups, you can specify the usergroup restrictions here as well.

### ***Creating Store Sub-Categories***

If you have many products, you may want to create sub-categories of the store that contain your products. Each sub-category will have its own listings page that will list the category’s sub-categories (if they exist) and products (if they exist). Furthermore, adding products below a sub-category will generate a breadcrumb with links to all parent categories.

## ***Creating Products***

As stated previously, products can be either **shippable** or **downloadable**.

Shippable products require a *weight* attribute (in pounds) and will force a user to give shipping information (name, address, and shipping method) when checking out.

Having trouble checking out when purchasing shippable products? Make sure that you have created at least one shipping method in the control panel at Storefront > Shipping.

Downloadable products are special and a unique feature of Caribou. These products don't have a physical weight or require shipping information from the user. Upon purchase, a personal link the uploaded file will be sent to the user. These personal links protect the file from direct access. Furthermore, each link can only be used twice before it de-activates. This protects your downloadable products from being shared.

Products can be restricted for purchase by users of certain usergroups. This way, you have members-only products. Furthermore, you can specify different prices for your products depending on the user's usergroup. With this feature, you can (for example) charge non-subscribers \$20 for your eBook but only charge \$15 to regular subscribers.

Product options are attributes of products like Color, Size, Version, etc. Product options are assigned to individual products upon addition or editing of the product. You can charge additional money for a product option.

If you have product options that require user input, you can create product custom fields in Administration > Custom Fields. These custom fields will be attached the product like product options. For example, if you need to know the school name of everyone who purchases your "School Dance" product, create a custom field called "School Name" and assign it to the "School Dance" product. When users add this

product to their shopping cart, they will be required to enter a school name. When the order comes through in the control panel, you will be able to see the school name you entered in Storefront > Orders and Users > Billing History.

### ***Store Specials***

Specials create a special sale price on an item (or all items in a certain category) until a specific date. Specials can be created in Storefront > Specials. Alternatively, you can add individual sale prices to products upon adding or editing an individual product.

Products can only be subject to one special at a time. So, if you add a special on all Computer products for 10% off and then add a special on all Computer Mousepads (a subcategory) for 20% off, computer mousepads will be discounted at 20% off. In essence, specials are not additive – the special added last will overwrite all previous sale prices on applicable products.

### **Customizing Email Templates**

Caribou interacts with your users and administrators via email frequently. Every email sent by Caribou, from user registration emails to subscription welcome emails to subscription receipts, can be edited in Administration > Email Templates.

When editing an email template, you will see a list of Available Variables below the current email template. These variables are replaced with data when they are sent. For example, `[[FIRSTNAME]]` will be replaced with “Paul” when sending to a registered user with the name Paul Smith. The `[[SIGNATURE]]` tag is an editable email template itself. Tags can be used in both the *Subject* and *Body* fields.

If you want to edit the name and email address of the sender of these emails, edit the *adminemailname* and *adminemail* settings in Administration > Settings.

## Custom Fields

Custom fields are one of many tools included in Caribou that allow you to tailor Caribou to your needs. There are five classes of custom fields that Caribou deals with. Each class is discussed below as they are all unique and implemented differently.

### ***Article Custom Fields***

These custom fields are seen when adding or editing Articles / Pages in the control panel. Validation doesn't apply to these custom fields (it's assumed that administrators will follow the rules).

Article custom fields are available in topic page templates or article page templates with variables of the format: `$articlecustom_[custom field name]`. For example, if you have a custom field called "Meta-Keywords", you would include this in your `header.html` template (which is included in the `article.html` template, by default) like so:

```
<meta name="keywords" content="{ $articlecustom_Meta-Keywords}" />
```

Also, if you want to have a custom field that has a link to an external site and call this field "External Link", you could include this in the `article.html` template like so:

```
{if $articlecustom_External_Link != ""}<p>Please visit this external site: <a href="{ $articlecustom_External_Link}">{ $articlecustom_External_Link}</a></p>{/if}
```

The code above not only uses the custom field but also only uses it in the template if the custom field is not empty.

## **Topic Custom Fields**

These custom fields are very similar to article custom fields except that they are seen when adding or editing topics. Validation doesn't apply to these fields because, like articles, they are for administrators only.

Topic custom fields are included in article/page templates (e.g. `article.html`) with code in the format of `$topiccustom_[custom field name]`. For example, if you have a topic custom field like "Author Bio" which holds the current bio of the main author in the blog topic, you can use this in any article or topic template like so:

```
<p class="bio"><b>About the Author:</b><br />{$topiccustom_Author Bio}</p>
```

Topic custom fields are include in topic templates (e.g. `topic.html`) in the format of `$custom.[custom field name]`. For example, if you have a custom field called "Video Link", you would use the following code:

```
Welcome to this topic! Click <a href="{ $custom.Video Link}">here</a> to access the video for this topic.
```

## **User Custom Fields**

Perhaps the most commonly used custom field class, user custom fields are fields that are added to the user registration form. All of the address elements in the user registration form by default are actually just custom fields (thus, they can be removed at any time).

Validation does apply to these custom fields and, if the form fields are not completed or completed improperly, the user will see detailed errors on the form and be prompted for correction.

Data collected in the custom fields is viewable and editable in the control panel when editing a user in Users > Member Search. If a field is marked as "*Editable in*

*the Account Manager*”, users will be able to edit this data in their Account Manager under Edit Profile.

### ***Affiliate User Custom Fields***

This class of custom fields follows the same rules as User Custom Fields except that they apply to users who are registering as affiliates.

### ***Product Custom Fields***

Sometimes, certain information is necessary from a user for an individual product purchase made in your online store. This class of custom fields solves that problem. So, for example, if you perform an engraving on a product and need to know what to engrave, you may create a custom field called “Engraving Text” and apply it to the product. Then, when someone adds this product to their cart, they will fill in this information. Custom field data gathered here is viewable in Storefront > Orders for each order and in Users > Billing History when looking at all subscriptions and orders for a particular user. Validation applies to these fields so users must complete the fields properly before the product is added to their cart.

### ***Gathering Other Information***

If you want to gather other information from your users but can’t find a way to make it work with custom fields, you can create forms easily in Content > Forms. Forms can even be extended with form plugins that manipulate and use the data submitted to do other actions. See the Integrations section for details.

## **User Management**

In Caribou, you can manage all of your users and staff right in the control panel. Here, you can do things like suspending users, editing user profiles or logins, adding new staff administrators who can access the control panel, or manage the usergroups which users can then be promoted or demoted to upon subscription/expiration.

## ***Managing Usergroups***

Usergroups work tightly with subscription packages to fulfill the subscription membership capabilities of Caribou CMS. Users must belong to at least one usergroup at any time.

Normally, administrators will create a usergroup called “Registered” that holds all basic site registrations (not necessarily subscriptions). This usergroup becomes the base level usergroup and rarely gives additional access privileges. Marking this usergroup as the “Default” user group in Users > Members & Groups will mean that all initial registers will be placed in this group.

After a user purchases a subscription, however, you may want to move them to another usergroup for subscribers only. In this case, you’ll want to create at least one more usergroup other than the default user group. Then, when adding a subscription package, mark this new subscribers-only usergroup as the usergroup to promote users to. Now, when someone subscribes, they will be placed in the usergroup automatically. Upon expiration or cancellation, they will be moved out of this usergroup.

## ***Managing Users***

You can add, edit, or suspend user accounts in the control panel at Users > Member Search. You can also perform more specific functions such as adding subscriptions to an account or viewing the user’s billing history.

To change the expiration date of a manually-recurring subscription, access the users Billing History and select “extend” beside the expiration data. Auto-recurring subscriptions do not have this feature; they have a “cancel” link which will cancel the subscription in both your Caribou system and with the payment gateway.

Users can be suspended or un-suspended at any time. Suspended users cannot login and they will be notified that their accounts have been suspended if they attempt to do so.

If you think that a user may be sharing login information with another user, you can check out the user's Login History via Member Search in the control panel. With the aid of hostnames and login times, you can investigate the matter.

### ***Managing Staff/Administrator Accounts***

To add more individual user accounts to your control panel (useful for sharing control or when there are multiple content authors), access Users > Staff in the control panel. Staff are assigned to staff groups that have certain privileges assigned to them. Furthermore, individual staff accounts contain certain content queue privileges. These queue privileges either force all content submitted by the staff member to be placed in the Content Queue before publishing or give them permission to approve/reject content in the queue, themselves.

Staff account information is accessible in content templates. This is useful if you want to include the author's name, email address, and/or biography with content that they have submitted. Access these variables in this manner:

```
Submitted by {$authorfirstname} {$authorlastname} (<a
href="mailto: {$authoremail}">email me</a>!)
/* other variables listed below: */
{$authorusername}
{$authorgroupname}
```

### **Affiliate System**

The affiliate system included in Caribou is a standard two-tier affiliate system that allows you to:

- Accept affiliate registrations (must be activated before they can login),
- Share personalized campaign media with links that track affiliate referrals,
- Automatically track commissions earned by affiliates, and
- Track payments sent to affiliates and the current amount owed to affiliates.

A two-tier system allows you to not only entice affiliates to sell your products/subscriptions for a commission but also to refer people to your site to become affiliates themselves. The 2<sup>nd</sup> tier commissions are paid to the affiliates who referred the affiliates that referred the paying customer.

### ***Setting Up the Affiliate System***

The affiliate system is managed in the Affiliates tab in the control panel. Before the affiliate system is activated on your site, you must check the box beside *Enable the affiliate system?* at Affiliates > Setup.

All settings in the Setup page include tips on how to use them properly.

After you've setup your commission settings, you should start creating Campaigns and adding media to the campaigns. This media will be provided to affiliates in the Affiliates Control Panel so that they can cut and paste links/code on their own websites to promote your products or subscriptions and earn money.

### ***Day-to-day Affiliate System Maintenance***

When your affiliate system is up-and-running, you will want to visit the Affiliates > Affiliates page in the control panel as often as you would like to pay out earnings to your affiliates. From this page, you can see the account balance on each user's account. For each user, you can also view the log of commissions earned, payment sent, and statistics on this affiliate's activity.

### **Sending Newsletters**

You can send newsletters to your website's members via the control panel. By default, Caribou includes a user option where user's can opt-in to your newsletter upon registration. When sending a newsletter, you can choose to send only opted-in users or to ignore this setting. Furthermore, you can choose which usergroups to send the email to.

Are your members not receiving their emails? Caribou uses proper email headers and PHPMailer to try and ensure the success of your email delivery. However, filters often distrust automatically-sent emails and may mark it as spam. The best solution to this is to encourage users to place your administration email (editable at Settings as “adminemail”) on their email server’s whitelist.

## Your Design: Caribou & Templates

### Templates Overview

When someone views any part of your Caribou website, the output that is sent to their browser (HTML, CSS stylesheets, JavaScript files, images, etc.) is 100% dependent on the templates system.

Modifying site designs quickly and efficiently is easily done by a Caribou Professional. Contact Caribou for a quote or for help with modifying the templates yourself.

An individual **template** looks like a standard HTML file but it also has programming markup in the **Smarty** scripting language. This programming markup is essential to Caribou as it allows for dynamic insertion of variables (such as “title”, “body”, “publish date”, “comments”, etc.). Furthermore, references within these templates can pull in data from other templates, template plugins (PHP files), and template functions (variable modifiers; ex. “strtoupper” which would make a string of characters entirely uppercase). This documentation will not cover Smarty functions and features. Those are available at <http://www.smarty.net>. An example template may look like:

```
{include file="header.html" title="Test Page"}  
<h1>{$title}</h1>  
<div class="articlebody">{$body}</div>  
{include file="footer.html" }
```

Templates are not individual pages for your website but rather files that dictate how many parts of your website will be displayed. For example, most themes (like the example code above) make use of a “header.html” file as well as a “footer.html” file to display content that wraps around all pages of your website. The header template may include such things as a logo, navigation, and a box for the page’s content. The footer template will close this box and likely include a footer with

website links, contact information etc. Editing either the header or footer template will be reflected across the entire website automatically as most templates (such as “article.html”, “form.html”, and “store\_product.html”) include both the header and footer.

A collection of templates is referred to as “*the* template” or the “theme” of your website. Caribou comes with a few default themes to choose from but it is highly suggested that you modify a theme before releasing your website to make it your own.

### Selecting a Theme

You are able to select the theme for your website during the installation wizard. Preview images are displayed for each theme and you can select the one that most closely reflects the design you want for your website.

After installation, you can change the theme of your website at any time in the Layout > Templates section of the control panel. Simply click “set default theme” and modify your previous selection.

### Modifying a Theme in the Theme Editor (Novice to Intermediate Users)

Caribou allows for quick and simple online editing of your website’s design in the **Theme Editor**. It can be accessed in Layout > Theme Editor in the control panel. This feature is excellent for day-to-day theme editing and also for web designers to offer their clients who may want to make small changes to the design without the source files or technical knowledge.

Administrators can use the Theme Editor to browse their website page-by-page as if they were a regular website visitor. Forms and links all work as expected. When you would like to make a change to the design of the page you are visiting, select “edit this template”. Caribou will load up the editable template where you can make instant changes to your website.

Furthermore, Caribou will link to included templates (such as the header and footer) and CSS stylesheets under “Included Files” to the right of the file editor. This is helpful because often the changes necessary to change the design are in CSS stylesheets or other template files. Caribou allows you to edit these easily.

Certain images are able to be edited directly within the Theme Editor, as well. Where this is possible, a link will appear in the top bar of the Theme Editor browser that says “upload new [image name] image”. The new image must be of the same filetype as the previous image.

Note: If you are editing a template in the Theme Editor or via FTP, you can mark images for quick upload in the Theme Editor by assigning them the class “editable”.

```

```

To upload a new image for inclusion in your templates, you can select “upload an image / asset” from the right sidebar whenever you are in the File Editor.

If you experience file errors when modifying your templates, make sure that all template files and their directories are writable by the PHP software and have the correct file permissions and owners. To do this, CHMOD all files in /templates to mode 0777 and make sure that the webserver user owns the file with the CHOWN command.

### **Modifying a Theme via FTP (Intermediate to Advanced Users)**

For most users, the Theme Editor will provide enough functionality for day-to-day modifications of the design. However, more advanced users may require greater control over the template files. Full control when editing templates is achieved by editing files locally on your computer and uploading via FTP.

This is simple for anyone who is comfortable with HTML and CSS as well as FTP. Each theme is completely self-contained in its own directory so one needs only to download the files from their current theme’s directory and upload new/modified

file to this directory to update the theme. All themes are located in /templates. This folder contains a directory for each theme as well as the following directories:

- /templates/common – This directory holds JavaScript framework files that would be redundant to have in each individual theme folder.
- /templates/default – These are the theme files for the administration panel. Editing is rarely if ever necessary.

### ***Template File Descriptions***

All filenames in released themes are self-explanatory, for the most part. Below is a table detailing the purpose of each standard template file (outside of the /core directory which contains files that are only for experts to edit because of essential template markup):

<b>Template</b>	<b>Description</b>
404.html	Displays the standard File Not Found page.
article.html	Displays an article or page. Variables include title, body, date, etc. Custom article fields can be accessed like articlecustom_[custom field name] or topiccustom_[custom field name].
error.html	Displays a system error.
file.html	Displays a file download page. Variables include title, summary, filesize, downloadlink, etc.
footer.html	Displays the bottom wrapper of each HTML page (by default).
form.html	Displays a frontend form (ex. contact form) as created/managed in Content > Forms in the control panel.
frontpage.html	The site's home page. By default, most Caribou site's are blog style and therefore the front page closely reflects a standard topic page.
gallery.html	Displays the main page for an image gallery with all thumbnail images of the gallery and other variables such as title and summary.
gallery_image.html	Displayed when the user accesses a single gallery image. By default, the gallery is sortable with Next & Previous links.
header.html	Displays the top wrapper for all pages (by default). Navigation is

	likely outputted here dynamically. Also includes a logo.
restricted.html	Displayed to a user upon a user permission error (ie. content has been restricted to a certain user group and this user is either not logged in or not in the user group). By default, this page simply loads the user_login.html template with a "restricted" error message.
results.html	Displays product and content search results from a search.
search.html	Displays an advanced search page with topic, publish date, and keyword fields.
store_cart.html	Displays a user's shopping cart.
store_category.html	Displays either the store frontpage (with all sub-categories and products) or a specific store category page (with all sub-categories and products).
store_product.html	Displays a single product page. Includes an "Add to Cart" form, product images, full description, etc.
thankyou.html	Displayed to a user upon successful purchase or free trial signup for both store purchases and subscriptions.
topic.html	Displays a main topic page. Often used as a template for blogs.
user_home.html	Displayed at the user's Account Management homepage. Includes all relevant account management links, a list of active subscriptions, etc.
user_login.html	Displays a login form as well as links for user registration or forgotten password retrieval.
xml_feed.xml	Displays an RSS feed as setup in the control panel in Content > Feeds.

### ***Template Variables***

There are thousands of variables available in the Caribou system and many are page-specific. To understand what variables are available for use in each template (e.g. when viewing an article, or displaying a topic page), use the following tips:

- Look at the existing template. Templates are generally very inclusive in their use of variables to show template editors what sort of data is available.
- Use the Theme Editor. Navigate to the page in the Theme Editor and select “edit this template”. On the right hand side will be a list of all variables available in the template.
- For content files, examine `/includes/class.content.php` to see the array returned for each piece of content. These arrays are passed on to the templates and can be used in anyway. For an example, check out the extremely simplified code snippet below:

```

class Content
{
    function Article () {
        // database and processing code would be here
        $array = array(
            'title' => $article['title'],
            'body' => $article['body'],
            'authorusername' =>
$article['authorusername'],
            'date' =>format_date($article['date'])
        );
        return $array;
    }
}

```

By examining this function, it's clear that variables “title”, “body”, “authorusername”, and “date” are available in the template. Your `article.html` template file may look something like:

```

{include file="header.html" title=$title}
<h1>{$title}</h1>
<div class="byline">Published {$date} by {$authorusername}</div>
<div class="body">{$body}</div>
{include file="footer.html"}

```

- For non-content items such as store items, use the above process to examine `/includes/class.store.php`.

## Template Plugins

As you examine `/templates` and its theme children, you will notice a `/plugins` directories in each theme. These directories contain standard Smarty template plugin files. All files in these directories are loaded automatically by Smarty.

Essentially, you can create functions that can be referenced in your template files. For example, the standard Caribou template function below will return a complete URL based on the location of your installation:

```
<?php
function smarty_function_url ($params, &$smarty)
{
    include_once("includes/library/main.php");
    return url ($params['url']);
}
```

Now, in your templates, you only need to call `{url}` with the proper parameter to output a properly formatted URL. Example template file:

```
Please <a href="{url url="test-page.html"}">visit this test page</a>.
{* link will be outputted as http://www.example.com/test-page.html *}
```

For more information on how to write or access Smarty plugins, visit <http://www.smarty.net>.

## Caribou Themes and CSS Stylesheets

Default Caribou themes (and themes created by a Caribou Professional) are heavily CSS-based. The HTML code is semantic and does not contain any styling whatsoever. By adhering to these standards, Caribou allows you to modify almost every single aspect of your website's design simply by access `/css` in your theme's

folder and editing the CSS stylesheets. While this documentation will not instruct on how to edit CSS, those knowledgeable in the field will surely see the possibilities and power in such a setup.

## Creating, Managing, and Including Menus

Caribou 2.0+ includes a menu creation and management tool right in the control panel. This tool is available at Layout > Site Menus in the control panel. You can maintain multiple menus within Caribou. The majority of sites will have one menu that runs throughout the entire site (default name: “main”).

### *Using the Menu Management Interface*

When the interface loads, you will be presented with two boxes. On the right is a box that lists all content (articles, forms, downloads, topic/blog pages, etc.). On the left are the current elements in their current order for the menu. Dragging content from the right box to the left box will add that content to the current section of the menu. Dragging an element from the left to the right box will remove that element (and all of its children) from the menu. They will be available to re-insert immediately. Elements can be ordered by dragging up and down in the left box.

The standard setup is for 2 potential tiers of navigation. This would look something like:

- Home
- Store
  - Golf Clubs
  - Polo Shirts
- Corporate Info
  - About Us
  - Contact Us

The starting view of your menu in the menu management interface is the first tier. To add or edit **children** of any link element, click the site map icon in the element's

row in the left box. Only standard content items can have children. Special elements (marked with a star) generate their own second tier navigation items. Also, external links aren't allowed children because when someone clicks them, they are taken away from the site.

### ***Link Options***

Both special elements and regular content items have link options that can be edited:

- *Display Text* is the text that is seen in the link. You may want to change this if the current text is too short or inaccurate.
- *Link* is the address the link points to. This is only editable for external links.
- *Viewable to User Groups* will only show the link to members of the selected usergroup(s) if not set to public.
- *Hide children unless this node is active?* will, when checked, not display the links in this element's sub-tier unless the user is viewing this element (or one of its children).
- *Hidden unless parent node is active* is related to the above. However, with this option, you can hide certain sub-links despite the parent link being set to show all children. So, you may want to show 3 main pages under your Documentation link but hide the 7 others. Only when someone is viewing the Documentation page or one of its subpages will all 10 sub-links be displayed.

### ***Using Menus in Templates***

By default, one menu (the "main" menu) is loaded in the templates. However, you can modify this inclusion or add more menus in any template you would like.

To include a user-generated menu in a template, use the following template code:

```
{menu load=" [menu_name]" }
```

That code will load a menu like:

```
<ul >
  <li class="active"><a class="active"
href="index.php">Home</a></li >
  <li class="parent"><a href="store/">Store</a>
  <ul class="child">
    <li><a href="store/cat/clubs">Gol f Cl ubs</a></li >
    <li><a href="store/cat/shirts">Shi rts</a></li >
  </ul >
</li >
</ul >
```

There are other options for the menu template function that can add a class and/or an id to the menu's main ul tag. You can use neither, either, or both in the menu call. For example:

```
{menu load="navlinks" id="nav" class="mainnav" }
```

The above template code will generate menu HTML with a surrounding ul element like:

```
<ul class="mainnav" id="nav">
  <!-- links would be here -->
</ul >
```

### ***CSS Styling of Menus***

As is exemplified above, the HTML that Caribou generates for menus is clean XHTML that is easy to style. To help with CSS styling of menus, a list of element class rules and markup rules are below. It may be easiest to simply use the menu manager and look at the final HTML yourself, though.

- All sub-links (children) are included in a ul element within the parent li tags.
- You can specify the class and ID of the main menu ul element through the menu plugin call (see above).

- All 2<sup>nd</sup> tier child ul elements have the class “children”.
- All 3<sup>rd</sup> tier child ul elements have the class “babies”.
- All 2<sup>nd</sup> tier child li and a elements have the class “child”.
- All 3<sup>rd</sup> tier child li and a elements have the class “baby”.
- All active elements have the class “active”.
- All elements with children have the class “parent”.

New to CSS? Or unfamiliar with how to turn unordered lists (ul 's) into menus? Start by setting the “list-style” attribute of the ul tag to “none”. This will turn your bulleted list into a row of text boxes with links.

### ***Using Custom Fields in Templates***

For more information on using custom fields in templates, check out the section on Custom Fields.

## Security

Caribou protects your memberships, products, passwords, product downloads, and site content (articles, pages, galleries, file downloads, etc.) in many ways. In this section, we'll examine the way your site is protected.

### ***Content is accessible only to specific usergroups***

You can restrict access to almost any component of Caribou to specific usergroups. There is no way for a user to get around this restriction without sharing login information with a subscribed user. However, you can track these users down by examining a user's Login History in the control panel. Users with expired subscriptions will be removed from the subscriber's user group and will no longer have access to your content. This restriction is server-side so it is impossible for users to use old-fashioned techniques to evade the security.

### ***File uploads are protected from direct access***

All file uploads, whether they are for downloadable store products or regular content items, are protected from direct access. Caribou takes the file you've uploaded and saves it under a protected filename. Then, through the .htaccess file that is used by your web server, Caribou denies all direct access to this file. The only way for users to download the file is to go through a proper Caribou link. These links then subject the user to the same restrictions as any content item. Product download links can only be used twice before they expire.

Unfortunately, if you upload streaming FLV files as content items in Caribou, there is no way to restrict users from using browser harvesting techniques to find the file and save it to their computer. These streaming files are impossible to protect because they must be directly accessed in order to stream the video immediately (instead of waiting for the entire thing to load). There's no way around this with current technology. However, users generally don't do this (many don't even know the technique for stealing videos exists).

### ***Passwords are stored securely***

At no time in Caribou will you see a user or administrator password. Why? Because those passwords are not stored as plaintext in the database. Instead, they are stored as password hashes. When a user attempts to login, the password they submit goes through the same hashing process and is compared to the database's password for verification. Why store hashes? Hashes cannot be reverse-engineered without an incredible amount of dedicated computer calculations. So, if someone were to get access to your database, they would be left with a lot of nonsense passwords and unable to recover the true passwords of you and your members.

### ***Credit card information is never stored in Caribou***

Storing credit card information sounds simple and like something that would make auto-recurring available for all gateways. However, there are many legal issues to deal with when storing credit card information. One of these issues is the issue of PCI Compliance. Essentially, it costs hundreds of thousands of dollars to verify that your storage techniques and business operations are PCI Compliant (and therefore compliant with MasterCard and Visa's security policies). So, with Caribou, your data is never stored. Auto-recurring payments work with your gateways own features. CVV2 numbers (those 3-4 digits after your credit card number) after *never* stored on your server in any fashion. This is done for security reasons and is required by major credit card companies.

## Integrations

While Caribou can do a lot by itself, its power is enhanced through **integrations**. Integrations are scripts/plugins that you install right in your control panel (Administration > Integrations) that enhance or extend Caribou's functionality, often through connections with other services or applications. Example integrations include forum software, payment gateways, shipping notifiers, and form plugins.

### *Writing Integrations*

Before integrations can be installed, they must be dropped into the `/includes/integrations/` folder. The filename of the integration must adhere to the following format guideline:

```
[integration name]_[integration type].php  
// ex: authnet_gateway.php  
// ex: phpbb_forum.php
```

Each type of integration (forum, form, shipping, and gateway) share three common functions that are used to install and edit the integration in the control panel. These functions are as follows:

- `int_[integration name]_[integration type]_install()`
  - *e.g. int\_phpbb\_forum\_install()*
  - This function adds the integration into the database along with a base64\_encoded, serialized array of its settings with default values.
- `int_[integration name]_[integration type]_update()`
  - *e.g. int\_authnet\_gateway\_update()*
  - This function is executed before each time the "Edit Settings" screen is loaded for the integration.
  - In most cases, it is left blank. However, sometimes it is useful to pre-edit the settings based on updates to the Caribou system. For

example, you may want to add recently-added usergroups to a setting's list of available options.

- `int_[integration name]_[integration type]_settings()`
  - e.g. `int_direct_shipping_settings()`
  - This is not a completely required function but rather a function included standard in all first-party integrations.
  - It takes the base64\_encoded, serialized array stored in the database and returns a PHP array.
  - Most functions in the integration will begin with a call to this function like so:

```
$settings = int_phpbb_forum_settings();
```

The rest of the functions in an integration depend on the type of the integration being written. Each of these functions is detailed below. It should be noted that the best way to learn how to write integrations is to look at the included integrations with Caribou.

Note to developers: Each integration function can only be passed one argument and it is passed as a reference. In the following sections, the keys of that argument (an array) are listed and extra details are given if necessary. Be wary of modifying the argument directly because, being passed by reference, changes you make to it will be passed on to the rest of the script. However, this can come in handy in some cases. More details on these situations are below.

## Forum Integrations

### ***Included Integrations***

Caribou includes two forum integrations out-of-the-box. You can connect and protect certain areas of your forums with Caribou if you are using phpBB3 or vBulletin 3. A comparison of the two integrations is below:

Capability	phpBB3	vB3
Automatic account creation in forums	Yes	Yes
User automatically logged into forums upon Caribou login	Yes	No
User automatically logged out of forums upon Caribou logout	Yes	No
Password changes are reflected in forums	Yes	Yes
Forum group membership synchronized with Caribou group membership upon subscription and expiration	Yes	Yes

If you modify your usergroups or forum structure, simply uninstall the forum integration from Caribou and re-install it. You will then see updated settings in the integrations manager. Not doing so may result in errors.

### ***Writing Forum Integrations***

To write a forum integration for Caribou, it is best to load a current integration file to use as a framework. All forum integrations must include the following functions:

- `int_[filename]_install`
- `int_[filename]_update`
- `int_[filename]_settings`
- `int_[filename]_login` (argument: array containing keys: `[username]`, `[email]`, `[password]`)
- `int_[filename]_logout` (argument: array containing keys `[username]`, `[email]`)
- `int_[filename]_register` (argument: array containing keys `[username]`, `[email]`, `[password]`, `[firstname]`, `[lastname]`, `[verification]`, `[signupdate]`, `[groupid]`)
  - `signupdate` is often "NOW()", the MySQL function
  - `verification` is the 5-character verification code necessary for verifying email addresses. As long as this code exists in the database, the user cannot sign in because they have not yet verified it.
- `int_[filename]_password` (argument: array containing keys `[username]`, `[email]`, `[password]`)

- int\_[filename]\_subscribe (argument: array containing keys [username], [package])
  - *package* is the ID of the subscription type and should correlate with a setting that specifies which group is connected with that type
- int\_[filename]\_expire (argument: array containing keys [username], [package])

## Payment Gateway Integrations

Gateways connect Caribou’s ecommerce functions to your merchant account. Most gateway providers also provide you with a merchant account and you pay a monthly fee on top of a transaction fee. The two types of gateways that are important to Caribou are *merchant accounts* that allow you to accept credit cards directly on your site and *3<sup>rd</sup> party gateways* such as PayPal that require users to leave your site momentarily to complete their purchase. Adding a payment gateway is normally very easy. However, Caribou includes many gateways to choose from right out of the box.

### ***Included Gateways***

Caribou includes a variety of merchant account and 3<sup>rd</sup> party gateways for you to choose from.

The main thing of note for each gateway is whether or not the gateway integration supports **auto-recurring billing**. Auto-recurring billing is a special feature of some gateways that allows you to charge subscription payments automatically each month. Caribou will be notified of each successful payment and the user’s subscription will be renewed automatically for as long as payments are made.

Name	Type	Auto-Recur?	Notes
Authorize.net	merchant	Yes	Must use Silent URL Post feature.
First Data (formerly	merchant	Yes	User cancellations and expirations must ne handled manually by the administrator.

LinkPoint)			
InternetSecure	merchant	No	Despite being a merchant account, users are required to complete purchase off-site.
NetBilling	merchant	Yes	Must configuration Scriptable Reporting Interface, Control Keyword, and Dynamic Security Code.
Offline	offline	No	Accept cheques and money order payments with this. Invoices can be marked as paid in the control panel. Only when an invoice is marked as paid will the order / subscription be active and appear in the control panel and user's account manager.
PayPal	3 <sup>rd</sup> party	Yes	Uses Instant Payment Notification and subscriptions for easy integration.
PSiGate	merchant	No	Great for Canadian merchants.
Sage Pay (formerly Protx)	merchant	No	Great for UK merchants.
2checkout	3 <sup>rd</sup> party	No	Similar to PayPal.

### ***Writing Gateway Integrations***

Gateway integrations can be simple non-recurring scripts or involved, auto-recurring scripts with multiple callbacks. Below is a list of all functions that are required or available in payment gateway integrations:

- `int_[filename]_install`
- `int_[filename]_update`
- `int_[filename]_form` (argument: *gateways* array reference)
  - Add an array with the following values to the *gateways* array to list this payment method in checkout:
    - *name* – equivalent to the integration filename
    - *displayname* – the method name to show in the drop-down

- *type* – either “creditcard”, “creditcard\_uk”, or “redirect” (for PayPal and Offline, etc.)
  - *cctypes* (if *type* is “creditcard” or “creditcard\_uk”) – array of accepted credit cards (with the name as the key, display name as the value)
  - *text* (if *type* is “redirect”) – a string of text to display upon method selection in drop-down
- `int_[filename]_validate` (argument: *gateway* class reference)
  - The *gateway* class is passed, but you must run the `Create()` method to initialize the gateway with the proper name.
  - Then, call `AddVariables()`, `AddRequired()`, and `CheckRequired()` in that order to validate the submission.
- `int_[filename]_process` (argument: *gateway* class reference)
  - To accept the transaction, use the `Accepted(message, invoice_id (optional), is_subscription (optional))` method. Passing an *invoice\_id* will mark that invoice as paid. Passing *is\_subscription* as TRUE will keep the invoice active and awaiting other auto-recurring payments. *invoice\_id* is required if *is\_subscription* is TRUE.
  - To reject, use the `Rejected(message)` method.
- `int_[filename]_callback` (optional)
  - For 3<sup>rd</sup> party transactions, the callback is where `Accepted()` and `Rejected()` will run.
- `int_[filename]_expiring` (optional) (argument: *sub* array containing data for the subscription expiring as well as key [*invoiceid*] which is the ID of the invoice linked to the expiring subscription)
  - array includes all package database invoice as well as *invoiceid* which is the corresponding active invoice.
  - This method is really only for auto-recurring subscriptions that don't have a pushing gateway – we must pull the data from the gateway.
- `int_[filename]_cancel` (optional) (arguments: array containing keys [*invoice*] and [*id*])

- *id* is the subscription database ID
- *invoice* is the corresponding invoice.
- This function is for auto-recurring subscriptions only. Manual subscriptions won't even see this button.

## Form Integrations

Form integrations are plugins that take form data and manipulate it or utilize it *after* it has been submitted to the database.

### ***Writing Form Integrations***

There are 3 potential methods in form integrations and they are all passed the same arguments:

- `int_[filename]_save` (argument: array containing keys [*form*], [*values*])
  - *form* is the ID of the current form
  - *values* is an array of the submission values
- `int_[filename]_edit` (argument: array containing keys [*form*], [*values*])
  - Note: Form rows are only able to be edited in the backend.
- `int_[filename]_validate` (argument: *validated* array as reference)
  - Identify which form was submitted by using `form_get('form_id')`. This returns the numeric ID of the form.
  - Using `form_get_all()`, you can retrieve all of the variables submitted to the form (e.g. "field10", "field11", etc.).
  - If you find an error in one of the fields, simply add a key/value pair to `$validated` in the form of:

```
<?php
$validated['fieldname'] = 'Error text.';
```

## Shipping Integrations

Shipping integrations are activated upon sale of an item that a) has a shipping method and is using shipping, b) is using the shipping *handler* specified in the integration. For example, an integration file called `ups_shipping.php` will be activated upon each sale using the “UPS” shipping handler (but not a potential handler called United Postal Service). There is only one method available per shipping integration:

- `int_[filename]_sale` (argument: array containing *all order variables from \$gateway->variables*)
  - The order variables passed include the associative arrays *type* (all subscription data) and *user* (all user data) as well as an array of *products, total*, etc. Examine any gateway integration for a full list of variables that are being added to the gateway.

## Glossary

### **auto-recurring billing**

This feature of certain payment gateway integrations allows you to automatically charge for subscription renewals. If this feature isn't available on a gateway, users are required to pay manually (at any time) to renew their subscriptions.

### **backend or Control Panel**

The control panel for your website where you manage all aspects of the site. Multiple administrators can access the backend but all aspects of it are password-protected for obvious security reasons.

### **blog**

A blog is a collection of content submissions with the latest posts first. A blog can be created in Caribou by creating a topic in Content > Site Topics & Blogs and then submitting content to this topic.

### **children**

In the context of menus, children refer to sub-links of a parent link. So, the parent "Documentation" may have the children, "Installing", "Configuring", and "Licensing".

### **demote user to group**

An option for subscription packages, users will be added to this usergroup after a subscription cancellation.

### **downloadable products**

These products available in your online store do not have weight attributes or require shipping information from the user upon checkout. You will have to upload a file when adding the product. A personal link to this download will be emailed to

the user upon purchase. This download can only be used twice before it de-activates.

### **frontend**

The “public” section of your website. This includes all content (both public and members-only), your online store, user account pages, etc. The frontend isn’t controlled directly but rather through the backend or control panel.

### **gateway**

A payment gateway is a method for accepting payments online. Gateways can be either merchant account systems such as Authorize.net, NetBilling, ProTx, and First Data where credit cards are used to make payments without users leaving your website or they can be 3<sup>rd</sup> party processor accounts such as PayPal or 2checkout where users temporarily leave your site to make payment. Caribou tracks all payments (and non-payments).

### **integrations**

An integration or plugin is a script that can be installed through Caribou’s control panel and connects Caribou to other applications or services and/or extends Caribou’s functionality. Example integrations include phpBB forum software, Authorize.net, Paypal, and First Data payment gateways, and UPS shipping notifications.

### **IonCube**

A source protection system for PHP. By encoding two files in a Caribou installation, the licensing system is protected.

### **MySQL**

A secure database storage system. While looking directly at a MySQL database file or output is meaningless to most users, Caribou uses PHP to access your site’s MySQL database and uses it to store all of its data and content.

## **PHP**

A server-side scripting language that is extremely common for web applications because of its speed, universality, and ease of use. Web servers and server administrators provide support for PHP – users don't need to manage it whatsoever.

### **promote user to group**

An option for subscription packages, users will be added to this usergroup upon successful subscription. They will be removed upon cancellation.

### **shippable products**

These products in your online store will have a weight attribute (in pounds) and require shipping information from the user upon checkout.

### **Smarty Template Engine**

A powerful, open-source template engine built for PHP applications. It separates the display (templates, HTML, CSS, images, etc.) from the application and therefore allows for greater control of your website's design without interfering with programming logic.

### **streaming video**

Streaming videos are becoming a key feature of many content websites. You can add streaming videos (FLV files) to your website through Content > Files.

### **template**

An HTML file that includes Smarty template markup. Each potential link or component on your website references a template to be displayed to the browser.

### **Theme Editor**

A component of the control panel, the Theme Editor allows you to browse your website as if you were a regular visitor. When you see something you want to edit,

you select “edit this template” and Caribou will show you the editable template and link to included templates and CSS stylesheets. Save changes and view your modifications immediately.