Movable Type

Six Apart
Movable Type

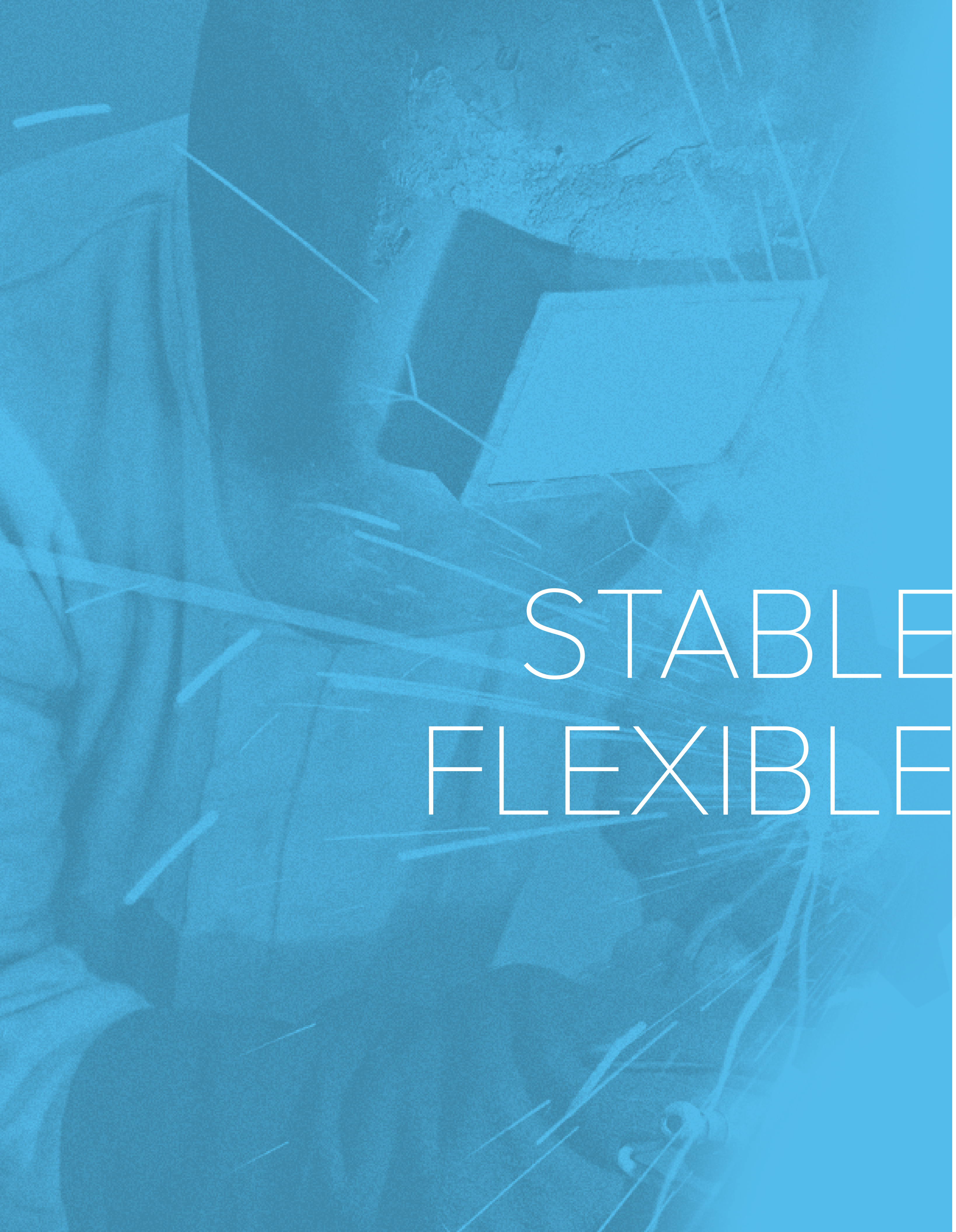
Movable Type

MAKE YOUR WEB PRESENCE FELT WITH MOVABLE TYPE. WE MAKE IT SIMPLE TO CREATE AND EASY TO MANAGE YOUR CONTENT.

WHY CHOOSE MOVABLE TYPE?

Publishers large and small love our professional publishing platform for engaging audiences and building brands.

Movable Type pioneered the blogging and content management revolution in 2001 and continues to provide digital solutions for small business and enterprise clients.



STABLE FLEXIBLE

We pioneered enabling users to flexibly manage content without sacrificing the strengths of static publishing. More than a decade later, the track record proves static sites consistently handle more traffic with fewer computing resources.

Building your website with Movable Type is easy. The template markup is about as simple as HTML, and the resulting page code is completely in your control. Further, templates can be structured as you see fit using intuitive includes to eliminate repetition and reduce error.

SCALABLE
EXTENSIBLE

With Movable Type, you get stability, flexibility and security along with a robust content management framework. Below are some of the parts that work together to make managing your website a great experience for everyone involved.

Most content management systems consume ever more processing power and memory as your site grows, degrading user experience, your bottom line, or likely both. Movable Type supports unlimited users to create as many websites and blogs with as much content as you desire without affecting page loading times.

At the core, Movable Type is a powerful content repository with sensible structure that has withstood the test of time. From custom fields and themes to plugins and the Data API, there are many possibilities for developers of all abilities to achieve their goals.

Dashboard

Google Analytics
Display website and blog access information in the Site Stats widget.

Blogs and Websites at a Glance
Get a quick overview of the blogs and websites to which you have access with quick links to create new entries and other daily administration tasks.

Personal Stats
Counts of and links to your own entries, pages, comments and drafts to keep tabs on your progress.

Movable Type News
Keep abreast of Movable Type developments with this automatic news feed from the software developers.

Content Management

Multiple Blogs
Organize your data however you see fit by creating as many blogs as you'd like, each with its own publishing options. And we mean as many blogs as you'd like. They're unlimited!

Multiple Websites
Unlike other systems, multisite support is included! You can use Movable Type to create just one website or thousands!

Entries and Pages
Each website and blog supports both entries and pages so you can manage your content in a way that makes sense to you.

Search Engine Friendly URLs
Static publishing means your URLs are inherently search engine friendly. No URL fancy rewriting necessary!

Multiple Categories and Folders
Entries can be assigned to any number of categories, and pages placed under any number of nested folders, allowing your content organization to reflect your business.

Entry and Page Versioning
Specify how many previous versions of entries and pages to save on a per blog basis so you can revert changes and have an audit trail where it is important, and save resources where it is not. Each version can have a comment to describe the changes.

Scheduled Publishing
Entries and pages can be set to automatically go live at a specified date and time, enabling you to streamline your publishing process.

Scheduled Expiration
Entries and pages can be set to automatically be unpublished at a specified date and time, enabling easy management of sponsored content, contests and more.

Feed Based Syndication
The default themes contain a readymade ATOM feed template so your content will automatically be available to feed readers and other services. Movable Type's infinitely flexible templating allows you to customize the feeds any way you'd like by adding custom fields and other data, or even by changing the format entirely.

Search Engine and Log
Use Movable Type's built in search engine to allow users to search for content on your site. All searches are logged to the activity log for later review.

Pagination
Easily link to previous and next entries within or across categories.

Entry, Page, Comment Pagination
Split long entries into multiple pages by easily inserting page breaks using the Pagination plugin.

Ping Servers
Notify ping services of website updates automatically, configurable per blog.

URL Names
Entries and pages have configurable basenames to allow customization of the resulting URL.

Author Experience

Multilingual
Out of the box support for English, German, Dutch, Spanish, French and Japanese.

Autosave
Entries, pages and templates are saved automatically to prevent loss of work after unfortunate events like a system crash, power outage or accidental browser closing.

Preview in New Window
Ability to preview content in popup windows.

Bookmarklet
Post entries and pages using convenient bookmarklets.

Batch Editing
Save time by modifying status, title, primary category and date information for multiple entries or pages simultaneously.

Moblog Publishing
Using the PostOffice plugin, save or publish text and images to your blogs by sending an e-mail, allowing for easy on the go publishing or accepting contributions from specified or anonymous users.

Default Status
Control whether entries and pages are published or set to draft status by default on a per blog basis.

Default Text Format
Control the default format of entry and page bodies on a per blog basis. Options include rich text, Markdown, basic line break conversion and none.

Default Fields
Select which standard fields are visible by default on entries and pages.

Rich Text Editor Styles
Customize CSS styles used in the display of content in the rich text editor by pointing to a remote file or a file within the current theme.

Symbol Replacement
Set whether punctuation and other symbols are replaced with their HTML character entity or ASCII equivalents for backward compatibility with databases and other systems that do not support UTF-8.

Custom Fields and Media

Extend Any Object
Add custom fields not just to to entries and pages, but also to comments, assets, media files, categories, folders, blogs, websites and users.

Custom Template Tags
Change the template tag used to access the data for a given field to anything you would like.

Required Fields
Make custom fields required or optional.

Default Values
Set default values for any custom field if desired.

Multimedia and File Uploading
Custom fields support file types including images, audio, video and generic files.

Image Resizing and Thumbnailing
Create thumbnails of any size. Movable Type supports four major image manipulation protocols: GD, NetPBM, Imager and ImageMagick.

Asset Manager
See at a glance all files in the system to edit metadata, update the files and see where they are used.

Data API

Create Content
Save new entries, comments and assets via external programs via the Movable Type Data API.

Read Content
Access existing entries, comments TrackBacks, user accounts, blogs, websites, categories and even site statistics via external programs via the Movable Type Data API.

Update Content
Update existing entries, comments, TrackBacks and user accounts via external programs via the Movable Type Data API.

Delete Content
Delete entries, comments and TrackBacks via external programs via the Movable Type Data API.

Theming and Publishing

Tag Modifiers
Dozens of functions can be applied to the output of any template tags or sections to do things like escape for XML, URLs or JavaScript, do find and replace operations and many other text processing functions.

Link to File
Save templates to files, allowing integration with file based versioning or backup software. Linked template files can also be edited directly instead of through the Movable Type interface.

Template Versioning with Notes
Specify how many previous versions to save on a per blog basis so you can revert changes and have an audit trail. Each version can have a comment to help understand the changes.

Background Publishing
Templates can be published on demand or sent to the publishing queue to improve the user experience.

Dynamic Publishing
Templates can be set to dynamic mode, where the system will generate them on the fly.

Automatic Backups
Never lose work! Savw existing templates with a single click when applying a new theme or refreshing the templates from an existing theme. A special listing screen allows you to view and restore any backed up template and see the date it was saved.

Template Access Control
User roles and permissions can be used to only allow designated users access to templates on a per blog basis.

Template Exporting
Use the export theme function to produce an archive of all of your templates for backup or distribution.

Template Cloning
Easily create new templates based on existing ones.

Page Modeling
Index templates have a one to one relationship with published files.

Listing Templates
Archive templates can be used to create many published files that correspond to lists of entries or pages organized by date, category, folder or author. You can have separate templates for each of these or even use a single template for all of them. This enables simple creation of concepts like category or monthly archives.

System and E-mail Templates
Easily change the code used for system functions like user registration, commenting and more, including the content of e-mails sent by the system.

Template Includes
Template modules are templates that do not directly correspond to published files, but rather can be included within other templates. This enables reusing code, such as for headers, footers or sidebars, or even just breaking large templates into smaller pieces that are easier to manage.

Template Widgets
Create sets of widgets, which are like template modules but can be rearranged by dragging and dropping. This can allow for visually designing a sidebar, for instance, allowing blog managers to pick from a set of existing widgets to design their respective pages.

Parent Website Module Includes
Template modules can be placed within a website such that all child blogs can access them.

System Module Includes
Template modules can be placed at the system level such that all child websites and blogs can access them.

Native File Includes
Movable Type can incorporate the contents of any given file during the publishing process, useful for integrating with external applications.

PHP Includes
Set template modules to use PHP style includes on a per blog basis.

Apache Server Side Includes
Set template modules to use Apache style includes on a per blog basis.

Active Server Page Includes
Set template modules to use ASP style includes on a per blog basis.

Java Server Page Includes
Set template modules to use JSP style includes on a per blog basis.

Blog Level Cache Control
Enable or disable template module caching at the blog level and on a per blog basis.

Template Level Cache Settings
Control the cache expiration for each template module individually, based on age or based on creation or modification of comments, users, TrackBacks, folders, pages or assets.

Community

Automatic Blog Creation
A personal blog can be created upon user registration if desired. Options include setting the blog theme, default language, default time zone and the website the blog should be created under.

Authentication Options
Users can log into your site using their existing membership with any OpenID compatible service, including Google and Yahoo!.

User Registration
Movable Type includes a complete user registration system so you can build your own community to drive engagement.

System Level Toggling
Enable or disable comments and TrackBacks at the system level.

Blog Level Toggling
Enable or disable comments and TrackBacks on a per blog basis.

Spam Handling
Option to automatically delete spam comments after a certain number of days and to control the aggressiveness of the spam filter.

Comment Link Cleansing
Automatically apply a “nofollow” attribute to links in comments so your site’s search ranking is not affected by the sites linked to in comments. Additionally, trusted commenters can pass through this filter if enabled.

Comment Moderation
Set whether comments are immediately published for all users, authenticated users, trusted users or no one.

Limit Comment HTML
Disallow HTML in comments entirely or allow only certain tags.

E-mail Notification
Control whether e-mails are sent to blog editors when comments are submitted or when moderation is required.

Advanced Notifications
Using Notify Who?!, control exactly who should receive entry, comment and TrackBack notifications for each blog. Options include notifying the entry author, arbitrary e-mail addresses, a special blog address book or any combination of these. For entry notifications, you can configure the plugin to send them automatically or to simply provide defaults for the Share entry screen. Automatic notifications can be disabled on a per entry basis by toggling the link directly above the entry save and preview buttons.

Auto Link URLs
Transform URLs in comment text into HTML links.

Text Format Control
Comments can be treated as plain text or processed through the convert line breaks or Markdown filters.

CAPTCHA Support
Use supplied CAPTCHA images to protect against spam comments and user registrations. Optionally, the reCaptcha plugin can be used to integrate with that popular service.

Comment Confirmation Page
Redirect users to a special confirmation page or opt to return directly to the entry or page right after a comment is submitted.

Anonymous Recommendation
Control who can use the recommendation feature.

Upload Control
Set the destination for files submitted via the community pack’s “create entry” form on a community blog or forum.

Entry Junk Filter
Set whether entries posted by users who do not have permission to publish entries on a community blog or forum are run through junk filters for comments and scored for spam.

Architecture

Website Separate from Database
Static publishing means users can still access your website even if the database goes down.

Structured Programming Language
Movable Type’s core is written in Perl, the legendary programming language that is more structured and secure than PHP.

Multiserver Support
Extremely high traffic websites can benefit from Movable Type’s support for many server setups. You can publish from a main app server to many front end servers, or even spread some of the system’s functionality across any number of servers.

Many Operating Systems Supported
Movable Type supports Windows Server and OSX as well as Debian, Ubuntu, CentOS, Red Hat Enterprise Linux and any other similar operating system.

MySQL Support
All versions of Movable Type support MySQL databases.

Oracle Support
Movable Type Enterprise supports Oracle databases.
MS SQL Server Support
Movable Type Enterprise supports Microsoft SQL Server.

Roles and Permissions

Role Based Permissions
Define a variety of user types, each with a distinct set of permissions. Making Movable Type match your business structure is easy!

Fine Grained Permissions
Assign to each user any number of roles that can differ by blog and website. A user can be an administrator for one blog, but only be allowed to moderate comments on another.

Newly Created User Settings
Customize the roles that are automatically assigned to users upon registration on your site.

Security

Password Validation Rules
Set the password policies, such as minimum length and requiring a combination of uppercase and lowercase letters, requiring a combination of letters and numbers and requiring special characters.

User Lockout
Set thresholds for locking out users who attempt login with an incorrect password a certain number of times within a certain period.

IP Address Lockout
Set thresholds for locking out IP addresses that are used to attempt logins with an incorrect password a certain number of times within a certain period.

IP Address Whitelist
Prevent trusted IP addresses from being locked out. This is helpful for preventing one user in an office from locking out the company’s own IP address.

IP Banning Manager
See and control all banned IP addresses.

Site Path Limitation
Set a systemwide base site path to prevent any sites from being published outside a given root directory.

System Comment Toggle
Disable comments for all websites and blogs in one place.

System Trackback Toggle
Disable receipt of TrackBacks for all websites and blogs in one place.

System Notification Pings Toggle
Disable notification pings for all websites and blogs in one place.

System User Registration Toggle
Disable new user registration for all websites and blogs in one place, and set designated users to receive notifications upon user registration.

Publishing Controls

Basename Length
Control the length of automatically generated names used in URLs.

Entry and Page Listing Controls
Limit an entry listing to a certain number of entries, pages or days, control the order, set the number of words in automatically generated excerpts and control the date format according to language.

License Chooser
Set an explicit Creative Commons license per blog to easily inform your users how your content can be used and shared.

File Path Control
The system can publish files to anywhere on your system you want, including outside the designated blog root if desired.

Website URL Control
For extra security and convenience, you have the flexibility to run Movable Type at a different domain name or subdomain than is used to access the published website.

File Extension Control
Set the default extension for published entries, pages and listings per blog, or customize it for individual templates.

Preferred Archives
Designate as primary one of multiple templates that publish entries or pages for different platforms or formats. This allows control of the file pointed to by the entry or page’s permalink used throughout the site.

Empty Category Archives
Enable or disable publishing of category archives that contain no entries.

Tools

Listing Framework
Most listing screens are powered by the Movable Type listing framework, which enables quick sorting by any of the column headings, powerful custom search filters and display options that include controlling the number of items displayed and the columns displayed.

Periodic Tasks Script
To better facilitate background publishing, Movable Type includes a script that can be called with a scheduled task to handle publishing items in the queue and other maintenance tasks. This script can also be run in continuous daemon mode or limited to a certain number of processes.

Listing Filter Manager
Manage at the system level all saved listing framework filters and see who created them and when. Click the filter names to jump right to that particular search.

System Find and Replace
Movable Type includes a powerful search and replace function that will help you find exactly what you seek within entries, pages, comments, TrackBacks, templates, assets, users, blogs or the activity log. You can limit your search to specific fields, date ranges, and for advanced users, you can even use regular expressions.

Page Not Found Detection
Find and fix broken inbound links to your website by logging all 404 errors with the Clean Sweep plugin. Users get redirected to a helpful error page, while you get to see at a glance the “top attempted” wrong links, which you can then choose to automatically redirect to the proper page.

Plugin Manager
See all plugins and their settings at a glance and enable or disable individual plugins at the system level.

Entry Import and Export
Easily migrate your data into or out of the system or between systems with the basic entry export utilities.

Theme Export
Create a theme package from your current templates, categories, folders and custom fields so you can easily deploy your current theme on another site or distribute your work to others.

System Backup and Restore
Make a backup archive of your entire system and restore from a backup from within the administration screen.

Activity Log
See all recorded activity in one place, including user registrations and logins, failed login attempts, search records, information about comment, entry, page, blog, website and template creation, error messages and more.

System Information Screen
Get the total number of users and commenters and information about the server environment for informational purposes and to assist with troubleshooting.

System E-mail Test
Set the e-mail address used by the system and send test e-mails to verify functionality.

Debug Mode
Developer friendly debug modes can be enabled to display basic debug messages, stack traces, database queries issued, template build time reports and more.

Static Publishing vs. Dynamic Publishing: What’s the Difference?

Static Delivery

When Tim Berners-Lee invented the World Wide Web in 1989, there were no web content management systems. The first website consisted of simple text files written in the newly created HyperText Markup Language. Users requesting the pages were delivered copies of the files by web server software. From that point, web pages have become more complex, but that first delivery system still exists today, and we know it as “static” publishing.

At its simplest, static web publishing involves creating an HTML file for each web page address, or URL. The files are saved on the web server in folders that correspond to the URL structure of the website.

Static file delivery is the fastest possible method because the web server only needs to read the file from disk or memory and send it to the user. No other processing is required.

Dynamic Delivery

A CMS can also separate content from design. The system generates the final HTML by inserting the content into predefined templates. These systems can take many forms. Some use content stored in files, whereas others store content in databases. Some generate HTML files at the time of content creation or on some other schedule, and other systems generate HTML at the time the user requests a page.

That last idea of generating HTML “on the fly” is what we often call “dynamic” publishing. This method usually requires the web server to process all the code required to build the final HTML from the theme files and the content itself before anything can be served to the user. Dynamic publishing therefore requires much more processing power and memory than static publishing.

If you manually create HTML files for each web page, you are definitely using static publishing. If you use a system in which the HTML delivered to a user does not exist until the user requests it, and only then it is generated by the CMS, you are definitely using dynamic publishing. But most large websites occupy an ocean of gray area in between these two extremes.

Static publishing can employ a range of dynamic techniques

Movable Type by default generates full HTML files for each page within the website, making it a superb choice for websites that require the fast page loading and high performance possible with static HTML delivery. Using Movable Type’s intuitive template system to build your website allows you to modularize code for easy reuse, eliminating the problem in traditional static publishing where changes must be made in many files. Instead, changes can be made in one place, and the CMS can automatically regenerate all the HTML files.

Even with such static publishing, dynamic elements can still be incorporated to enhance the user experience.

For example, the main pages such as the home page, section pages or individual content item pages can be statically generated, while other pages that depend on user input can be dynamically generated. Movable Type uses this strategy for functions like displaying search results, user registration and commenting. Those pages are created on the fly by the CMS, while the majority of the site’s pages are published as static files available for instant delivery.

Moreover, using PHP or other programming languages, dynamic elements can be incorporated into static files.

Dynamic publishing and caching

While a CMS that generates all HTML from scratch upon each user request would be considered to be purely dynamic, that CMS would also be much more resource intensive and slower than is probably necessary.

To compensate, most dynamic publishing CMSs use some sort of caching to prevent the need to generate the same HTML over and over. Each system can have its own ways of doing this, and performance can vary by the website. As a simple example, a website with 10 pages is much more likely to be easily cached than a website with 10 billion pages. From there, as the number of page elements customized for each user or request increases, the less helpful caching will be in making the system performance approach the speed of a static publishing system like Movable Type.

Decoupled Content Management

Content management on the web has evolved substantially over the years. The first websites were manually edited, static HTML files. Over time, more advanced functionality and automation were added, resulting in the rise of the content management system. Many of these systems are highly dynamic systems that handle all aspects of the website: content creation and management, data storage and content delivery. Such systems are what we call highly "coupled" systems, which specifically refers to the degree to which management and delivery of content are connected.

Coupled systems have become popular for many reasons. They make getting up and running easy, as everything you need to run a website is in one package. Many systems are available for free and aim to satisfy a large market by cramming as many features as possible into the system. And as popularity of these systems has increased and their user communities have grown, there are more options for support, leading to even more adoptions.

But as larger and busier sites have been built with such systems, many weaknesses of the coupled model have become apparent.

- Handling more traffic can be complicated and involve adding replicated database servers, as well as multiple application and front-end servers. This means a greatly increased cost of ownership.
- Having one system handle delivery, data management, and storage, means a problem in any piece of the system can cause the entire website to fail. For instance, a database problem can prevent any page of the site from loading.
- Since the CMS is exposed on the front-end, security concerns are a major drawback. A successful attack could not only result in defacement, but also deletion of all data, since all the parts are connected.
- Site administrators are stuck with the content delivery model provided by the CMS, which is usually limited to a single web view that does not take into account pushing content to multiple channels.

The solution to all of these problems is to separate the monolithic systems into independent and replaceable parts. Architectures like this are known as "decoupled" systems. The primary distinction between coupled and decoupled systems is that in coupled systems, content is delivered to end users by the same system in which content is managed and edited, whereas in decoupled systems, content management is completely separate from content delivery. Sometimes the data storage layer can even be independent of the content management layer.

Decoupled systems resolve all the problems mentioned above to various extents. CMSs like Movable Type that publish static content can completely solve the data security problem, as the CMS can be firewalled and hidden from public view. Another benefit of the static publishing method is increasing traffic capacity involves only increasing front-end server resources and doesn't necessarily require touching the application or database servers. Further, a database problem will not affect the content delivery as it would in a coupled, dynamic system.

Finally, a decoupled system could push content to one or more channels in a generic format such that front-end developers can work on delivery without any knowledge of the CMS. This enables separate development teams with tailored expertise to focus on each aspect of the system. Front-end developers can work on the web design and user interactivity without worrying about content storage or management. Application developers can enhance author experience and add productivity features independently of the front end, so long as the data is output in a consistent format. There can even be multiple front-end teams for different channels, such as for handling content being pushed to mobile devices or even sports stadium Jumbotrons. Decoupled systems can therefore increase productivity and lower costs while enabling the flexibility to deliver content for the 21st century.

www.moveabletype.org

Movable Type