Migrating a Website to Drupal

Smoothly Transfer Content, Users, Interface, and Metadata

Drupal's advantages for running enterprise websites are well established by such companies as FedEx, The Economist, and The White House. But many who would like to use it hesitate because they fear loss of assets now managed by their current site's platform, or expect the procedure to transfer those assets to be prohibitively difficult.

This paper examines scenarios for migrating from other content-management systems (CMSes) and web applications, such as Vignette, Jive, FatWire, ColdFusion, and Joomla[1]. It evaluates reasons for migration, describes migration methods and tools, and suggests best practices.

Executive Summary

Websites become harder to maintain as they age. First, evolving web trends demand capabilities that older technologies don't have. Second, enterprise data and site structure become messy, calling for a reorganization. Third, business factors (such as the availability of qualified consultants) may make the legacy system uneconomic to maintain.

Many organizations seek to address all three problems by migrating their sites to a modern CMS. But standing in their way are uncertainties about the migration process, among them:

- How can we best plan the migration?

- Will we be able to transfer all our data, including users, tags, and metadata?

- Do the reasons for migrating justify the costs?

---

[1] Migration procedures vary depending on the source CMS' back-end database. Some notes for migration from specific CMSes are at http://drupal.org/handbook/migrating.

This paper answers such questions, particularly as they relate to migrations from such common enterprise CMSes and web applications as Jive, Vignette, FatWire, ColdFusion, and Joomla , as well as from sites built in static HTML.

The target CMS is Drupal, which is free and open source software. Drupal, which powers more of the world's top 10,000 websites than competing CMSes[2], has emerged as a leader because of its open architecture, high-quality code base, and extensive selection of third-party support options. It additionally has migration utilities for both one-time moves and ongoing integrations with legacy systems.

We'll take you through the entire migration process, from surveying assets to full implementation. In particular we'll look at techniques to manage data transformations to actually improve your assets during the migration.

Quotes without footnote references were gathered in interviews.

Why Migrate?

Websites rarely stop functioning all of a sudden. Beyond obvious failures like bad links, subtler problems abound. You might discover that a new feature is hard or expensive to add, for example, or notice a decline in site visitors. Business reasons, such as the promise of lower costs or better search-engine optimization (SEO), may also drive migration.

Some of the most common reasons include:

Freedom from a single-vendor solution. No matter how good a proprietary CMS is, it's a castle built on sand if it makes you dependent on a single provider. The vendor could abruptly stop developing it in favor of another of its products, as Microsoft did to Content Management Server 2002 so it wouldn't compete with SharePoint 2007[3]. Or the CMS' direction may change when the company that created it gets acquired, as was the case with SiteFlash and NewsFlash when Adhesive Software disappeared. Similar situations have affected products by Alterian, Oracle, and Open Text[4].

2   http://www.backendbattles.com/Content_Management_Systems
3   http://www.microsoft.com/cmserver/letter.mspx
4   http://www.jboye.com/blogpost/vendors-kill-products-and-make-customers-pay/

"[FatWire] was proprietary, so we were locked into the vendor's roadmap and features. We would have been with any proprietary solution, really."

— Publishing company Chief Information Officer (given on condition of anonymity)

To "bet on a stronger horse". One problem common to small and shrinking CMSes is the lack of available consultants. A healthy developer ecosystem requires two things: software design that allows outside parties to develop for it, and a user base big enough to incentivize consultants to learn it. Only a handful of CMSes even come close to fulfilling these requirements. Where the developer ecosystem doesn't exist, you're dependent on the CMS vendor for even the simplest changes.

"Once we knew that our CMS product [Collage] was going to be mothballed, it was time to move on. Utilizing unsupported commercial software wasn't going to work. We realized this was our opportunity to reassess and execute on a long-term plan that would solve some of our long-standing issues."

— Nicholas Maloney, Web Architect, Bentley University (bentley.edu)[5]

To take advantage of new web technologies. Every day brings new versions of software and services that affect how you deliver information on the web. There are changes on all levels, from server applications (such as Apache) to third-party services (such as Google Calendar) to infrastructure software (such as PHP). Only CMSes with an active developer community can hope to keep up with — and take advantage of — all the changes.

"Single sign-on APIs were one pain point [in Jive SBS]. We also played around with its social functions at some point, but realized we couldn't develop fast enough with it."

— Alex Kirmse, Zappos IP, Inc. Senior Front-End Development Manager

To follow social trends. Internet technology is important, but it's all done with people. Audiences are constantly discovering and embracing new behaviors, such as microblogging and content tagging, or moving en masse from one social network to another. If you can take advantage of these changes, you'll not only improve visitor satisfaction: You may also gain valuable community-created content. CMSes that fail to help you follow these trends risk leaving you out of the next popular sensation.

One example is Facebook[6], the wildly popular social-networking site that's become a visit-me-first internet portal for millions of people. It's possible to pipe content from your site to Facebook "pages" that appeal to that site's members, thereby greatly extending your reach. But writing code to take advantage of

---

5   http://acquia.com/community/resources/library/bentley-university
6   http://www.facebook.com

Facebook's Application Programming Interface (API)[7] can be difficult — unless your CMS provides assistance. For Drupal that assistance comes in the "Drupal for Facebook" module[8], a free download that greatly simplifies the connection between your site and Facebook.

Paul Chason, Managing Partner of Acquia partner Mediacurrent, said in a webinar that one client moved from their proprietary CMS because it was weak on social-networking features the client needed, including "commenting, tagging, and user-generated content[9]."

To integrate multiple systems. Acquisitions and other business changes often require merging data from diverse sources in one place. Some CMSes play well with others; some don't. Migrating to a CMS that understands various data sources can allow some legacy systems to stay in place while delivering a unified presentation to site visitors.

> "Part of the motivation to move the existing content over to Drupal was to escape the rigid complexity and cost associated with the Vignette CMS. The Vignette dataset was a 1.66GB Oracle database — and that didn't include the more than 15,000 images referenced in the Vignette data which also had to be imported into the new site[10]."
>
> — *Laura Scott, President and Creative Director at Drupal consultancy pingVision[11]*

To accommodate growth. As a website grows above thousands of pages and millions of hits per month, small performance issues multiply. That's especially noticeable in older or home-grown CMSes, as they were typically built without modern traffic loads in mind. But even some "enterprise" CMSes fail under high loads, simply because they've never been tested in such strenuous real-world situations.

> "We had performance problems with [Jive], which seemingly couldn't handle any kind of traffic. It could have been a misconfiguration on our part, but those issues disappeared when we migrated to Drupal."
>
> — *Alex Kirmse, Zappos IP, Inc. Senior Front-End Development Manager*

To reduce costs. There are several good open-source CMSes available for free, and yet some CMS vendors still charge tens of thousands of dollars in licensing fees. That makes sense only as long as buyers feel they're getting equal value from the CMS in terms of functionality or support. Anyone who doesn't is likely to seek migration to another CMS.

---

7   http://developers.facebook.com/docs/guides/web
8   http://drupal.org/project/fb
9   http://acquia.com/community/resources/acquia-tv/best-practices-migrating-legacy-based-cms-
      drupal
10   From a case study at http://drupal.org/popular-science
11   http://pingvision.com. Quoted in "Popular Science Magazine (PopSci.com) Case Study", http://
      drupal.org/popular-science

"We'll be saving hundreds of thousands of dollars in the first year alone [after migrating from ColdFusion], just on slashing proprietary web server licenses[12]."

— *Mike Meyers, Chief Technology Officer of Clarity Digital Media (examiner. com)*

These reasons for switching raise an important question: If your current CMS isn't cutting the mustard, which one would? For many enterprises, the answer is Drupal. The next section details how Drupal addresses each of these common shortcomings.

Why Drupal?

Since its creation in 2001, Drupal has become a leading CMS among enterprises. One automated measurement[13] lists it as the number-one CMS among the world's top 10,000 sites, powering (among others) nowpublic.com, ubuntu.com, and crackberry.com. Another found that Drupal powers more than one percent of the top million sites[14]. Approximately 500,000 sites in total run Drupal, and people download the Drupal software from Drupal.org about a quarter million times per month[15].

Drupal is licensed under the GNU Public License, ("GPL"), which means that no fees of any kind are required to deploy, extend, or maintain Drupal. This license additionally guarantees[16] that your organization will always be free to run, study, modify, and redistribute Drupal.

"Acquia Drupal[17]" is a popular alternative to "core Drupal" that also includes several of the software's most popular extensions, or "modules". Like Drupal itself, it's licensed under the GPL and shares its freedoms.

But these facts don't explain why Drupal has achieved such success among high-visibility, high-traffic, mission-critical sites. Simply put, Drupal solves the problems that make organizations seek a new CMS. Specifically, **Drupal is not ruled by any individual vendor**. As an open-source project, anyone can use and modify Drupal's code, which is entirely in unencrypted plain-text files. Its API is

---

12   Quoted in "Revolutionizing The Online Media Market Through Open Source Social Publishing", http://acquia.com/resources/webinars/peek-behind-scenes-%E2%80%93-how-clarity-media-moving-examinercom-drupal-7
13   http://www.backendbattles.com/Content_Management_Systems
14   http://www.drupal.org/project/drupal
15   "The State of Drupal" keynote, http://sf2010.drupal.org/conference/sessions/state-drupal, at approximately 25:00. Summary at http://www.ojctech.com/blog/drupalcon-2010-keynote-highlights.
16   Excerpted from "The Free Software Definition", http://www.gnu.org/philosophy/free-sw.html
17   Details at http://acquia.com/products-services/acquia-drupal; downloadable from http://acquia.com/downloads

well-documented[18], and its system of "hooks" makes it infinitely extensible. As a result, anyone with an intermediate knowledge of PHP and a willingness to learn Drupal's API can effect changes; if your organization doesn't have such skills in-house, many qualified service providers are available to help you[19].

Like other open-source projects, Drupal has no absolute authority — a fact that makes some people nervous. Project founder Dries Buytaert exercises strong guidance, controlling "official" releases that appear on the Drupal.org website. (Mr. Buytaert is also a founder of the Drupal support company Acquia.) But typical problems of single-vendor products, such as vendor lock-in and poor extensibility, simply aren't possible with Drupal.

Drupal quickly adapts to new technologies, social trends, services. Where Drupal doesn't address a need, individual developers create modules to fill the gaps — typically, within days or even hours of the appearance of new, complementary technologies. Over 4,000 modules are currently available for Drupal 6, the most-popular of which are incorporated into Drupal 7.

Distributions are another way developers deliver Drupal-based solutions to address emerging needs. A distribution is an installable package that combines Drupal, supporting modules and design files (called themes), and custom programming for a specific solution or purpose. Dozens of distributions[20] let administrators quickly set up sites that are optimized for, among other things, publishing[21], intranets[22], and high-performance content delivery[23].

Drupal scales to accommodate growth. Companies that produce proprietary software have long promoted performance and scalability as two of their main selling points. (Microsoft and Oracle have so targeted Linux and MySQL, respectively.) But criticisms of Drupal on these bases have been widely disproven in the field. One notable example is found in Wikimedia's fundraising system, a Drupal-powered sub-site that successfully served 20,000 requests per second[24]. Proving Drupal's scalability is http://www.akademika.no, which as of October 2008 had almost five million pages and sold almost three million products[25].

---

18   http://api.drupal.org
19   http://drupal.org/drupal-services
20   http://drupaldistrowatch.com is one site that tracks Drupal distributions.
21   OpenPublish (http://openpublishapp.com) and Managing News (http://managingnews.com)
22   Open Atrium (http://openatrium.com)
23   Pressflow (http://pressflow.org)
24   http://fourkitchens.com/blog/2006/12/19/four-kitchens-builds-wikimedias-new-fundraising-
         system
25   http://drupal.org/node/320616

Several consultancies specialize in optimization and tuning for high-demand Drupal sites, among them Four Kitchens, which produces the Drupal distribution Pressflow.

For the popsci.com migration, pingVision's Laura Scott noted that the publication had "a large and active user base" that they expected to keep growing. After migrating away from Vignette 7 CMS, the site achieved record-high loads of 60 pages per second, including one period of over 1.1 million page views in 24 hours.

Drupal has a long projected life. Over its ten-year lifetime, Drupal has grown from the contributions of thousands of developers, writers, and designers. Version maintainers, who are appointed by project founder Dries Buytaert, release maintenance versions to address security issues and minor bugs approximately once a month; major ("integer") versions have historically appeared about once every two years. (Drupal 6 was released in February 2008 and is the current version as of May 2010. Drupal 7 will probably be released during the summer of 2010.)[26]

The project officially supports the previous and current integer versions, i.e. currently Versions 5 and 6. As a result, When Drupal 7 is officially released, Drupal 5 will no longer be supported by the Drupal project, although private, commercial support is commonly available.

Every release version of Drupal has a simple upgrade path for site data. By comparison, several of the competing CMSes in our survey have undergone at least one revision whose upgrade path was either difficult or that required the purchase of new licenses.

> "[When we used Jive,] we'd ask, 'How do we fix this,' and Jive would say, 'You have to upgrade to the newest version'. We'd ask, 'Well, how do we upgrade to the newest version,' and they'd say, 'There is no clear upgrade path'."
>
> — *Alex Kirmse, Zappos IP, Inc. Senior Front-End Development Manager*

Drupal costs less. As stated earlier, Drupal costs nothing to acquire or run. By comparison, a typical first-year license for Jive costs $59,000; a similar license for FatWire costs $120,000, while one for Vignette costs a whopping $400,000[27]. Costs for licenses and support beyond the first year are likewise far lower for Drupal, amounting to an overall year-to-year cost reduction of as much as 90 percent, which is usually enough to quickly cover migration costs.

26   http://fourkitchens.com
27   Assumptions from which these figures were calculated, and other details, are in the white paper "TCO for Open Source Social Publishing: Going Beyond Social Business Software", http://acquia.com/community/resources/library/tco-open-source-social-publishing

Part of Drupal's lower costs come from its greater availability of resources, which results in favorable market forces. The same study that compared license costs among various CMSes also found that developers with Jive-specific knowledge cost considerably more than those with Drupal-specific knowledge — and were in considerably shorter supply.

Taking Stock of Your Current Site

Two types of questions face you when you decide to migrate your site to Drupal (the "target site") from your current system (the "source site").

First are the **organizational questions**:

- What assets comprise the source site?

- What parts will you migrate?

- Do you want to match the source site's look and functionality exactly, or will you make improvements during the migration?

- Does the target site need to integrate with legacy systems that can't be moved, such as customer databases or third-party data sources?

Answering these questions early will help you answer **technical questions**, which include:

- What sort of repository does the source site use to store data?

- How does the source site structure non-content data such as user profiles, taxonomies, and such metadata as creation dates?

- What tools are most appropriate for a fast and error-free migration?

- What procedures will migrate the site with minimal downtime and maximal data access?

The following section exposes details behind these questions by examining the elements that comprise your current site, discussing their management in Drupal, and giving tips to avoid common migration issues.

Collecting Site Assets

To know what you have, you first have to know how and where it's stored. Assets usually exist in three places: As records in a database, as discrete files on a known server, and as elements stored in a place or format outside of your control.

Most CMSes use **database records** to varying extents to store text content, non-text content such as graphics (as Binary Large Objects, or BLOBs), settings, and metadata. (Some use databases in limited ways to store operational code as well: In Drupal, for example, a piece of content in the database could include PHP code.) The main issue with transferring database information is that table structure varies from one CMS to another. For example, user profiles might be stored in a single table in one CMS, but in multiple tables in another.

Discrete files may contain multimedia such as graphics and movies, downloadable assets such as white papers, or any other kind of content that's not stored in a database or external site. They could even contain the site's main text content, if the site was created in the "traditional" way as a set of HTML files. (That's usually the case for sites created using Adobe Dreamweaver.) Transferring these files to the target site is easy by simply copying them. The hard part is telling your target site where to find them, as file paths almost always change in the transfer.

External site assets such as mapping data are often easy to transfer. They're typically embedded in the source site by reference, for example using iframe or object tags. In such cases all that's needed is an identical reference in the target site. The offsite asset remains the same, unaffected by the migration. Sometimes, however, continued access to those assets requires migration of additional data (such as an API or certificate key), changes to the agreement with the asset's third-party provider (authorizing a new target URL, for example), or modifications to the conduit that delivers the information (such as a specialized database driver).

Types of Site Assets

Regardless of how your CMS structures asset storage, the assets themselves are usually similar from one CMS to another. Knowing a bit about these elements — and how they're stored in Drupal — will help you decide policies for their migration.

Text content

What it is: Text content includes all articles, blog posts, pages, and other information formats intended for display. Most CMSes store text content as records with at least a Title field and a Body field. Additional fields are common: For example, an e-commerce site might store product information as text content with fields such as price, size, and quantity available.

Further, most CMSes attach meta-information fields to each piece of text content. Examples of such information include a "key" that's unique to each record, the record's creation date, a link to the user who created it, access controls, and so on. One important class of metadata comprises tags that categorize content: CMSes often use these tags to aggregate similar content into (for example) topic pages.

In Drupal: Drupal calls records of such text content "nodes", and stores them in a set of database tables that begin with node. The record's primary key is stored in the node table, while the content itself is in the node_revisions table.

Drupal distinguishes the purposes of various text content — separating blog posts from catalog pages, for example — through the use of "content types", stored in the node_type table. The Nodetype module[28] provides limited capabilities to move content from one content type to another.

> "I start with a universal content type that includes such things as a developer notes field, and CCK fields to track errors. We did a large, messy HTML import for a client. There was no way we could tell what content types their HTML data should map to. So we just built an "Import" content type, cloned it about 30 times, and ran the import. Then we said to the client, 'You guys are going to have to go in as editors and transform these things into the types of content they're supposed to be.' We were then able to start customizing the content types."
>
> — *Ken Rickard, Senior Engineer, Palantir.net*

Content types contain a defined set of fields, including the default Title and Body field mentioned above. Each node of a particular content type comprises the same fields. In Drupal 6 you can add other fields using the Content Construction Kit module[29], commonly known as CCK; in Drupal 7 you can add fields to content types without any additional software.

It's possible to create text content records in Drupal with a series of SQL commands, whether done manually or driven by a program. Creating a record properly might affect a half-dozen database tables, though, so the potential for error is substantial. Further, you might need to merge or split tables, depending on how they're stored in the source site. Therefore, content transfer is usually done via a combination of general data-manipulation and Drupal-specific tools, as will be discussed in the "Migration Tools" section.

---

28   http://drupal.org/project/nodetype
29   http://drupal.org/project/cck

Issues: One common problem occurs when text content hasn't been "cleaned" of code such as HTML, PHP, or other markup tags. Fortunately, Drupal filters content to allow no PHP and only a small subset of HTML by default through its system of "input formats" (called "text formats" in Drupal 7). It's still a good idea to determine whether such unclean text exists on the source site and determine whether it would be easiest to correct on the source site, on the final Drupal site, or during migration.

> "Is the legacy content formatted in a specific way? Are paragraphs separated by new lines? Does Drupal need to display your legacy content in a rich-text format on the new site? If it does, the migration method will need to replace the new lines with <p> tags.)"

> — *Paul Chason, Managing Partner at Drupal consultancy Mediacurrent*[30]

A related issue arises when text content includes poorly formed HTML or unexpected characters. This commonly occurs when eight-bit text (such as curly quotes) moves from a format that understands them to one that doesn't, or moves among operating systems that differ in how they handle control characters such as line feeds. Automated tools such as HTML Tidy[31] can catch some such problems, but you should also at least spot-check text content manually.

The source of text content varies tremendously, adding potential complication. In an Acquia webinar[32], Senior Engineer Ken Rickard of Palantir.net talked about a "continuum of pain" for source data in Drupal migrations. Content in MySQL or PostgreSQL is the easiest to move, as those are formats that Drupal already understands natively. (Drupal 7 adds a database extraction layer that may lead to other "native" formats.) Content stored in other kinds of SQL data and in XML is less easy to migrate, as its structure and transfer methods aren't as predictable. Harder still is HTML-formatted data, followed by contents in such non-standard formats as Microsoft Word documents. Rickard recommends that all data be converted into MySQL/ PostgreSQL tables using standard tools (described below) before attempting migration into Drupal.

Whatever your content's source, document how you get data out of the source site's repository.

> "You're probably not going to have a successful export right off the bat. It's very important that you go through some trial and error while trying to get a good working data export file. Review the file every 100 records or so to make sure the data looks correct."

> *Paul Chason, Managing Partner at Drupal consultancy Mediacurrent*[33]

30   Webinar, "Best Practices for Migrating a Legacy-Based CMS to Drupal", http://acquia.com/community/resources/acquia-tv/best-practices-migrating-legacy-based-cms-drupal (at about 9:00)
31   Overview at http://www.w3.org/People/Raggett/tidy/; source at http://tidy.sourceforge.net
32   "Playing Nicely With Others: Integrating Drupal with Third-Party Data", http://acquia.com/community/resources/acquia-tv/playing-nicely-others-integrating-drupal-third-party-data
33   Webinar, "Best Practices for Migrating a Legacy-Based CMS to Drupal", http://acquia.com/community/resources/acquia-tv/best-practices-migrating-legacy-based-cms-drupal (at about 13:45)

"Even in one-time migrations, make sure that your process is: repeatable, so you can restart it at any time; testable; and done in stages, so content providers can continue to create while you get stuff together."

— *Joshua Brauer, Acquia*

Images and other non-text content

What it is: Non-text content includes such items as image, video, and audio files. They're typi-cally stored as discrete files on your server and referenced by location, although some CMSes store them as BLOBs within the database itself.

In Drupal: Drupal doesn't have a standardized way to store and reference such non-text content, although two methods are most common. The first involves simply uploading the files to Drupal's file repository and then referencing them via HTML. The second uses modules added to a core Drupal installation to provide file handling capabilities that are more native-seeming. (For images, the most common ones are FileField, ImageField, ImageAPI, and ImageCache[34]. These are all included with Acquia Drupal.) In either case, the files themselves almost always go into a subdirectory of ~/docroot/sites/sitename/files/.

Issues: Such non-text content might be spread throughout many directories in the source site, particularly if that site has had a long life or had many contributors. Some files may also live in the database and need to be extracted, while others might be on external servers (such as YouTube). Finding and organizing the content may therefore present a considerable challenge.

Once collected, you'll need to re-reference these files either by inserting HTML into the target site's content or as is required by the modules you've decided to use to manage multimedia.

"One obstacle we had to overcome [for migration of In-Fisherman.com]: Image files were stored on a server elsewhere, with no URL pointer stored in the FileMaker Pro database. To solve this, we inserted Drupal image file paths during the export, then transferred the image files themselves to an application server."

— *Paul Chason, Managing Partner at Drupal consultancy Mediacurrent*[35]

User information

What it is: User information provides a way to authenticate people for various types of site access and tag those people with personally identifying characteristics such as a name and email address. User records may also help a site track activity more accurately than cookies alone permit.

---

34   All are available as free downloads at http://drupal.org/project/*name*, where *name* is filefield, imagefield, imageapi, and imagecache, respectively.
35   Webinar, "Best Practices for Migrating a Legacy-Based CMS to Drupal", http://acquia.com/com-munity/resources/acquia-tv/best-practices-migrating-legacy-based-cms-drupal (at about 24:00)

In Drupal: By default, Drupal distinguishes Anonymous users (who have provided no user-specific information) from Authenticated users (who have provided at least a user name and email address). Administrators can create any number of additional user "roles" and grant role-based access permissions: For example, a newspaper site may have Writer, Editor, and Salesperson user roles, each with permissions appropriate for its members' jobs.

Drupal stores basic user information in the users table, with a user ID (uid) primary key. The first two users are special cases: User 0 is a catch-all for Anonymous users, while User 1 is the all-access "superuser" created when you install Drupal.

Enabling the Profile module lets you add fields to user profiles, for example to store such details as their office location and job title. Drupal stores such information in the profile_fields and profile_values tables.

Drupal links user-created content to the user who created it, typically presenting a hyperlink so visitors contact or can learn more about its author.

Issues: Because most CMSes store password information in encrypted format, you might not be able to recover it manually if password migration fails.

Many enterprises store user information in an external system, so migration plans need to include steps to integrate Drupal with that system. Fortunately, that path has been well-trod already and resources are available. (Two free modules of particular interest are LDAP Integration and LDAP Provisioning[36].) Drupal developer Matt Butcher detailed in an Acquia webinar[37] how he used SOAP to combine two external user-authentication systems with Drupal's own to merge organization members, magazine subscribers, and internal users within Drupal.

User authentication and tracking relies on cookies that are likely to differ between the source and target sites, so users will probably need to log in again after the migration.

Metadata

What it is: Metadata are information that describes other information, such as tags, content creation and modification dates, "last logged in" user information, and various kinds of workflow flags. The line between "content" data and metadata is not always clear, but a general distinction is that content can stand on its own while metadata is always subordinate to the information it describes.

In Drupal: Metadata pervade nearly every table in Drupal's database. For example, the users table contains about a dozen fields storing such metadata items as the user's time zone, language, and signature. The node table has flags to show whether the node is published and if comments are permitted.

---

36   http://drupal.org/project/ldap_integration and http://drupal.org/project/ldap_provisioning, respectively.
37   At approximately 26:00 in "Playing Nicely With Others: Integrating Drupal with Third-Party Data", http://acquia.com/community/resources/acquia-tv/playing-nicely-others-integrating-drupal-third-party-data

Drupal creates some metadata, such as content creation date. Users can change other metadata, such as whether a node is published. Drupal lets you determine which user roles have permission to change such metadata.

Issues: CMSes handle metadata very differently. Flags that exist in one may be completely absent in another, or split into several fields. Determining which metadata to migrate, and how to map it, can be a complex process.

For fields that Drupal automatically updates (such as modification dates), you need to ensure that the data you want overrides Drupal's auto-created data in fields where it's important to keep the source site's information.

Design and User Interface elements

What it is: A site's design determines how both administrators and visitors find and experience information. It includes both visual elements such as graphics and layout, and interactive elements such as menus.

In Drupal: Drupal implements a site's design through its "theme", which is mostly a combination of PHP files for page logic and CSS files for display. In general, a unified theme pervades an entire Drupal site, although you can implement different designs for specific pages, content types (for example, blog posts), or elements on a page.

Drupal theming is a study unto itself. When choosing a design firm, be sure it has experience working with Drupal, either directly or through another firm with implementation expertise.

Issues: Because you're moving between CMSes that handle and display content differently, it's very likely that you'll find some interface elements that can't be reproduced exactly in Drupal. Accept that some modifications will be necessary to avoid disappointment.

When you migrate, it's a good time to consider a site redesign at the same time. It may be easier to make a new design than to slavishly re-create the previous one; if you decide to do so, plan for a full design review process. This should preferably happen before the data migration, but it could also be done in parallel.

If you decide to redesign the site as part of the migration process, be prepared to change taxonomies to take advantage of the new design.

> "Many times a relaunch means a branding makeover from top to bottom. That doesn't only take into account layout and design, but also content. So content categories are often reshuffled, or new categories are created."
>
> — *Paul Chason, Managing Partner at Drupal consultancy Mediacurrent*[38]

Pay special attention to points of user input, such as forms. A checkbox must remain a checkbox in the target site; functionality changes if it becomes a radio button.

---

38   Webinar, "Best Practices for Migrating a Legacy-Based CMS to Drupal", http://acquia.com/community/resources/acquia-tv/best-practices-migrating-legacy-based-cms-drupal (at about 10:00)

Business logic

What it is: Business logic is a general term for systems that transform data into useful forms. For example, your current site might request visitor information through a form, then add it to a database and send it by email to several recipients. Business logic in your current site might further aggregate that information to provide sales leads and customer-service tracking.

Other forms of business logic are implied in your site's data structure. Consider a book review in a magazine. Storing the book's name and author as separate objects enables new ways to use that information, but also requires that you reference it in well-defined ways[39].

In Drupal: Drupal can implement some forms of business logic natively, either in core Drupal or through add-on modules. The example given above would be fairly easy to implement the addition of the Webform and Views modules[40], both of which are included in Acquia Drupal.

If your business logic comes from an outside system (such as middleware connected to an Oracle database), you'll very likely be able to integrate it into your Drupal site, although custom programming might be necessary to link the two.

Issues: Logic that's an integral part of the source site's CMS might have to be rewritten from scratch, at substantial cost. If so, you could either create it as a standalone program that Drupal interacts with through a protocol such as SOAP, or write it as a Drupal-native module.

Logic derived from a site's data structure can be complicated. Drawing diagrams of the source site's data objects, and how they relate to each other, may help you map how to structure the target site's data.

There's a lot of room for error when recreating business logic. Run several tests with a tried-and-true subset of your data and check the results before committing to a full migration.

---

39   Palantir's Senior Architect and Consultant Larry Garfield gives this example from migration of the magazine Foreign Affairs, starting at about 17:00 in "Playing Nicely With Others: Integrating Drupal with Third-Party Data", http://acquia.com/community/resources/acquia-tv/playing-nicely-others-integrating-drupal-third-party-data
40   http://drupal.org/project/webform and http://drupal.org/project/views, respectively.

Migration Tools

Now that you've evaluated your assets, it's time to decide what to do with each of them. You essentially have three options:

Remove the asset from the site. Migration offers an opportunity to streamline a site by removing unneeded junk.

Leave the asset in an external location, whether that's the original CMS, a separate database or other business system, or a remote service. This is a fairly common arrangement, and in fact Drupal is sometimes used as a framework to support other CMSes[41]. In this case the most important matters are that connections between Drupal and the external system are fast and reliable, and that Drupal handles the asset as desired.

Palantir's Ken Rickard calls this scenario a "continuous migration" or "continuous integration", as data must regularly travel between Drupal and the external system. (One example he gives is of a company that wants to continue to use its legacy tools for content creation, but automatically migrate pieces to Drupal for publishing.)

Rickard recommends using XML feeds where possible, although in some cases Drupal can access content in a foreign source database directly. Drupal 7, due out in 2010, abstracts the database layer to (theoretically) allow it to use data in any database program with the appropriate driver. (The planned "Drupal 7 driver for SQL Server" module[42] will take advantage of that feature.)

Move the asset to the target site. Some assets only need to be moved one time; from then on, they live only on the target site. The remainder of this section focuses mostly on these "fire and forget" migrations.

Certain tools can perform the majority of the work in many cases. In particular the Migrate module is essential for one-time migrations. (It's described in the Drupal Tools section, below) But you'll probably use a variety of software to manage data before and during migration. Here are some of the more common ones.

Migrating to a newer version of Drupal

This paper hasn't discussed one important kind of migration, from an earlier version of Drupal. Within major versions of Drupal (i.e., those with the same integer number), the procedure is fairly simple: You basically just run a script (update. php) that's part of Drupal to bring the database in line with the new version. But there are some caveats for migrations between major versions.

- Before migrating, make sure that all the modules your site uses are available in the new version. If they are, add them to your Drupal installation before performing any data migration. If not, disable them on the source site and redesign as necessary first.

- The Drupal community maintains the current integer version and the one previous, e.g. Drupal 6 and Drupal 5. It also works to ensure that migrations between the latest two versions work as expected when you run the script.

- If you need to migrate between more than one Drupal version — from Drupal 4 to Drupal 6, for example — you need to first update from Drupal 4 to Drupal 5, then from Drupal 5 to Drupal 6.

---

41   One example is Wikimedia's donation system, described at http://tomgeller.com/content/can-
        drupal-handle-high-traffic-sites#comment-134 et seq. In that case Drupal handles record-keep-
        ing while the remainder of the site runs on the MediaWiki CMS.
42   http://drupal.org/project/sqlsrv

General Tools

Text-management tools are excellent for making mass changes on sources in plain-text format, including both database exports and HTML text files. Scriptable text tools are particularly useful for such tasks as conditional changes and those that are part of a larger procedure. Literally dozens of text tools meet this criterion; Emacs and Vim are notable for being free, scriptable, and available for a wide variety of operating systems[43].

Database tools specific to the source site's database format may allow you to make certain changes that are either difficult or impossible through text-management tools. phpMyAdmin is the most popular tool for MySQL, the database program common to many CMSes[44].

QueryPath[45] is a PHP library by Drupal developer Matt Butcher[46] that lets you perform a long list of complex operations on HTML and XML texts. It's available at http://querypath.org.

Pentaho BI[47] is an open-source, Java-based suite of programs to manage business intelligence (BI). Sandy Smith of Web strategy company Forum One[48] reports great success using Pentaho to transform data as it came from the source site.

"Pentaho's biggest thing is that it has a lot of good connectors. It has a way to, without a lot of programming, do the needed transformations. So if you need to take a couple of disparate data sets to combine, it allows you to do that through configuration. It's somewhat enterprise-level, but the basic tools are open-source."

— Sandy Smith, Manager of Technical Development, Forum One

"One use case that's become really common [using the Migrate module] is moving from Drupal to Drupal. People are on Drupal 4 or Drupal 5 and need to get to Drupal 7. But they can't get the whole upgrade path for all the modules they were using up to where they want to be, so they actually treat their old Drupal site as a foreign database."

—*Moshe Weitzman, Co-Founder of Cyrve, a consultancy specializing in Drupal data migration*

One extremely ambitious migration currently underway involves moving one of the world's largest websites — Examiner.com — from ColdFusion to the unreleased Drupal 7. While this isn't a project for the faint of heart, Examiner.com will continue to run a supported version of Drupal for an extra year or two because of their daring. The details are in an Acquia webinar, "A Peek Behind the Scenes – How Clarity Media is Moving Examiner.com to Drupal 7[56]," and Case Study, "Revolutionizing the Online Media Market Through Open Source Publishing[57]."

56   http://acquia.com/resources/
     webinars/peek-behind-scenes-
     %E2%80%93-how-clarity-media-
     moving-examinercom-drupal-7
57   http://acquia.com/resources/
     library/case-study-examinercom

---

43   An extensive comparison of text editors, with links to further information and software sources, is
     at http://en.wikipedia.org/wiki/Comparison_of_text_editors.
44   A list of other database tools is at http://en.wikipedia.org/wiki/Category:Database_administra-
     tion_tools.
45   http://querypath.org
46   http://technosophos.com
47   http://pentaho.com
48   http://forumone.com

Drupal Tools

The Drupal tools listed here are all free modules that are available for free from the Drupal.org website. Two pages on that site attempt to provide up-to-date overviews of migration modules: "Contributed modules for migration, deployment, backup, and import[49]" and "Comparison of Content and User Import and Export Modules[50]."

Essential

Drush (http://drupal.org/project/drush) is an extensible command-line interface to Drupal with over a hundred commands available, and many more available through the dozens of modules that extend it[51]. It's a must-have tool for Drupal developers generally, and is required for migrations to use the Migrate module (below).

Import/Export utilities

Migrate (http://drupal.org/project/migrate) is an extensible framework for loading, transforming and saving data into Drupal. It automates creation of content nodes -- a feature common to all import utilities listed here -- but goes further by also handling users, user roles, fields, comments, and URL aliases.

Migrate version 2 (available for Drupal 6 and 7) has been extensively rearchitected and can import data directly from Oracle or Microsoft SQL Server, or any other database with a PHP driver.

> "[Migrate] is quite pluggable, so if people have a really strange data source, they can just write a little class that has a few methods in it. It's pluggable enough that someone could write a Migrate fetch driver that [retrieves content] from a directory of HTML files and then saves them to Drupal."
>
> *— Moshe Weitzman, Co-Founder of Drupal consultancy Cyrve[52] and co-maintainer of the Migrate module*

---

49   http://drupal.org/node/417192
50   http://groups.drupal.org/node/21338
51   A complete list of Drush commands is available by typing "drush help" when the module has
     been installed. A somewhat dated command list is also at http://drupal.org/node/477684. A list
     of modules that extend Drush is at http://drupal.org/taxonomy/term/4654.
52   http://cyrve.com

Feeds (http://drupal.org/project/feeds) uses a different model to migrate data into nodes, users, or taxonomy terms. It combines two previous modules (FeedAPI and Feed Element Mapper) to allow data input from file sources as well as RSS feeds, among other improvements. Reflecting its name and origins, Feeds is especially good at managing ongoing migrations that accept periodic updates from an external source[53].

Import HTML (http://drupal.org/project/import_html) takes static HTML sites and attempts to create nodes from them. It's an ambitious effort, considering the variety of source sites possible, and the module has several requirements and needs considerable configuration. Even so, it could save you hundreds of hours of work over a manual migration.

Node Export (http://drupal.org/project/node_export) can both export and import node content, despite its name. While not as flexible (or heavy) as the Migrate module, Node Export perform simple content migration tasks quickly.

CMS-specific tools

Joomla to Drupal (http://drupal.org/project/joomla) moves content, users, and taxonomy terms ("sections and categories") directly from Joomla's database. It can be used for either one-time or continuous migrations.

Wordpress Import (http://drupal.org/project/wordpress_import) is a versatile tool for moving content, users, tags, and other information from this extremely popular blogging program. Its source is a representation of the site in WordPress eXtended RSS (WXR) format, so only information found in that format is available. (User passwords are notably absent, so new passwords will be necessary in Drupal.)

Ten Tips for a Successful Site Migration

1. **Catalog site assets.** Write down everything that comprises your site, from content to user authentication to system integration links. See the "Taking Stock of Your Current Site" section in this paper for tips on creating this inventory.

2. **Catalog site functionality.** Now that you've documented what your site is, it's time to note what it does. Walk through the site

---

53   Alex Barth, Lead Developer at Drupal consultancy Development Seed, presented a session at DrupalCon 2010 (San Francisco) about using feeds to aggregate and import data into Drupal; video of that session is at http://sf2010.drupal.org/conference/sessions/aggregate-and-import-feeds.

and ask: What tasks can I do on this site? What do I expect of it? Chat, e-commerce, games, mapping, and social networking are all examples of functions that will require additional modules (or custom programming) in Drupal.

3. **Decide what to move.** You're often better off not migrating some assets. Some may remain in their original repository; others will become part of another project; others will simply be jettisoned. Ensure that all assets will end up in their optimal place, which might not be on the target site.

4. **Clean up and normalize the old site.** In the process of examining site assets, you may discover some that should be moved, but have become messy over time. Determine whether it will be easiest to clean them on the source site, on the target site, or during the migration process.

"Sometimes the data source is parsable for the most part, but maybe in the body of the article they have tags and images that are baked in; in the Drupal site those want to be separated. You basically have to make a decision at that point about whether you want to put a lot of effort into regular-expression parsing and try to get them out (to the extent that's even possible). But sometimes it behooves a client to take a fresh look at all the articles on a site, and clean them up."

— *Moshe Weitzman, Co-Founder of Drupal consultancy Cyrve*

5. **Test with a subset of the site's actual data, if privacy policies permit.** Phony data (such as is created by a "Lorem Ipsum" generator) won't reveal data-related flaws that would appear during the actual migration, while tests using the full data set might take too long to complete.

6. **Implement functionality on the new Drupal site. Ensure** that you're moving not only data, but also function. All modules, custom programming, and data schema should be in place before you move content.

7. **Prepare stakeholders for the change.** Even with the best designers and programmers, there will be visible differences between the source and target sites. Users might have to create new passwords or learn new procedures to use the site. Include documentation and training in your plans as needed.

8. **Move in phases.** A migration is a multi-part process, and some parts should be done before others. It's usually best to create

users on the target site before creating content, for example, as all content in Drupal references an author ID by default. Also, some data sets might be too large to effectively move all at once because of system limitations. Have a way to move segments of data and keep track of which segments you've moved. Joshua Brauer, a Drupalist at Acquia, recommends migrating assets in the following order:

1. Users, including user roles and permissions (if applicable). You might need to do user profiles later, if you decide to store them as nodes by using the Content Profile module[54].

2. Taxonomy terms and vocabularies, as you'll need to assign them to content as it comes in.

3. Other metadata.

4. Content.

"In an ideal situation, you'll limit changes [during migration] as much as is practical. But sometimes the tendency is to say, 'We're going to try to go a week or two without creating any new users', instead of saying, 'Let's build a process that makes it easy and repeatable to migrate users, and run a user migration on a regular basis.'"

— *Josh Brauer, Drupalist, Acquia*

9. **Plan for problems.** Migrations almost never go perfectly on the first try! It's an iterative process: You move some data, then notice a problem; fix the problem, try again, notice a different problem; and so forth. Make sure you have ways to roll back changes at any step to make corrections as needed.

10. **Establish a relationship with a Drupal consultant who'll be on call.** Unless you're already experienced with Drupal, you'll run into issues that seem impassible to you, but that a good Drupal consultant will be able to fix quickly. Acquia is partnered with more than 150 Drupal consultancies across the globe to help you find one that matches your business needs.

"If you're relaunching the site, [people] will probably be creating content until the week before launch. They may introduce content into the site in a format that you didn't consider when building the final import script. So it's always a good idea to have a Drupal consultant on call while making that final import."

— *Paul Chason, Managing Partner at Drupal consultancy Mediacurrent[55]*

---

54   http://drupal.org/project/content_profile
55   Webinar, "Best Practices for Migrating a Legacy-Based CMS to Drupal", http://acquia.com/community/resources/acquia-tv/best-practices-migrating-legacy-based-cms-drupal (at about 29:30)

About Acquia

Acquia empowers enterprises with the open-source content-management system Drupal. Co-founded by Drupal's creator in 2007, Acquia helps customers manage their growth and scale their online properties with confidence. Acquia's software, consultation, cloud infrastructure, and services enable companies to realize the full power of Drupal while minimizing risk, as it's done for Examiner.com, Al Jazeera, and over 700 others.
See who's using Drupal at
http://showcase.acquia.com
and learn more at
http://acquia.com.