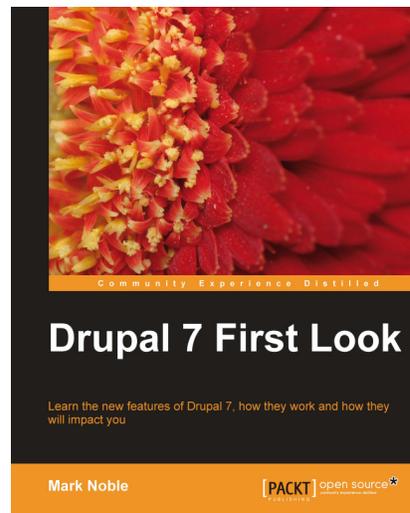# Drupal 7 First Look

Mark Noble

Chapter No. 3
"Site Building with Drupal 7"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.3 "Site Building with Drupal 7"

A synopsis of the book's content

Information on where to buy this book

About the Author

Mark Noble has worked in software development and website design for nearly 15 years in a variety of capacities including development, quality assurance, and management. He has worked in a variety of industries including architecture, engineering, library automation, telecommunication, and more. He takes pride in developing software and websites to make businesses run more effectively and delights in helping users to get their jobs done more easily.

Mark owns and operates Turning Leaf Tech, LLC (`http://www.turningleaf tech.com`) —a company dedicated to building high-quality web-based solutions for businesses. Projects include both Drupal-based projects and non-Drupal-based sites and solutions. Work ranges from complete site design and implementation to upgrading sites, and building new features to existing sites.

When he isn't behind a computer, Mark enjoys playing with his family, photography, hiking, and traveling.

> I would like to thank my family for their support during the writing of this book. You are all amazing, and I love you very much. I would also like to thank the Drupal community for putting out such a fantastic open source application. Without you, books like this would not be possible.

Drupal 7 First Look

Welcome to Drupal 7, the next generation of the popular Open Source content management system.

If you are new to Drupal, you will find Drupal 7 to be easy to use, yet extremely powerful. Over the course of this book we will work our way through the most important features of Drupal 7, so you can begin to gain familiarity with Drupal.

For those of you who used Drupal 6 or an earlier version of Drupal, you will find many new enhancements that are designed to make your day-to-day administration tasks easier as well as giving you additional power to extend Drupal to meet the needs of even the most advanced sites. We will spend much of the book discussing the new features of Drupal 7 as well as how Drupal 7 differs from Drupal 6, so you can upgrade your existing sites to the new platform and take advantage of all that Drupal 7 has to offer.

What This Book Covers

Chapter 1, What's New In Drupal 7?, introduces the reader to the important changes and new features of Drupal 7. It also gives a brief account of the functionality that has been removed entirely, moved into a contributed module, or removed in favor of existing contributed functionality.

Chapter 2, Installation and Upgrades, deals with installing Drupal 7 and upgrading your Drupal 6 site to Drupal 7. It also covers setting up custom installation profiles, so you can install multiple sites using the same basic configuration.

Chapter 3, Site Building with Drupal 7, shows you how to create custom content types with custom fields, add content to your site, translate your content, and leverage all of the new Drupal 7 functionality. It also looks at changes to the Drupal 7 content management system.

Chapter 4, Drupal 7 Administration, looks into changes that have been made to the Administration interface in Drupal 7. It covers information about the new functionality that has been added to Drupal 7 and areas where configuration options have been moved or renamed from Drupal 6 to Drupal 7.

Chapter 5, Drupal 7 for Themers, walks through all of the major changes to the Drupal theme system that were introduced in Drupal 7. It looks at changes to the template files, JavaScript, CSS, and API. It also shows how to upgrade an existing Drupal 6 theme to Drupal 7.

Chapter 6, Drupal 7 Database Changes, deals with the changes to the database abstraction layer of the Drupal API. It introduces the reader to the DBTNG layer.

Chapter 7, Drupal 7 for Developers, looks at the API in terms of upgrading your existing modules and site code to Drupal 7. It also discusses changes related to the Drupal API including changes to the core system, menu system, Form API, and file uploads.

Appendix, Post Publication Changes, because this book was written before the final release of Drupal 7 (much of it was initially written based on early Alpha and Beta versions of the software) some screenshots and descriptions have evolved as Drupal has progressed to its final release. This Appendix will describe any areas that changed after publication. The Appendix is not part of this book but available online at www.packtpub.com for free download.

3
Site Building with Drupal 7

One of Drupal's strengths has always been the ease with which you can add, edit, and organize the information on your site. There are several key changes to Drupal 7 that will make your life as a content editor easier, including:

- Improved interface for adding content makes your work more productive.
- The ability to add and modify fields in Core rather than using the contributed CCK module makes it easier to customize and validate the information on your site. This functionality was built from and replaces the contributed CCK module.
- Improved categorization with taxonomies.
- Built-in support for image handling.
- Improved support for handling files and attaching files to content.
- More control over text formats.

In this chapter, we will cover all of these new features and enhancements. By the end of the chapter, you should be able to create custom content types with custom fields, add content to your site, translate your content, and leverage all of the new Drupal 7 functionality.

Creating content

Creating content for your site is at the core of any Content Management System like Drupal. The primary changes for Drupal 7 relate to an updated interface. Let's look at the new interface in detail.

Selecting a content type to create

To create content in Drupal 7, first log in to your site and then click on **Content** from the site toolbar. Drupal will now display the new Content Administration page.
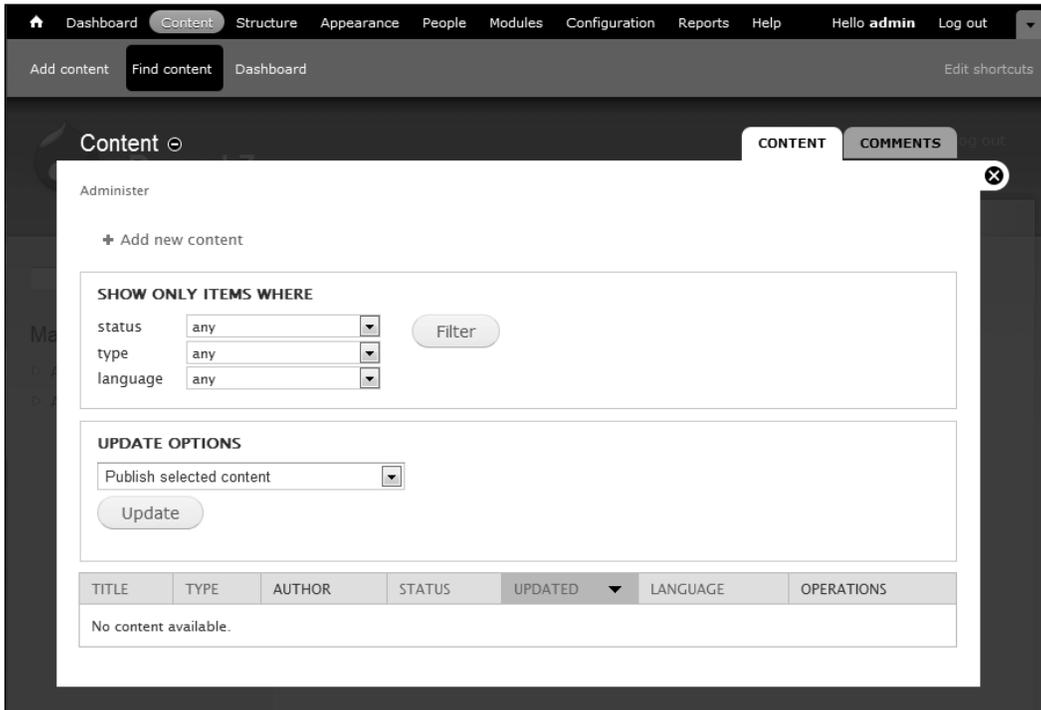
In Drupal 6, this page could be displayed by selecting **Administer | Content Management | Content** from the Navigation menu.

From here, click on the **Add new content** link. Drupal will now display a page allowing you to select the type of content you want to create. Depending on the modules you have installed and enabled, you will have different content types available.
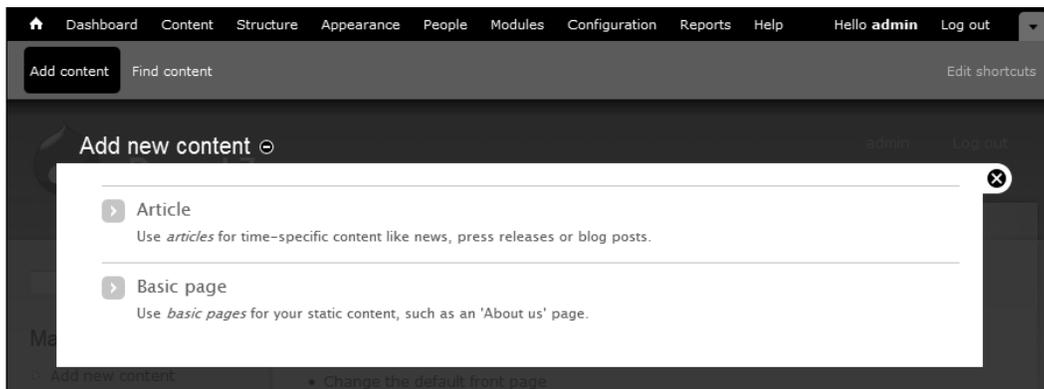
In previous versions of Drupal, this page could be reached by selecting the **Create Content** link from the Navigation menu.

You can also select the type of content to add using the shortcut bar. You can access the shortcut bar by clicking on the toggle at the far right of the toolbar:

The shortcut bar has a list of links in it that can be used to quickly access commonly-used functionality, and it appears as follows:


You can customize the links in the shortcut bar and users can use either the default set of shortcuts or they can have their own. We will look into building shortcuts in the next chapter.

Now select the type of content you want to create. For this example, we will use the **Basic page** type.

Content UI

The interface to create content has been altered drastically from Drupal 6. Let's go through the interface in detail.
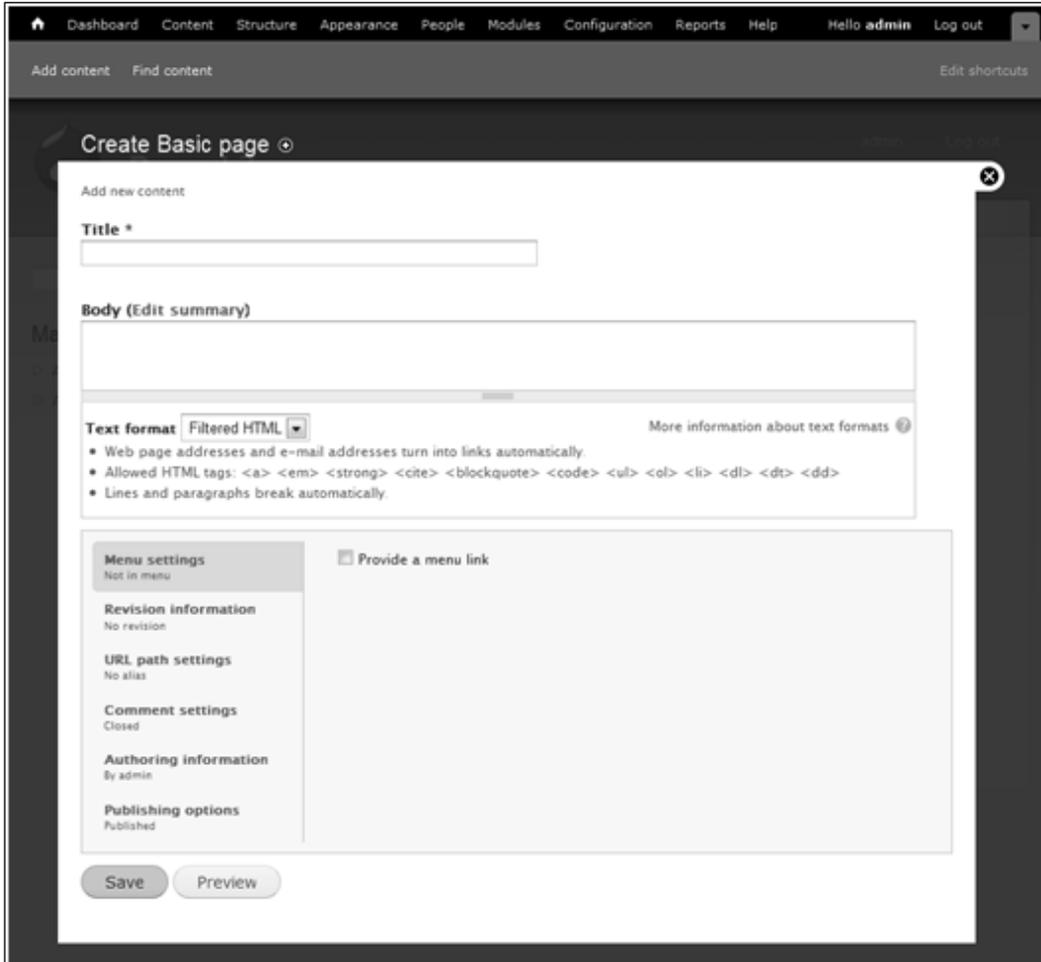
The top section of the page should be familiar to experienced editors. This is the place to enter your title as well as the full text of the page. In a departure from previous versions, the node summary, which is used when multiple nodes are displayed on a page, is an entirely separate optional field.

Creating node summaries

To create a node summary, click on the **Edit summary** link next to the **Full text** label. Drupal will display a second field that can be used to define your summary.

Summary (Hide summary)

Leave blank to use trimmed value of full text as the summary.

Full text

The **Summary** entry in Drupal 7 is much easier for new content editors to work with. There is now a separate text area where the Summary can be entered. Of course, you can still use a subset of the full text by simply leaving the **Summary** blank.

Formatting text

Below the **Full text** field, you will find a list of available text formats that can be applied to the text field.

Full text (Edit summary)

Text format | Filtered HTML ▾   More information about text formats ❓
Web page addresses and e-mail addresses turn into links automatically.
Allowed HTML tags: <a> <em> <strong> <cite> <blockquote> <code>   <dl> <dt> <dd>
Lines and paragraphs break automatically.

This display is much more compact than that in previous versions making it much easier to select a text format to apply. Text formats are used to limit which HTML tags can be inserted into the full text field. By default, Drupal 7 includes **Plain Text**, **Filtered HTML**, and **Full HTML** formats. If you enable the PHP filter module, you will also have access to the **PHP code** text format. These are all configured identically to the Drupal 6 Input Filters of the same name.

Text formats can be edited by selecting **Configuration** from the administration toolbar and then selecting **Text formats** from the **Content Authoring** section. The interface for creating text formats is very similar to the Drupal 6 interface for creating and editing input formats. You can still assign text formats to roles and control which filters run on the input. However, some of the filters have been renamed and additional information has been added to the display to help you select which filters you want to apply. The filters map as follows:

- **Convert URLs into links** was called URL filter in Drupal 6
- The **Convert line breaks** filter was called Line break converter in Drupal 6
- **Correct broken HTML** was called HTML corrector in Drupal 6
- **Escape all HTML** was part of the Drupal 6 HTML filter that was broken up to improve usability
- **Limit allowed HTML tags** was also part of the Drupal 6 HTML filter
- The **PHP evaluator** remains as is from Drupal 6

Additional node properties

Below the text format, you will find a set of vertical tabs containing all of the optional properties you can apply to your new node:


This display replaces the collapsible field sets used in Drupal 6. The new display is much better looking and easier to use. The contents of each tab are similar to prior versions.
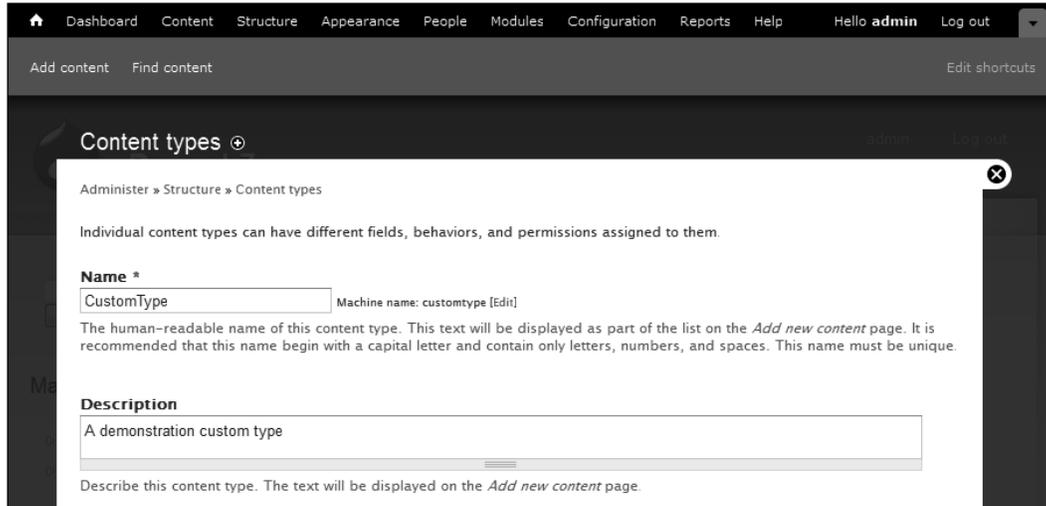
Creating new content types with custom fields

Drupal 6 provided the ability to add new content types, which represented different types of content on your site. Visitors could then search the different types of content and you could change how they were displayed. However, Drupal 6 by itself was pretty limited as to the types of information you could attach to new content types. To get around these limitations, a majority of sites installed the contributed CCK module (short for Content Construction Kit). This module allowed administrators to add fields to a content type. Each field could be configured with the type of information it could contain, how it was validated, and what type of input control was used in edit forms. Because creating new content types was done so often and because a majority of sites used CCK, the Drupal core team decided to add much of the functionality from CCK into Drupal core.

Creating a custom content type

Content types are created within the **Structure** portion of the Administration interface. Start by clicking on the **Structure** link in the toolbar and then click on **Content types** from the **Structure** page. Depending on what modules you have enabled, the resulting page will appear similar to the following screenshot:
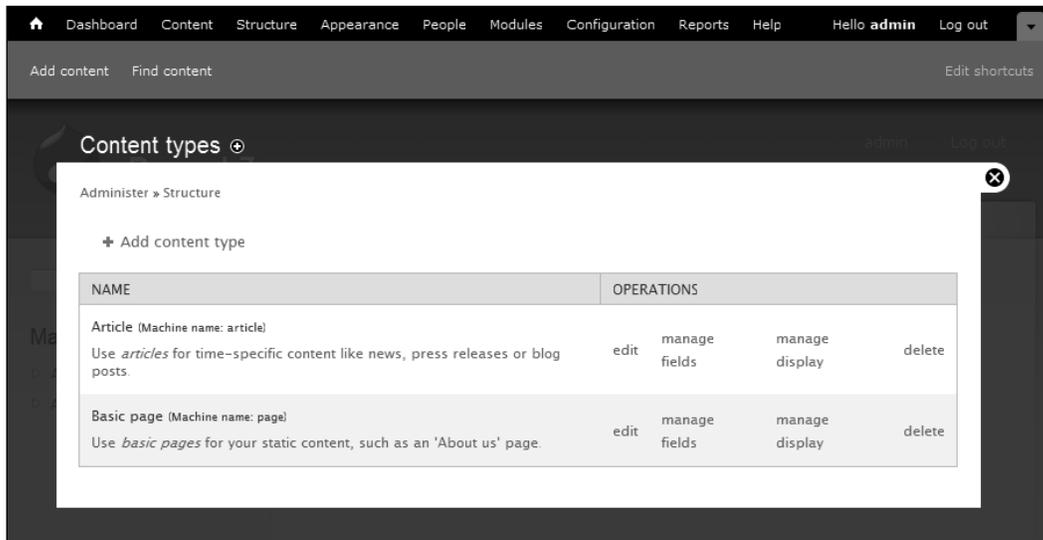
To add a new content type, click on the **Add content type** link just under the header. Drupal will now display a form that allows you to enter information about the content type you want to create. Let's look at each section of the form.

The top portion of the section allows you to name your content type and set a **Description** for the type. A nice new feature of Drupal 7 is the automatic generation of the **Machine name**. Of course, you can also enter your own machine name by clicking on the **Edit** link next to the default machine name.

Additional content type properties

Below the header information is another set of vertical tabs, which allow you to configure additional properties for the content type.

The **Submission form settings** and **Publishing options** are very similar to Drupal 6 and are mostly self explanatory, so we'll leave them for now.

Display settings

The **Display settings** section is new to Drupal 7 and appears as follows:
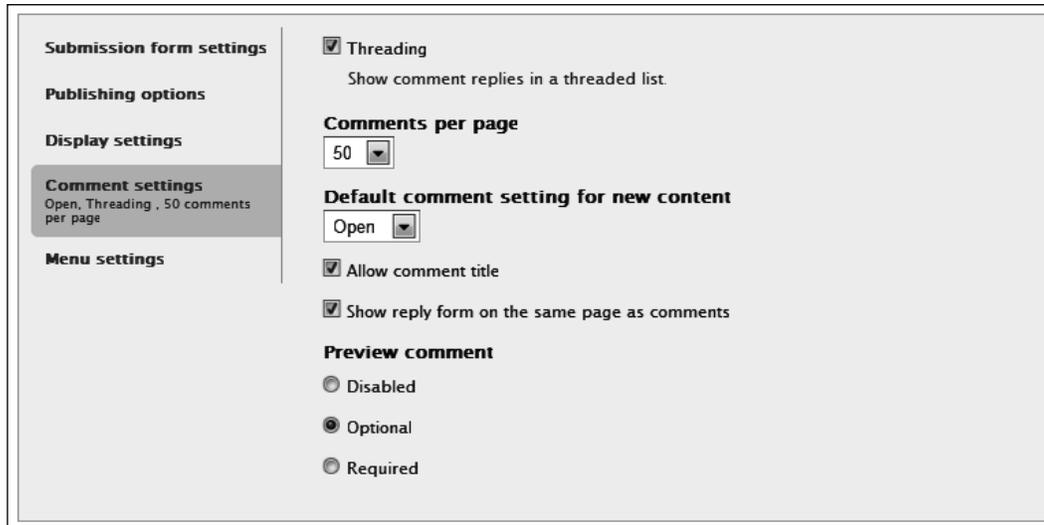

The display settings are used to control whether or not the author of each post is identified in the display as well as the total length of the teaser or trimmed post when the node is displayed in a listing with other nodes.

Comment settings

The **Comment settings** in Drupal 7 have been greatly refined from Drupal 6. Drupal 6 offered nine different sections of configuration and over 20 different choices taking up two screens worth of real estate. In Drupal 7, the list has been reduced to four basic settings and only eight different controls.

The remaining controls serve the same function that they did in Drupal 6. The controls which were removed were not widely used and reasonable defaults have been inserted instead.

Menu settings

The **Menu settings** allow you to determine which menus are allowed to contain links to content of this type. This makes menu management much easier since you do not have to scroll through potentially hundreds of items when setting menu links for a particular piece of content.

You can select as many menus as you would like and set a default menu for the content type. This can save you and your content editors a great deal of time while building content.

After you are happy with your content type, simply click on **Save content type** to save the new content type in the system. If you plan to add fields immediately, you can also click on **Save and add fields**.

Field API

Now that we have looked at the modifications to creating content types in Drupal 7, let's look at the new Field API that allows us to define custom fields for our new content type. There are a variety of ways to access the fields for a content type. From the list of content types, you can click on **manage fields**, or if you are already editing a content type, you can click on the **MANAGE FIELDS** tab.

After you select the **MANAGE FIELDS** tab, you will be taken to a list of the existing fields for your content type, with a quick way to add either a new or existing field.

> The **Title** and **Body** fields are now fields. You are required to keep the **Title** field, but you may modify or even remove the **Body** field. This gives a great deal of additional flexibility to your content types.

To create a new field, simply enter the label you want to use for the field as well as the name of the field. The name is used to identify the field internally and must consist of only lowercase letters, numbers, and underscores. After naming the field, you will need to select the field type. The available field types in Drupal 7 are:

- Boolean
- Decimal
- File
- Float
- Image
- Integer
- List
- List (numeric)

- List (text)
- Long text
- Long text and summary
- Term reference
- Text

Many of these field types will be familiar to past users of CCK. However, there are also some important new field types that were either not present in prior versions or required extensions to CCK. These include **File**, **Image**, **Long text and summary**, and **Term reference**.

After you select the type of field, you will need to select the widget to use when displaying the field. In Drupal 7, these include:

Widget name	Applicable field types
Select List	Boolean, List, List (numeric), List (text), Term reference
Checkboxes/ radio buttons	Boolean, List, List (numeric), List (text), Term reference
Single on/off checkbox	Boolean
Text field	Decimal, Float, Integer, Text
File	File
Image	Image
Text area (multiple rows)	Long text
Text area with a summary	Long text and summary
Autocomplete term widget (tagging)	Term reference

Let's look in more detail at each field type.

Boolean fields

Boolean fields allow you to add fields that only have two possible values. Typically, these are yes/no, true/false, on/off, and so on. Let's add a sample field. For the field label, enter **Is Drupal the best CMS**, for the field name enter **is_drupal_best**, select **Boolean** for the field type, and **Check boxes/radio buttons** for the Widget. Click on **Save** to add the new field.

After the field is saved, Drupal will present a form where you can set the options for the field. The contents of this form will vary depending on the type of widget you select, but for our configuration it will appear as follows:
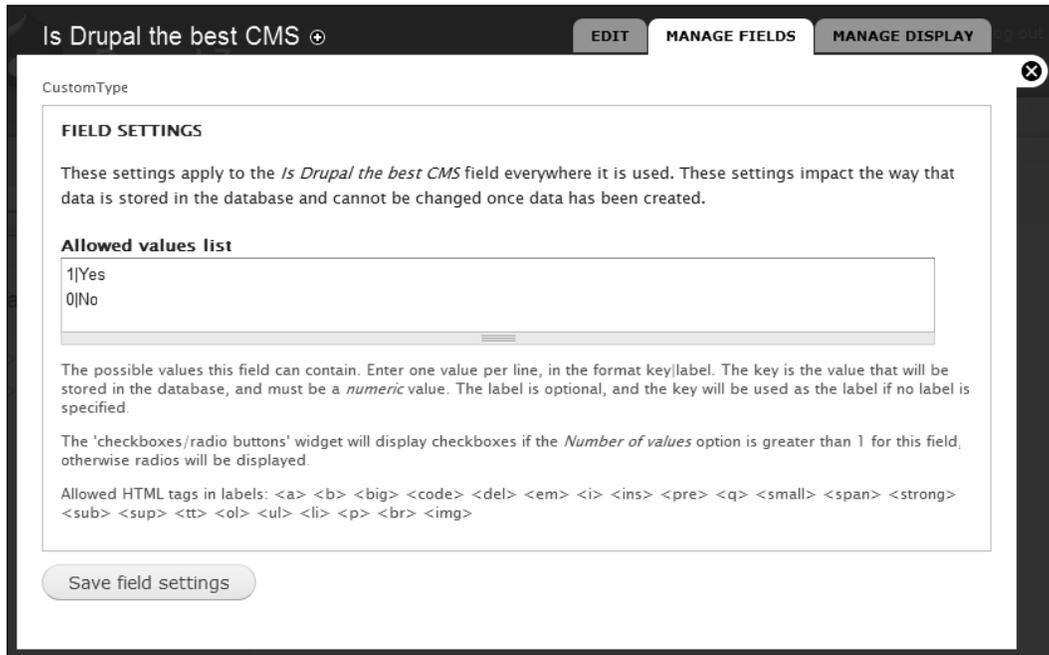
Let's look more at **Allowed values list**. The **Allowed values list** field allows you to specify the options you want to display, with the value for each option. Typically, a Boolean field will use 0 for the option representing no/false/off and either 1 or -1 for the option representing yes/true/on.

> The options will display when the field is being edited in the same order as the values in **Allowed values list**. It's a good idea to put the most common value first.

After setting the allowed values, click on **Save field settings**. Drupal will now prompt you for information related to default values, whether or not the field is required, and help text as shown below:

Updated field *Is Drupal the best CMS* field settings.

CUSTOMTYPE SETTINGS

These settings apply only to the *Is Drupal the best CMS* field when used in the *CustomType* type.

Label *

Is Drupal the best CMS

☐ Required field

Help text

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub> <sup> <tt>   <p>
 <img>

DEFAULT VALUE

The default value for this field, used when creating new content.

Is Drupal the best CMS

○ N/A

○ Yes

○ No

IS DRUPAL THE BEST CMS FIELD SETTINGS

These settings apply to the *Is Drupal the best CMS* field everywhere it is used.

Number of values

1 ▼

Maximum number of values users can enter for this field.

Allowed values list

1|Yes
0|No

The possible values this field can contain. Enter one value per line, in the format key|label. The key is the value that will be stored in the database, and must be a *numeric* value. The label is optional, and the key will be used as the label if no label is specified.

The 'checkboxes/radio buttons' widget will display checkboxes if the *Number of values* option is greater than 1 for this field; otherwise radios will be displayed.

Allowed HTML tags in labels: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub> <sup> <tt>   <p>
 <img>

Save settings

You can also override the default label and change the allowable values. Feel free to modify these settings as needed. For this example, the defaults are fine. When you are finished, click on **Save settings**.

You will now be returned to the list of fields and our new field will be displayed:

| ⊹ | Is Drupal the best CMS | field_is_drupal_best | Boolean | Check boxes/radio buttons | edit | delete |

> You can modify the widget used to display your field by clicking on the name of the active widget. After modifying the widget, review your options to make sure they are still valid for the new display.
>
> The Boolean field is new to Drupal 7 and did not exist in the base installation of CCK for Drupal 6. In Drupal 6, the Integer type was typically used when a Boolean was needed.

Numeric fields (Decimal, Float, and Integer)

The Decimal, Float, and Integer fields are used to store numeric information. The Decimal and Float fields both allow numbers with decimals to be entered, while the Integer field only allows whole numbers. The Float and Decimal fields store their data in different types within the database, and the Decimal field gives you additional control over the precision and scale of the values which can be input. In all three cases, Drupal takes care of all input validation for you, so you don't have to worry about someone trying to enter "two" as a value. Drupal will also make sure the numbers fall within the range of valid values you specify. Let's take a look at adding a numeric field.

The basic procedure for adding a numeric field is similar to that for adding a Boolean field. First we will set the label and internal name for the field and then select the field type and widget for the field. Both Decimal and Integer types only allow the Text field widget, so our decision will be easy. For this example, we'll label the field **How many years have you used Drupal** and name it **num_years**. The type will be Decimal and the widget Text field.

> Make sure to carefully consider what type of field you want to use before adding the field, since you will not be able to change the field type after it has been created.

After you save the basic field information, Drupal will again prompt you for some additional information specific to the Decimal type:

Precision

10 ∨

The total number of digits to store in the database, including those to the right of the decimal.

Scale

2 ∨

The number of digits to the right of the decimal.

Decimal marker

decimal point ∨

The character users will input to mark the decimal point in forms.

Decimal marker can be set to **decimal point**, **comma**, or **space** based to match the expectations of your visitors.

> The Integer and Float types do not require you to set any additional settings.

The **Scale** and **Precision** settings work together to determine how the number is stored. If you increase the scale (number of digits to the right of the decimal), you will need to increase the precision as well if you want to retain the same number of overall digits.

After you have set these options to your desired values, click on **Save field settings**. You will now be presented with a set of options that allow you to set minimum and maximum values, as well as the prefix and suffix:

These settings apply to both Decimal and Integer fields. The **Prefix** and **Suffix** fields are valuable tools for giving your site visitors and editors more information about the type of data that is expected. After you have customized the settings, select **Save settings** to finalize the field. You can change these options after the field has been saved by clicking on the **Edit** link in the list of fields.

All three of these field types existed within CCK for Drupal 6 and are very similar to their Drupal 6 counterparts.

File fields

File fields can be used to allow editors to attach files to content. For example, you can attach training manuals or product data sheets to your content. Drupal will handle uploading files to the server, ensuring they have the proper type and are smaller than the maximum size. The basic procedure to add a file field is the same as the other fields we have looked at so far. Specify the label and name for your field and set the type to **File**. The only available widget is **File**. The custom settings for this type are as follows:

The **Enable Display field** setting is used to determine whether or not the file can be viewed when the node it is attached to is viewed. The **Files displayed by default** setting is used to determine if a link to the file should be included by default. This can be overridden for any file that is attached to the node. **Upload destination** determines if files will be stored within the **Public files** directory, which can be accessed directly by visitors, or if the files will be stored in the **Private files** directory, which is controlled by Drupal allowing you to add additional security for how the files are accessed.

After these basic options are set and saved, you will be taken to another screen where you can edit more advanced options as shown below:

The **Allowed file extensions** setting allows you to determine which types of files can be added. When files are uploaded, Drupal will validate that they are of the correct type. This is an important setting to help ensure that visitors are not uploading malicious content to your site. The **Maximum upload size** setting is also important to make sure that your server is not swamped with exceptionally large files. The **File directory** setting allows you to segregate files within your upload destination. This can be a great way of grouping files on your site so you always know the source of the file. In Drupal 6, you could achieve similar functionality using the FileField extension for CCK.Image fields

Image fields are an exciting and much sought-after inclusion in Drupal core. As the name implies, image fields allow you to attach images to your content type. Building an image field follows similar steps to other fields. First set the label and name for your field and then set the type and widget to **image**. The basic options allow you to control where the files are stored and allow you to provide a default image that will be used if the editor does not attach an image when they create an instance of the content type. If you do not have a default image, nothing will be shown if there is no attached image:

After saving these basic options, you will be presented with more advanced options giving you control over what types of images can be inserted as well as how large images can be. Image size can be controlled based on both the total size of the file in bytes, kilobytes, or megabytes, as well as based on the maximum and minimum dimensions of the image in pixels. This gives you a lot of control over what images can be attached to the node and helps when you are ready to theme your content, since you will already know how big the images can be.

The Alt and Title attributes can be activated to allow editors to give additional information about the image. The Alt and Title tags are used for tooltips as well as by screen readers, search engines, and browsers when the image cannot be displayed. In most cases, it is a good idea to enable these options. The full list of options is shown below:

CUSTOMTYPE SETTINGS

These settings apply only to the *image* field when used in the *CustomType* type.

Label *

image

☐ Required field

Help text

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub> <sup> <tt>   <p>
 <img>

Allowed file extensions

png, gif, jpg, jpeg

Separate extensions with a space or comma and do not include the leading dot. Leaving this blank will allow users to upload a file with any extension.

File directory

Optional subdirectory within the upload destination where files will be stored. Do not include preceding or trailing slashes.

Maximum image resolution

_____ x _____ pixels

The maximum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Leave blank for no restriction. If a larger image is uploaded, it will be resized to reflect the given width and height. Resizing images on upload will cause the loss of EXIF data in the image.

Minimum image resolution

_____ x _____ pixels

The minimum allowed image size expressed as WIDTHxHEIGHT (e.g. 640x480). Leave blank for no restriction. If a smaller image is uploaded, it will be rejected.

Maximum upload size

_____

Enter a value like "512" (bytes), "80 KB" (kilobytes) or "50 MB" (megabytes) in order to restrict the allowed file size. If left empty the file sizes will be limited only by PHP's maximum post and file upload sizes (current limit **64 MB**).

☐ Enable *Alt* field

The alt attribute may be used by search engines, screen readers, and when the image cannot be loaded.

☐ Enable *Title* field

The title attribute is used as a tooltip when the mouse hovers over the image.

Preview image style

thumbnail ▾

The preview image will be shown while editing the content.

In Drupal 6, you could achieve similar functionality using the ImageField module, which required both the FileField module and CCK.List fields

List fields allow editors to select from a list of possible values when they create an instance of the content type. List fields are great when you already have a predefined list of options that can be selected from and the list is reasonably short. As always, we begin by setting the label, name, and type for the new field. When you select a type, you can choose between **List (numeric)** and **List (text)**. The difference between these two types is whether the values which the editor selects are stored in the database as numbers or text. Using a numeric list has a couple of advantages:

- Database size is minimized because numbers can be stored more compactly than text

- Translation can be easier because labels can be mapped more easily than text and the values are not dependent on any specific language

However, using a text list can make it easier to understand the values when you are looking at the database tables directly (that is, with phpMyAdmin).

> When you use text lists, try abbreviating each option so it is only a few characters long. This helps to make the options more understandable while still reducing the amount of space needed to store each option in the database.

For our example, we will label the field **Favorite Drupal 7 Features**, name it **favorite_ features**, give it a type of **List (numeric)**, and set the widget to **Select list**. After creating the label, you will need to set the allowable values list. With both types of lists, you will set the allowed values using the following format: **key|label**. For numeric lists, the key will be a number, and for text lists the key can be text. Here are the options for our sample field:

Allowed values list

```
1|Fields in Core
2|New Admin Interface
3|DBTNG
```

The possible values this field can contain. Enter one value per line, in the format key|label. The key is the value that will be stored in the database, and must be a *numeric* value. The label is optional, and the key will be used as the label if no label is specified. Allowed HTML tags in labels: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub> <sup> <tt>   <p>
 <img>
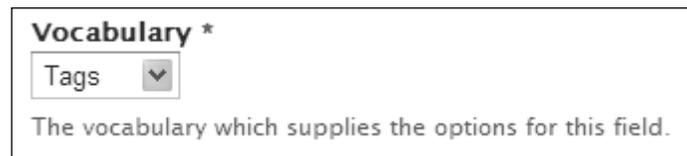
After setting your available values and saving them, you can set a few more options for the field. The first is a **Default value**, which can be used to automatically select a value if the editor creating the node does not select one. The other key setting is the **Number of values** that allows you to set how many selections the user can make from the list. This can be set from 1 to 10 or Unlimited. For example, if you wanted the user to pick their top 3 options, you could set this to 3.

In Drupal 6, you could accomplish this functionality using a CCK text field.

Term Reference fields

Term Reference fields blend taxonomies from Drupal 6 with the field system making it more intuitive for editors, administrators, and themers.

To add a Term Reference field, you will need to create a new taxonomy if you don't have one already. For this example, we will use the **Tags** taxonomy that is installed with Drupal 7. After you have a taxonomy ready, you are ready to create your field. Start by labeling and naming your field. Then, you will need to set the type to **Term Reference** and select a widget. We will use **Autocomplete term widget (tagging)** for now. After you save the new field, you can set the **Vocabulary** to use for the field:

The Term Reference field does not have any new advanced settings that you can apply to it other than setting the default value for the field.

You could achieve similar functionality in Drupal 6 using CCK and the Content Taxonomy module.

Text fields

Text fields are used to add textual information to a content type. Drupal 7 offers several different types of text fields that you can attach to content types including:

- Text
- Long text
- Long text and summary

Each of these types has specific capabilities and uses.

Text field

The text field is designed to store short pieces of text that are one to two lines long or less. The text field allows you to set the maximum length of the field:

Maximum length

255

The maximum length of the field in characters. Leave blank for an unlimited size.

The text field also allows you to set the size of the textfield, which controls the width of the text box that is used for inputting data. For example, if you set the maximum length of the text to 255 and the size of the textfield to 60, you will be able to see approximately 60 characters in the input box and will be able to add up to 255 characters of text. Drupal takes care of doing all of the validation for you.

☐ Required

    Check if a value must be provided.

Size of textfield *

60

Text processing

◉ Plain text

○ Filtered text (user selects input format)

The final option available for the text field is how to process the text. If you select **Plain text**, Drupal will ensure that the text is rendered as plain text and that special HTML characters are properly escaped. If you select **Filtered text**, the user will be able to select a text format from those available to them to apply to the text. Drupal will then automatically apply the text format when the text is rendered.

Long text field

The long text field is designed for use when you need more than a line or two of text. The long text field allows multiple lines of text to be entered and displayed. Much like the simple text field, the long text field allows you to set the maximum length for the text. By default, this is set to unlimited, but you can restrict the length to a shorter value if needed.

The long text field allows you to customize the number of **Rows** that are displayed in the text area when the long text is being edited as well as how text is processed.
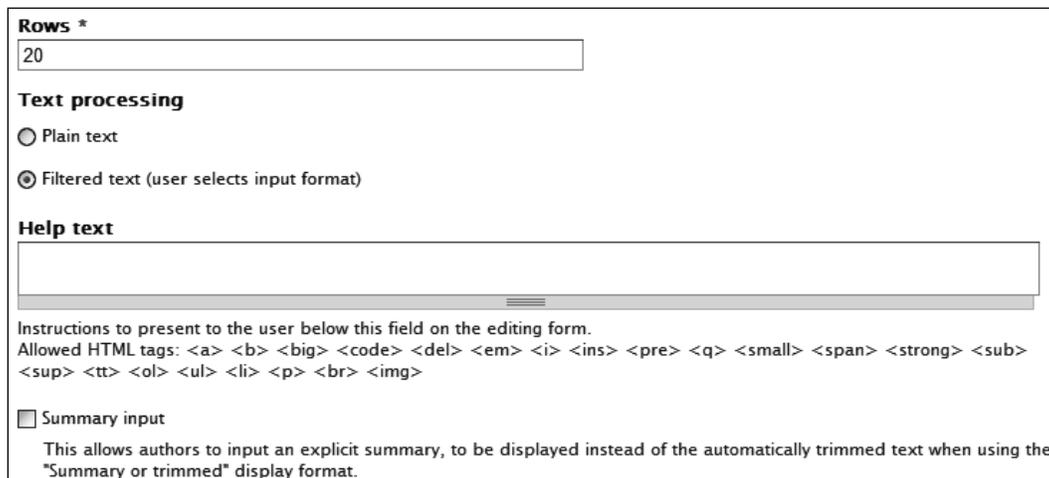
```
Rows *
5

Text processing
○ Plain text

◉ Filtered text (user selects input format)
```

You can set the number of rows to be displayed based on the expected length of the text to be entered. If the text will normally be longer, you may want to increase the number of rows displayed.

Long text and summary field

The long text and summary field allows you to add a long text field that also allows the user to enter a short summary for the field, similar to the way body fields work for standard nodes. The basic settings are identical to the long text field, and the advanced settings are similar as well.

```
Rows *
20

Text processing
○ Plain text

◉ Filtered text (user selects input format)

Help text


Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub>
<sup> <tt> <ol> <ul> <li> <p> <br> <img>

☐ Summary input
   This allows authors to input an explicit summary, to be displayed instead of the automatically trimmed text when using the
   "Summary or trimmed" display format.
```

The main differences between the long text and long text and summary fields are the number of rows is set to **20** rather than **5** by default. Also, text processing is set to **Filtered text** rather than **Plain text**, and there is a new **Summary input** setting. The **Summary input** setting determines whether or not authors can override the default trimmed text for the summary.

[82]

In Drupal 6, you could create text fields and long text fields using the Text CCK type. The long text and summary field is new to Drupal 7.

Field display

After you have built your content type, you will most likely want to modify the display of the fields. Luckily, Drupal 7 offers a great deal of control over how individual fields are displayed when the content type is displayed in various situations. You can modify the field display by editing your content type and then selecting the **MANAGE DISPLAY** tab. You can also access this page directly from the list of content types.

Drupal allows you to set the display format for four main display types: **Basic**, **Print**, **RSS**, and **Search**. Display types may be added or removed depending on what modules you have enabled. Each display type will have one or more sub views that can be configured. For example, the **Basic** display format allows you to configure the **Teaser** view and **FULL CONTENT** view, whereas the **Print** display format only allows the **Print** view to be configured.

> The **Print** display type is added when you enable the Book module.

In all cases, the configuration is similar. You are allowed to modify both the display of the label and the format of the value. The available settings for formatting the label are:

- **<Hidden>**: It causes the label to not be displayed
- **Above**: It causes the label to be displayed over the value
- **Inline**: It causes the label to be displayed to the left of the value

By default, the label is always displayed as bold text with a colon after it. This can be modified by your theme using CSS.

The available formats depend on the field type. Every field also allows **<Hidden>** as a valid value which, as the name implies, removes the field from the display. The following listing shows the available formats by field type with a brief description of what each format does:

Field type	Format
Boolean, List (text), List (numeric)	Default: It displays the value of the field (information to the right of the | symbol when defining allowable values).
	Key: It displays the key of the field (information to the left of the | symbol when defining allowable values).
Integer, Decimal, Float	Default: It displays the number formatted according to the rules defined in the field, including prefixes and suffixes.
	Unformatted: It displays the raw value of the field without formatting.
File	Generic file: It displays an icon and name for the file with a link, so the visitor can download the file.
	Table of files: It displays an icon for the file next to the name of the file which includes a link to download the full file. The size of the file is also included.
	All files are organized into a table.
	URL to file: The full URL to the file is displayed, but the file is not linked.
Term reference	Link: It displays each selected taxonomy term with a link to browse all nodes that use the term.
	Plain text: It displays the selected taxonomy term with no links to browsing-related nodes.

Attachment	Size
☐ 1st Chapter of Drupal 7 First Look	306.67 KB

Field type	Format
Text, Long text	Default: It displays the text formatted according to the field settings and selected text format.
	Plain Text: It displays the text without any formatting applied. HTML tags will be removed.
	Trimmed: It displays a shortened version of the formatted text if it is longer than a certain length.
Long text and summary	The long text and summary uses the same options as the text and long text types, plus it adds a Summary or Trimmed format, which displays the summary if it exists. If no summary has been entered, a trimmed version of the full text of the field will be displayed.

If this level of formatting is insufficient for your needs, you can always build your own template and create a custom theme for your content type. We will explore this more in *Chapter 5*, *Drupal 7 for Themers*.

Summary

In this chapter, we have reviewed some of the major changes and additions to Drupal 7 that relate to content management. We especially focused on the Field API, which replaced the CCK module that was available as a contributed module in Drupal 6. As we saw, Drupal 7 adds a great deal of new functionality that makes content management easier.

In the next chapter, we will be exploring changes in Drupal 7 that relate to administration of your Drupal 7 site.

Where to buy this book

You can buy Drupal 7 First Look from the Packt Publishing website:
`https://www.packtpub.com/drupal-7-first-look-at-new-features/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com