



Certification in Open Source Technologies (Drupal)

Sample Material

V-Skills Certifications

**A Government of India
&
Government of NCT Delhi Initiative**

V-Skills



1. WHY USE DRUPAL?

Drupal is a free and open source content management system (CMS) written in PHP and distributed under the GNU General Public License. It is used as a back-end system for over 1% of all websites worldwide ranging from personal blogs to corporate, political, and government sites including whitehouse.gov and data.gov.uk. It is also used for knowledge management and business collaboration.

The standard release of Drupal, known as Drupal core, contains basic features common to CMS. These include user account registration and maintenance, menu management, RSS-feeds, page layout customization, and system administration. The Drupal core installation can be used as a brochureware website, a single- or multi-user blog, an Internet forum, or a community website providing for user-generated content.

Over 7000 free community-contributed addons, known as contrib modules, are available to alter and extend Drupal's core capabilities and add new features or customize Drupal's behavior and appearance. Because of this plug-in extensibility and modular design, Drupal is sometimes described as a content management framework. Drupal is also described as a web application framework, as it meets the generally accepted feature requirements for such frameworks.

Although Drupal offers a sophisticated programming interface for developers, no programming skills are required for basic website installation and administration.

Drupal runs on any computing platform that supports both a web server capable of running PHP 4.4.0+ (including Apache, IIS, Lighttpd, and nginx) and a database (such as MySQL, PostgreSQL or SQLite) to store content and settings. Drupal 7 requires PHP 5.2 or higher.

Easy to use - no programming knowledge needed! Read this tutorial to learn the basics of Drupal. Lots of features including Search Engine Friendly URLs(SEF), categories, search function and many more; Lots of modules to extend your site's functionality; Flexibility - you can easily turn your Drupal installation into a forum, blog, wiki and many other types of web sites; Free to use, open source. You can freely install Drupal and you can modify the source code to fit your needs; Lots of users and a large community - easy to find solutions to your problems. Latest version of Drupal is 7.0

1.1 Drupal Overview

Design is driven by the need to balance flexibility and simplicity. If a system is too simple, it can only be used for a single purpose - but if it is too flexible, it may be too difficult for new users to learn.

Drupal bridge conflicting goals by providing its users with the tools they need to make their own content management solution, while still providing some prebuilt components to help them get started. Thus, it can be described both as a content management system (CMS) and a content management framework (CMF) - one system which strives to have the strengths of both, without their deficiencies.

Certified Open Source CMS (Drupal) Professional

Most CMS are like a toy boat or toy truck - specific assumptions have been made about their use, assumptions it would be hard for you to override. Frameworks, on the other hand, provide you with raw materials only - you need to know a programming language, and have a clear design vision, to put them together.

Drupal is set of building blocks. Building blocks are already created- in the form of contributed modules - that you need to create a site that suits your needs, whether that is a news site, an online store, a social network, blog, wiki, or something else altogether.

Drupal core, and the thousands of contributed modules that build on it, require an initial investment to learn, but mastering the Drupal way is immensely rewarding. Drupal's passionate community is a testament to its power to liberate site builders from the simplicity/flexibility dilemma.

How Drupal Works?

People often think of a website as a collection of static pages, with some functions (like a blog, facebook, twitter, RSS News Feed etc) around the pages. At the time of maintenance there emerges a need of structured tree, nodes path defined websites.

Drupal, on the other hand, treats different content types as: a node. Static pages, blog posts, and news items (some possible node types) are all stored in the same way, and the site's navigation structure is designed separately by editing menus, views - is the content view, and blocks - supplementary content which is linked to this main content somehow.

It's a lot like the separation you find in standards-compliant page coding like XHTML or XML provides the structure of the data, while corresponding style sheets applies the presentation of the data. Nodes hold the structured information pertaining to a blog post (such as title, content, author, date) or a news item (title, content, go-live date, take-down date), while the menu system, as well as taxonomy (tagging of content) and views, create the information architecture. Finally, the theme system, along with display modules like Panels, controls how all this looks to site visitors.

As there are different layers, there can be completely different presentation and navigation schemes per user, in other words each user can customize their own features, navigations, etc.

What are Nodes

Node is core to provide the flexibility in Drupal.

We don't talk about "nodes" every day, but since they are at the heart of Drupal's design, they deserve further investigation. At its most basic, a node is a set of related information. When you create a new blog post, you are not only defining its body text, but also its title, content, author link, creation date, taxonomy (tags), etc. Some of these elements will be shown by the theme layer when the node is displayed. Others are meta-data that control when the node will show up at all - such as taxonomy or publishing status.

Since each item of content is stored as a node, and contains the same basic information, each can be handled in a standard way by both Drupal core and contributed modules. This allows site builders to choose exactly where they want content to show up, and exactly how they want it to look in each case. Most of a Drupal site builder's time is spent defining what kinds of information

Certified Open Source CMS (Drupal) Professional

you want to store in your nodes, and configuring the structures (menus, taxonomy trees, views, panels) in which to display them.

As suggested before, you aren't limited to a single way of presenting your site's content. You can define as many navigation schemes, custom themes ("skins" for the site), blocks, and feature sets as there are distinct audiences for your site.

Comments are second-class citizens in Drupal compared to nodes, but they also illustrate the Drupal way. Comments aren't just part of the blog system, since there isn't a separate "blog system." Comments can be enabled on any node type you choose - blog posts, news items, book pages and any other you may create.

Collaborative at the core

Creating an informational website that broadcasts from "one to many" is something that most CMSs do right out of the box. However, where Drupal really shines is when you want to empower site users to create content, and connect with each other - moving from "one to many" to "many to many."

With some CMS's, you can set up a blog, and you can install plugins to handle having a community of users, but what happens when you want to give individual blogs to each of your users, sorting their contents so that they can be displayed individually with their own skins, while also generating cross-blog topical digests, top five lists, and links out to elaborate, customized user profiles? What if you want to also integrate that in with forums, a wiki-like environment, and give each user their own gallery of taggable photos?

Drupal is designed from the ground up so site builders can delegate content creation, and even site administration, to users. All you have to do is define who gets to do what on your site (through user permissions), and then you can start collaborating.

Get started quickly, customize extensively

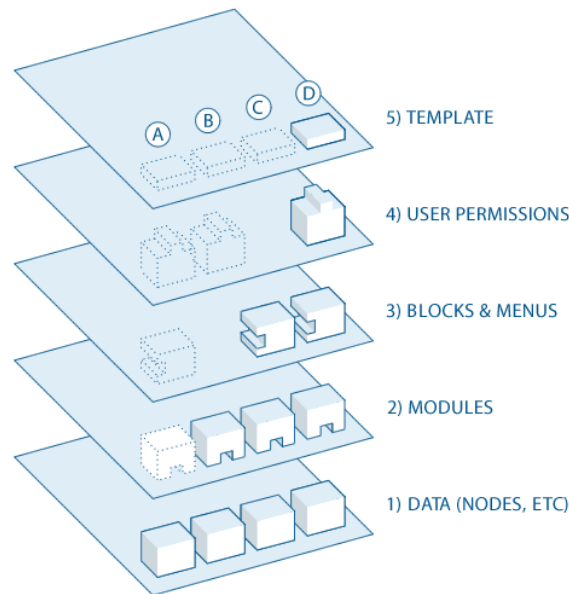
Drupal's flexibility is incredible, but installing it is surprisingly easy. With a simple FTP upload and a few short web-based configuration questions, you can connect with your database and have your first Drupal site up and running within an hour.

Pick one of the included themes, and just start adding content. Do you want to have visitors log in? Switch authentication on or off. Want to switch on some of the included tools? Turn on forums; enable commenting on node types; turn on the book module for wiki-like collaboration; create forums and polls; use taxonomy to give site content structured, hierarchical categorization or free-form tagging.

Do you want your own skin applied to the site? Drupal's theme system uses tiny snippets of PHP that you can insert into the appropriate spots in your design to replace your placeholder text with dynamic content. Drupal's generated markup is clean, standards-compliant XHTML. No old-school tables.

1.2 The Drupal flow

If you want to go deeper with Drupal, you should understand how information flows between the system's layers. There are five main layers to consider



- ✓ At the base of the system is the collection of nodes—the data pool. Before anything can be displayed on the site, it must be input as data.
- ✓ The next layer up is where modules live. Modules are functional plug-ins that are either part of the Drupal core (they ship with Drupal) or they are contributed items that have been created by members of the Drupal community. Modules build on Drupal's core functionality, allowing you to customize the data items (fields) on your node types; set up e-commerce; programmatically sorting and display of content (custom output controlled by filters you define); and more. There are thousands of different options within the fast-growing repository of contributed Drupal modules. They represent the innovation and collaborative effort of everyone from individuals to large corporations.
- ✓ At the next layer, we find blocks and menus. Blocks often provide the output from a module or can be created to display whatever you want, and then can be placed in various spots in your template (theme) layout. Blocks can be configured to output in various ways, as well as only showing on certain defined pages, or only for certain defined users.
- ✓ Next are user permissions. This is where settings are configured to determine what different kinds of users are allow to do and see. Permissions are defined for various roles, and in turn, users are assigned to these roles in order to grant them the defined permissions.
- ✓ On the top layer is the site theme (the "skin"). This is made up predominantly of XHTML and CSS, with some PHP variables intermixed, so Drupal-generated content can go in the appropriate spots. Also included with each theme is a set of functions that can be used to override standard functions in the modules in order to provide complete control over how the modules generate their markup at output time. Templates can also be assigned on-the-fly based on user permissions.

Self Assessment Exercise

Q.1 Drupal is a ?

- a. Open Source
- b. CMS
- c. CMF
- d. All of the above

Q.2 What is a node?

- a. Content
- b. Content Type
- c. Content Creator
- d. Themes

Q.3 Drupal architecture is layered as following from bottom to top?

- a. Data, Modules, Menus, Permissions, Templates
- b. Templates, Permissions, Menus, Modules, Data
- c. Data, Permissions, Menus, Modules, Templates
- d. Permissions, Data, menus, modules, Templates

Q.4 Top layer of Drupal is driven by?

- a. XHTML
- b. XHTML and CSS
- c. Menus
- d. Users

Q.5 In which scenario Drupal should not be preferred?

- a. Real Time applications
- b. Ecommerce
- c. Retail Based
- d. Content Based

Answers - 1-d, 2-b, 3-a, 4-b, 5-a

1.3 General Concepts

Content Management System (CMS)

A content management system (CMS) is the Collection of Procedures used to manage work flow in a collaborative environment. These procedures can be manual or computer-based. The procedures are designed to do the following

Allow for a large number of people to contribute to and share stored data

- ✓ Control access to data, based on user roles (defining which information users or user groups can view, edit, publish, etc.)
- ✓ Aid in easy storage and retrieval of data
- ✓ Reduce repetitive duplicate input
- ✓ Improve the ease of report writing
- ✓ Improve communication between users

In a CMS, data can be defined as nearly anything: documents, movies, pictures, phone numbers, scientific data, and so forth. CMSs are frequently used for storing, controlling, revising, semantically enriching, and publishing documentation. Serving as a central repository, the CMS increases the version level of new updates to an already existing file. Version control is one of the primary advantages of a CMS.

Types of CMS

Enterprise content management

An enterprise content management system (ECM) is content, documents, details and records related to the organizational processes of an enterprise. The purpose and result is to manage the organization's unstructured information content, with all its diversity of format and location. The system manages the content related commercial organizations

Web content management systems

A web content management (WCM) system is a CMS designed to simplify the publication of web content to web sites and mobile devices—in particular, allowing content creators to submit content without requiring technical knowledge of HTML or the uploading of files. Several web-based content management systems exist both in the Open Source and commercial domains.

Web group content management systems

The web group content management system (mostly known as GMS) is very similar to the WCM, with the main difference being the fulfillment of some specific requirements. These requirements aim to satisfy the needs of groups or small organizations that often have difficulty managing their online image, communicate with their members, plan events, set schedules and assign tasks to internal projects. Typically these tasks are distributed by different tools and the GMS unifies them into one.

Component content management system

In a component content management system, the content is stored and managed at the sub-document (or component) level for greater content reuse.

Drupal Concepts

Module

A module extends Drupal features and/or functionality. Core modules are those included with the main download of Drupal, and you can turn on their functionality without installing additional software. User modules can be created and installed; this requires a thorough understanding of Drupal, PHP programming, and Drupal's module API.

User, Permission, Role

Every visitor to your site, whether authorized or anonymous, is considered a *user* to Drupal. Each user has a numeric user ID, and non-anonymous users will have proper credentials of user id and email, etc. Other information can also be associated with users by modules; for instance, if you use the core Profile module, you can define user profile fields to be associated with each user.

Anonymous users have a user ID of zero (0). The user with user ID one (1), which is the user account you create when you install Drupal, is special: that user has permission to do absolutely everything on the site.

Other users on site can be assigned permissions via *roles*. To do this, first need to create a role, which you might call "Content editor" or "Member". Next, you will assign *permissions* to that role, to tell Drupal what that role can and can't do on the site. Finally, you will grant certain users on your site your new role, which will mean that when those users are logged in, Drupal will let them do the actions you gave that role permission to do.

You can also assign permissions for the special built-in roles of "anonymous user" (a user who is not logged in) and "authenticated user" (a user who is logged in, with no special role assignments). Drupal permissions are quite flexible -- you are allowed to assign permission for any task to any role, depending on the needs of your site.

Node

A *node* in Drupal is the generic term for a piece of content on your web site. Some examples of nodes

- ✓ Pages in books
- ✓ Discussion topics in forums
- ✓ Entries in blogs
- ✓ News article stories

Each node on your site has a Content Type. It also has a Node ID, a Title, a creation date, an author (a user on the site), a Body (which may be ignored/omitted for some content types), and some other properties. By using modules such as the contributed Content Construction Kit

Certified Open Source CMS (Drupal) Professional

(CCK) module, the core Taxonomy module, and the contributed Location module, you can add fields and other properties to your nodes.

Comment

Comments are another type of content you can have on your site. Each comment is a typically small piece of content that a user submits, attached to a particular node. For example, each piece of discussion attached to a particular forum topic node is a comment.

Taxonomy

Drupal has a system for classifying content, which is known as *taxonomy* and implemented in the core Taxonomy module. You can define your own *vocabularies* (groups of taxonomy terms), and add terms to each vocabulary. Vocabularies can be flat or hierarchical, can allow single or multiple selection, and can also be "free tagging". Each vocabulary can then be attached to one or more content types, and in this way, nodes on your site can be grouped into categories, tagged, or classified in any way you choose.

Database

Drupal stores information in a database; each type of information has its own *database table*. For instance, the basic information about the nodes of your site is stored in the Node table, and if you use the CCK module to add fields to your nodes, the field information is stored in separate tables. Comments and Users also have their own database tables, and roles, permissions, and other settings are also stored in database tables.

Path

When you visit a URL within your Drupal site, the part of the URL after your base site address is known as the *path*. When you visit a path in your Drupal site, Drupal figures out what information should be sent to your browser, via one or more *database queries*. Generally, Drupal allows each module you have enabled on your site to define paths that the module will be responsible for, and when you choose to visit a particular path, Drupal asks the module what should be displayed on the page.

Theme

The *theme* controls how your site is displayed, including the graphic look, layout, and colors. A theme consists of one or more PHP files that define the HTML output of your site's pages, along with one or more CSS files that define the layout, fonts, colors, and other styles.

Region, Block, Menu

Pages on your Drupal site are laid out in *regions*, which can include the header, footer, sidebars, and main content section; your theme may define additional regions. *Blocks* are discrete chunks of information that are displayed in the regions of your site's pages. Blocks can take the form of *menus*, the output from modules (e.g., hot forum topics), or dynamic and static chunks of information that you've created yourself (e.g., a list of upcoming events).

There are three standard menus in Drupal: **Primary Links**, **Secondary Links**, and **Navigation**. Primary and Secondary links are built by site administrators, and displayed automatically in the page header of many themes (if not, you can enable their blocks to display them). Navigation is the

catch-all menu that contains your administration menus, as well as links supplied by modules on your site. You can also create your own custom menus, and display them by enabling their blocks.

You can customize menus in several ways, such as reordering menu items by setting their “weight” or simply dragging into place, renaming menu items, and changing the link title. You can move a menu item into a different menu by editing the Parent property of the menu item.

You can also add custom menu items to a menu, from the Add menu item tab of the Menu administration screen. To create a menu item, you will need to provide the path to the content, explained above.

In all cases a menu item will only be shown to a visitor if they have the rights to view the page it links to; e.g., the admin menu item is not shown to visitors who are not logged in.

For terminology of Drupal terms in detail please visit - <http://drupal.org/node/937>

1.4 Content Types

A single web site could contain many types of content, such as informational pages, news items, polls, blog posts, real estate listings, etc. In Drupal, each item of content is called a node, and each node belongs to a single content type, which defines various default settings for nodes of that type, such as whether the node is published automatically and whether comments are permitted. (Note that in previous versions of Drupal, content types were known as node types.)

When you first install Drupal with the default installation profile, you will have two content types defined: "Page" and "Story". When you enable other core and contributed modules (by visiting Administer >> Site Building >> Modules), you will find that you have other content types available; you can also create your own content types. Here is a list of the content types associated with core Drupal modules

Blog Entry

A Blog (short for weblog) is an online journal or diary, and the core Blog module allows registered users on your site to create their own blogs. Each entry in a user blog has content type Blog Entry.

Book Page

Book pages are designed to be part of a collaborative book, enabled by the core Book module. In older versions of Drupal, only nodes of content type Book Page could be added to a book, but now nodes of any content type can be part of a book.

Comment

Comments actually aren't nodes, so Comment is technically not a "content type". Enabling the Comment module allows site visitors to add comments (typically short notes and replies to other comments) to nodes on the site.

Forum

A Forum node defines a topic for a forum discussion; people can reply to the topic by using comments. Forum nodes are organized into subject areas via Taxonomy (list of categories).

Page

The Page content type is enabled in Drupal in the default installation profile. Typically Pages are used for static content that can (but are not required to) be linked into the main navigation bar.

Poll

A poll is where a multiple choice question is asked, and users can answer and see other people's answers to questions.

Story

The Story content type is enabled in Drupal in the default installation profile. Stories are generally used for information whose relevance decreases as time passes (such as news items), so that newer Stories will typically be placed higher on the page than older Stories.

Related link to the Drupal Content Construction Kit is as follows: <http://drupal.org/project/cck>

1.5 Drupal-viewing/editing a page

In normal content or website design softwares like Frontpage, etc. there is a feature of looking at the preview the changes performed on the content or to the layouts. In Drupal there is nothing as such preview

- ✓ Drupal just provides a unified interface.
- ✓ Looking at the website, you are previewing it. To get the experience of a anonymous user, log out and experience.
- ✓ Drupal usually looks pretty much the same to all browsers as it does to editors, only with different features and menu items available.
- ✓ Authenticated users with appropriate permissions will see the 'edit' tabs above their pages. That's often the only difference between editor and user experience.

1.6 Understanding Drupal paths

In Drupal terms, a path is the unique, last part of the URL for a specific function or piece of content. For instance, for a page whose full URL is <http://somesite.com/?q=node/7>, the path is `node/7`. If site is using clean URLs(<http://drupal.org/node/15365>), the full URL in this example would be <http://somesite.com/node/7>; the path would still be `node/7`. Because URL aliases(<http://drupal.org/handbook/modules/path>) can completely replace what visitors see as the URL, the paths discussed here (which are still how Drupal decides what content to show) are sometimes called internal paths. Usually Clean URL's and internal paths are very effective way to create the Search Engine Friendly URL.

Drupal paths are important because many configuration screens in the Drupal admin area use them. For instance, when you are adding a new item to a menu, you tell Drupal what page the menu item should point to by entering the path to the page. Here are some examples of paths you might find in a Drupal site

- ✓ node/7
- ✓ taxonomy/term/6
- ✓ admin/content/comment
- ✓ user/login
- ✓ user/3

URL can have several forms

- ✓ [http://somesite.com/?q=\[something\]](http://somesite.com/?q=[something]) – In this case, the [something] after ?q= is the path. For example, if the URL is <http://somesite.com/?q=node/7>, the path is node/7.
- ✓ [http://somesite.com/\[something\]](http://somesite.com/[something]) or [http://somesite.com/\[your Drupal subdirectory\]/ \[something\]](http://somesite.com/[your Drupal subdirectory]/[something]) – In this case, the [something] after the base path of your Drupal site is the path. For example, your URL could be <http://somesite.com/node/7> or <http://somesite.com/mysubdir/node/7>; the path in either case is node/7.

How to find Drupal paths

There are several ways to find the path to a particular page on your Drupal site. If you know how to navigate to the page, you can go there and find the URL in your browser's URL bar.

- ✓ Roll mouse over a link to the page, and most browsers will show the URL in the status section at the bottom of the browser window.
- ✓ For Taxonomy term pages, find the URL in the Taxonomy administration page Administer > Structure > Taxonomy. Each term should be a link to its taxonomy page. Roll Mouse over the link or follow the link to find its URL.

1.7 User Roles and Permissions, Kind of users

Differentiating between different users:

In order to remove any confusion in user regards in terms of Drupal installation. This is important to understand as it will help plan the installation of Drupal and remove any confusions, in terms of users, as they may arise.

Different Users

Computer User

The person installing Drupal must have access to the computer where the installation lives. The individual installing Drupal on local system is Computer User. If you are installing Drupal at a remote server such as a web hosting, then the computer user account could be called one of the following

- ✓ Web hosting account login
- ✓ FTP login
- ✓ Cpanel login
- ✓ Secure shell (SSH) login

Who are the people who are associated with the computer user? In some cases, like for your own computer or for some FTP logins, these "users" are individual people. In other cases, such as a web hosting account login or Cpanel, one user (or "account") name/password is used to log in to control the account and may be shared by several people.

Database User

Drupal installation requires a database to run. (MySQL and PostgreSQL are the best supported database programs compatible with Drupal.) Databases typically have access control mechanisms and require "users" with the right permissions to change the database (add/remove/edit data, create tables, etc.). A Drupal installation needs to have full control over the database, so when you set up a new Drupal site, you (or your web hosting company) create a database user with full privileges and then give the user name and password to the Drupal installation so that Drupal will have full control over the database. The database user information is stored in the settings.php file, which is either in sites/default or a different subdirectory of sites in your Drupal installation. Who is the database user? The database "user" is not a person. It is an account created with the database software in order to give Drupal control of the database.

User/1

"User/1", also known as the "maintenance" account or "super-user account" is the Drupal account you are prompted to create immediately after you have successfully installed a new Drupal site. This account is unique to your site (it doesn't have anything to do with Drupal.org or any other web site). This account is different from all other users in a Drupal installation because it has no permissions limitations ("permissions" were called "access control" in Drupal 5 and before). "User/1", in Drupal 6 and before, is also the only account that can launch the update.php script that you need to run after a software upgrade. Who is user/1? User/1 shouldn't be associated with an individual person, but rather with the person or persons who have the responsibility for keeping software up-to-date on your site.

It is a best practice to avoid creating site content with user/1. That is so because it is awkward when responsibility for site maintenance done as User/1 needs to change to a new person if the original User/1 wrote content that still needs to be associated with him/her. The content written by the original author would have to then be assigned to a new user account. It's better to simply create that second account immediately after installing the site.

User/2 and all others

User/2 and all other registered users on your Drupal site should each be associated with an individual person. Registered users can be assigned to roles, which are given fine-grained permissions to allow different users different access to administer the site and add content.

User management module User rights are managed using Administer->User management.

User management is broken down to 5 sections:

- ✓ Access rules - Define different access rules for Drupal based site; allow/disallow users, emails and IP addresses
- ✓ Permissions - Define Modules availability for anonymous and/or authenticated users

Certified Open Source CMS (Drupal) Professional

- ✓ Roles - Drupal comes with two default user roles - anonymous user and authenticated user; you can add additional roles such as administrator, moderator, etc. as per the requirement of the content and process workflows.
- ✓ User settings - Configure the user registration settings, the emails notifying a user for new registration, password recovery, etc
- ✓ Users - shows a list of your current users; you can also add new users or block existing ones

1.8 Technology Used in Drupal

Below are the pieces of Drupal technology stack, these are the technology essentials for Drupal:

Server - A server is a computer which provides information or services to other computers on a network. Usually contains PHP, Apache, Database(MySQL)

Operating system - The software that runs the server. Unix, Linux, BSD, OS X and Windows are some examples. This OS decides the complete infrastructure for the site. i.e. it will be LAMP based or Windows based.

Database - A structured collection of records. Drupal uses a database to store most content and configuration settings for drupal based site, some content such as media files are generally stored in the server's file system.

Web server - The software component responsible for serving web pages. Examples are Apache and Microsoft IIS.

PHP - PHP is a programming language that allows web developers to create dynamic content that interacts with databases.

Drupal - A framework for building dynamic web sites offering a broad range of features and services include user administration, publishing workflow, discussion capabilities, news aggregation, and metadata functionalities using controlled vocabularies and XML publishing for content sharing purposes. A Drupal installation is generally comprised of a mix of core and contributed modules.

Self Assessment Exercise

Q.1 In open source Platform what are technology required for Drupal to Run?

- a. LAMP
- b. IIS
- c. UNIX
- d. PHP, MYSQL, Linux

Q.2 CMS is understood as ?

- a. Set of integrated workflow
- b. Set of integrated workflow to Manage Content
- c. Collection of Content
- d. Collaboration of Content

Q.3 Node is the backbone of Drupal, why?

- a. Provides the granularity and Flexibility of creating content
- b. Provides the API
- c. Provides generic environment
- d. Provides Different Content Types

Q.4 User/1 is a super user because?

- a. It is the first default user created in Drupal
- b. It is the maintenance account
- c. It is the admin
- d. It is Creator of the Administrator

Q.5 Taxonomy is?

- a. Document Classifier
- b. Tax creator
- c. Document indexer
- d. Document Generator

Comments in Drupal is?

- a. Is a content type
- b. Is not a content type
- c. Is not a node
- d. Both b and c

Answers - 1- a, 2-b, 3-a, 4-b, 5-a, 6-d

1.9 Drupal Security

Drupal being an Open Source CMS, many people have this question, whether content, and site itself are secure on Drupal. Answer to which is, world's most secure websites like White House switched to Drupal. Issues like Injection, Cross Site Scripting, Session Management, Cross Site Request Forgeries, and others all of standard solutions in the Drupal API.

Professional security audits of Drupal sites have generally found that the vast majority of security holes (90% or more) are present in the custom theme or modules written by that site's developers. That code did not get the same public scrutiny that all code on drupal.org receives.

In addition, problems at the server level (such as using insecure protocols like FTP) are more likely to be the means of a successful attack than vulnerability in Drupal - especially Drupal core.

1.10 Drupal versions and their numbering

At any given time, there are two major *release series* of Drupal which are supported. Currently, these are Drupal 5 and Drupal 6. Updated versions of each of these are issued on a regular basis. For example, within the Drupal 6 series, several versions have been and will continue to be released: 6.0, 6.1, 6.2, etc. These versions are sometimes collectively referred to as "6.x"; however, they are not all equal. Version 7 will be released in January, 2011.

How to Know which version is running? Administer >> Reports >> Status report. This will list your version number if you have Drupal 6.0 or later. In Drupal 5.x and earlier, the path to go to is Administer >> Logs >> Status report.

Failing that, look for a file called CHANGELOG.txt in the root of your Drupal directory and open it up to find the version you are running.

If CHANGELOG.txt is missing, you can also check in system.module for a line at the top like: `define('VERSION', '5.5');`

In deciding between Drupal 6 and Drupal 7, several factors should be taken into consideration. Drupal 7 is newer, will be supported longer, and contains more features and enhancements than Drupal 6; therefore, if all else is equal, use Drupal 7. However, the releases of contributed modules, themes, translations, and installation profiles compatible with each major Drupal release series sometimes lag behind Drupal itself, so if the functionality you need is not yet available for Drupal 7, this may affect your decision.

Drupal 8 is currently in development and is not yet ready for production use. When version 8.0 is released, Drupal 6 will no longer be supported or receive security updates, and you are strongly advised to upgrade your Drupal 6 sites to Drupal 7 at that time.

New stable releases of Drupal core are scheduled to occur within particular release "windows":

- ✓ A bug fix release window is on the first Wednesday of each month.
- ✓ A security release window is on the third Wednesday of each month.

A release window does not necessarily mean that a release will actually be made on that date, but it exists so that site administrators can know in advance which days they should be on the look out for one.

How do I know which version of a module or theme I'm running?

You can see the versions of your installed modules and themes on the modules and themes administration screens. Go to **Administer >> Site building >> Modules** or **Administer >> Site building >> Themes**, and you will see a column listing the version number.

Version Number Nomenclature

The version number indicates the version of Drupal core the contribution is compatible with, whether it is a stable or development release, and what specific "patch level" of the code it represents. These numbers have the form

CoreCompatibility-Major.PatchLevel[-Extra]

- ✓ **CoreCompatibility** is required, and will be something like "4.7.x", or "5.x", to indicate what version of Drupal core this contribution is compatible with. Since the API for Drupal core does not change across stable releases, a module that is compatible with Drupal core 4.7.1 would also work with 4.7.3. This is why the version strings for contributions contain the character 'x' to show that they are compatible with any core release ("4.7.x") not just a specific version of core ("4.7.1").
- ✓ **Major** is the major revision number, which indicates what set of functionality a given release has, if it is stable (bug-fixes only) or development (new features), and so on. The default branch for a given release of core (for example, the "DRUPAL-4-7" branch) will be major version 1. By convention, this major revision should always remain stable, and only security patches and bug fixes should be added to it.
- ✓ **PatchLevel** is for a specific release of a given major revision. The first release would be patch level 0. As bugs are fixed, each new releases would have a higher patch level number. If the patch level is the letter "x", it indicates a development snapshot.
- ✓ **[-Extra]** is optional, and is for specifying things like "-dev", to indicate a development snapshot from the end of a CVS branch (as opposed to an official release from a CVS tag). These snapshots, by their nature, include changing code. It is therefore hard to know exactly what revisions of each file they contain, and this makes them more difficult to debug. Development snapshots also use "x" for the patch level to further indicate the variable nature of the code they contain.

1.11 Third party resources available

There are lot of 3rd party sources available which provide drupal modules, information and functionality. There is a big Drupal Community i.e. Groups.Drupal.org. Lot of resources are available through Drupal Planet i.e. Drupal.org/planet.

1.12 Drupal Installation

Drupal provides an installation script that automatically populates database tables and configures the correct settings in the settings.php file. This section covers preparing for installation, running the installation script itself, and the steps that should be done after running the installation script has completed. It also explains how to do a "multi site" installation, where a number of different Drupal sites run off the same code base.

Certified Open Source CMS (Drupal) Professional

Before installing please go through the Drupal Terminology. Below are the best practices to be followed before proceeding:

If you are going to invest the time to set up a CMS, then you should protect your investment by following some simple best practices. The following list contains some quick pointers

- ✓ **Plan your site. Drupal** provides a good toolset to help you build your site but you still need to plan. Good wireframes and proper planning can avoid significant misunderstandings and problems later.
- ✓ **Plan for the future.** You should revisit and reevaluate your site each time there is a major version release of Drupal. This does not mean you have to upgrade it, but you should evaluate and plan for an upgrade approximately each 12-24 months.
- ✓ **Get involved in the community.** This will help you follow development trends and, while helping others, you may just come across a cool idea that solves your own problem.
- ✓ **Back up your site.** Back up both the database and the files on the web server. Test your backups! If you don't test them, you have no idea if you are doing it right.
- ✓ **Test your PHP snippets.** Drupal gives you a great deal of power and flexibility when using PHP code in blocks. Unfortunately, a stray character or a missing semi-colon breaks PHP. Drupal then attempts to evaluate this broken code on any requested page, the PHP interpreter chokes on it and therefore your whole site is broken. Fortunately, there is a very simple and easy solution. Instead of writing and testing your code inside the administer > blocks page, go to create content and create a new story or page node. Use PHP input format, write the code, and the Preview to debug your code. When you are satisfied that your code is working, copy and paste the code into the block.

System requirements

A minimum base installation requires at least 3MB of disk space but you should assume that your actual disk space will be somewhat higher. For example, if you install many contributed modules and contributed themes, the actual disk space for your installation could easily be 40 MB or more (exclusive of database content, media, backups and other files). Extra space for webserver, PHP and database will be required, this will be evident at the time of installation of a particular component.

1.13 Web server

Drupal has been deployed successfully on both Apache and IIS.

Apache (Recommended)

- ✓ Drupal will work on Apache 1.3 or Apache 2.x hosted on UNIX/Linux, OS X, or Windows. The majority of Drupal development and deployment is done on Apache, so there is more community experience and testing performed on Apache than on other web servers.
- ✓ You can use the Apache 'mod_rewrite' extension to allow for clean URLs.

Microsoft IIS

Drupal core will work using IIS 5, IIS 6, or IIS 7 if PHP is configured correctly.

To achieve clean URLs you may need to use a third party product. For IIS7 you can use the Microsoft URL Rewrite Module or a third party solution.

When using Drupal on IIS 7 with fastcgi you must install Hotfix kb954946, or wait until the hotfix appears in a package update (recommended). KB954946 was included in Windows 2008 Server SP2

Drupal is being developed to be web server independent, but we have limited or no reports of successful use on web servers not listed here.

The total file size of your Drupal installation will depend on what you add to your site, but Drupal core files alone will take up approximately 2 to 3 MB uncompressed. The exact size depends on the version of Drupal you have installed.

1.14 Database Server

Recommended: MySQL

- ✓ Drupal 5.x and earlier supports MySQL 3.23.17 or higher. MySQL 4.1 or higher is strongly recommended.
- ✓ Drupal 6 supports MySQL 4.1 or higher.
- ✓ Drupal 7 will only support MySQL 5.0.15 or higher, and requires the PDO database extension for PHP.

NOTE: Drupal makes use of some features not available on some inexpensive hosting plans so please check that your host allows database accounts with the following rights

SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER.

These rights are sufficient to run Drupal core 6.x and later versions.

Some contributed modules, and also Drupal core 5.x (but not Drupal core 6.x), additionally require the following rights

CREATE TEMPORARY TABLES, LOCK TABLES.

PostgreSQL

Note: Some contributed modules are not as abstracted from MySQL-specific code as everyone would like. If you are familiar with PostgreSQL please file issues with those contributed modules as you find them.

- ✓ Drupal 6 supports PostgreSQL 7.1 or higher
- ✓ Drupal 7 will only support PostgreSQL 8.3 or higher
- ✓ PHP 5.2.6 for Windows has a bug in its pgsql extension. You will need to replace it with the php_pgsql.dll from version 5.2.5.

SQLite 3.x - Drupal 7 only

Note: Some contributed modules are not as abstracted from MySQL-specific code as everyone would like. If you are familiar with SQLite please file issues with those contributed modules as you find them.

Other database engines

Microsoft SQL Server and Oracle are supported by an additional module. Not very successful.

PHP

Recommended: PHP 5.2 for Drupal 5 and 6, PHP 5.3 for Drupal 7

Required: PHP version 4.4.0 or higher for Drupal 5 and Drupal 6, PHP 5.2.5 or higher for Drupal 7

- ✓ PHP 5.3 is not yet supported by Drupal 5.x, but is supported by Drupal 6.14 core and higher (see the release notes for 6.14) and also by Drupal 7.x. Note that some contributed modules may not be compatible with PHP 5.3, and that some PHP 5.3 configurations still show warnings with Drupal 6.14.
- ✓ PHP 5.2.5 or higher is a requirement for Drupal 7. (Distributions of PHP 5.2.4 that include backported security patches, such as the version included in Ubuntu Hardy, should also work with Drupal 7.)
- ✓ PHP memory requirements can vary significantly depending on the modules in use on your site. Drupal 6 core requires PHP's `memory_limit` to be at least 16MB, and Drupal 7 core requires 32MB. Warning messages will be shown if the PHP configuration does not meet these requirements. However, while these values may be sufficient for a default Drupal installation, a production site with a number of commonly used modules enabled (CCK, Views etc.) could require 64 MB or more. Some installations may require much more, especially with media-rich implementations. If you are using a hosting service it is important to verify that your host can provide sufficient memory for the set of modules you are deploying or may deploy in the future.
- ✓ The PHP extension for connecting to your chosen database must be installed and enabled. Drupal's currently supported database connectors are: `mysql` (the original MySQL extension), `mysqli` (an improved connector for newer MySQL installations), and `pgsql` (for PostgreSQL). Drupal 6.x does not provide the option to select the `mysql` connector if `mysqli` is enabled in your PHP configuration.
- ✓ PHP XML extension required for blogging API, drupal. This extension is enabled by default in a standard PHP installation; the Windows version of PHP has built-in support for this extension. Enabling the XML extension also enables PHP DOM. DOM is now a systems requirement.
- ✓ An image library for PHP such as the GD library is needed for image manipulation (resizing user pictures, image and image cache modules). GD is included with PHP 4.3 and higher and usually enabled by default. ImageMagick is also supported for basic image manipulations in Drupal core but there is much less support from contributed modules.

Certified Open Source CMS (Drupal) Professional

- ✓ If you have administrator rights, and GD is not already available (look in phpinfo) it can usually be installed by going:
- ✓ `sudo apt-get install php5-gd`
- ✓ or on Redhat/Centos:
- ✓ `sudo yum install php-gd`
- ✓ PHP needs the following configuration directives for Drupal to work (only directives that differ from the default `php.ini-dist` / `php.ini-recommended`):
 - ✓ `register_globals: off`; this is the default value, but some hosts have it enabled
 - ✓ `error_reporting` set to `E_ALL & ~E_NOTICE`. In Drupal 7 it will be `E_ALL`.
 - ✓ `safe_mode: off`. Safe mode may interfere with file and image uploads.
 - ✓ Php Data Objects (PDO) must be activated for Drupal 7 to install and run correctly. Look in your `php.ini`. Uncomment (remove the leading semicolon) at line `extension=php_pdo.dll, extension=php_pdo_mysql.dll`. If these lines are not there, add them.
 - ✓ It is also recommended to have setting: `session.cache_limiter: nocache`
- ✓ Some of these settings are contained in the default `.htaccess` file that installs with Drupal, no need to set them explicitly. Note, however, that setting PHP configuration options from `.htaccess` only works under the following conditions:
 - ✓ With Apache (or a compatible web server)
 - ✓ If the `.htaccess` file is actually read, i.e. `AllowOverride` is not `None`
 - ✓ If PHP is installed as an Apache module
 - ✓ Drupal 7 may require the `time` parameter to be at least 30 seconds.
 - ✓ In some shared hosting environments, access to these settings is restricted. If you cannot make these changes yourself, please ask your hosting provider to adjust them for you.
 - ✓ If using `xdebug`, ensure that the `xdebug.show_exception_trace` value is set to 0, or this could cause Drupal's installer to crash.

Browser Requirement

Drupal is fully functional and all its functionalities are supported on any browser that support CSS and Java Script. To name some of the supporting browsers:

- ✓ Internet Explorer 6.x and later
- ✓ Firefox 2.x and later
- ✓ Opera 7 and later
- ✓ Safari 1.x and later
- ✓ Google Chrome

Self Assessment Exercise

Q.1 Drupal is supported on browsers?

- a. Fire Fox
- b. Internet Explorer
- c. Chrome
- d. Any Browser which supports Java Script and CSS

Q.2 To Install Drupal Database User should have following permissions?

- a. Read only
- b. Write Only
- c. Read Write
- d. Admin

Q.3 Which is the latest PHP version supported with Drupal?

- a. 5.0
- b. 6.1
- c. 7.0
- d. 5.3

Q.4 Drupal is supported with which Webservers?

- a. IIS
- b. Apache
- c. JBOSS
- d. Both a and b

Q.5 Space required for Drupal to install?

- a. 1 MB
- b. 40MB
- c. 3MB
- d. 5MB

Q.6 On Apache Server install what rights are required on .htaccess file?

- a. Read Only
- b. Read Write
- c. Write
- d. Execute

Answers - 1-d, 2-c, 3-d, 5-c, 6-b

1.15 Basic installation Auto default features

Quick install for beginners

Basic installation instructions for beginners installing Drupal 7 from Drupal.org on an account at a localhost or webhost where you do not have ssh (command line) access.

Before you start, check that the software versions on your server meet the system requirements.

Download and extract files

To download Drupal and copy it to your webserver:

- ✓ Using your web browser, navigate to <http://drupal.org/project/drupal>.
- ✓ Click the "download" link beside the version of Drupal you want to download. This will save the file to your computer. The file is a compressed archive which contains many files and folders. Make note of the name of the file and where it is downloaded to on your computer.
- ✓ Log in to your hosted account's control panel (your hosting company will have provided this address to you).
- ✓ Create a target directory where your site will be hosted.
- ✓ Upload the Drupal files:
- ✓ Navigate to your HTML directory
- ✓ Assuming that your ISP will allow you to upload an archive file and uncompress that file, the process of uncompressing the file will generally create a new directory which contains all of the files and folders in the archive. After it has been unarchived, you can rename this folder (eg. "drupal7" or "gallery_blog").
- ✓ Please note: Attempting to upload the contents of the archive one by one is best avoided as the arrangement of these files must be maintained if Drupal is to function properly.
- ✓ Be sure to consider how you will organize your folders if you are hosting multiple sites. You will want to create the folder for your new Drupal 7 site at the same level as other sites.
- ✓ Upload your compressed .tar.gz or .zip file
- ✓ Uncompress or Unarchive the file on the server. (This process will vary from one ISP to another.)

Note: - Avoid attempting to reconstruct this set of files or changing the location of specific files will likely break your new site.

Note: - As an alternative, to extract the files locally (on your computer) you can generally double click on the file. If your computer does not already know how to open this file, you may need to locate a tool to extract compressed files—you can do an Internet search using the name of your Operating System (eg. Mac OSX, Ubuntu, Red Hat, or Windows) and 'extract tar.gz'.

- ✓ Rename the folder which contains the uncompressed files. Locate the file and extract its contents. The result should be a new directory which has the same name as the archive file that you downloaded.

Certified Open Source CMS (Drupal) Professional

Set up a Domain and point it to the directory containing Drupal's files

To set up and configure a domain name which points to this new directory, you will need to refer to your Internet Service Provider (ISP). This process can be accomplished in a variety of ways and the process is specific to the tools provided by your ISP. In general you want to:

- ✓ Register a Domain Name eg. `http://somesite.com`.
- ✓ Configure your hosting account to use this domain name.
- ✓ Point the domain name to your new folder/directory, which contains the uncompressed Drupal files

Note: These steps can take some time. Many ISPs provide an 'Add Domain' function which allows you to simply specify the domain name to be used and the directory which it points to. It can take minutes or hours for this setup to go into effect.

If you are registering a sub-domain, the portion of the name which is before the domain name, perhaps in place of the 'www' (eg. `http://myown.somesite.com`), these tend to process almost immediately, but again it depends on your ISP.

Create the configuration file and grant permissions

In order to set up new site

- ✓ Modify the `settings.php` file via your browser.
- ✓ By default at the time of un-archiving these files, only `default.settings.php` file exists with permissions of Read Only.
- ✓ Copy the file, rename it to `settings.php`
- ✓ Temporarily change the permissions so that the server can Read and Write to this file when a user makes changes via a web browser.
- ✓ Move up one level to the `/sites` directory and change the permission on the `/default` directory (folder), as this is the place where files created or uploaded via the web interface will be stored.
 - ✓ Copy the file `default.settings.php` and rename it to `settings.php`
 - ✓ Use ISP's file manager to navigate to the `/sites/default` directory
 - ✓ Select the `default.settings.php` file (usually done by checking a box next to the file)
 - ✓ Copy the file, naming the new copy 'settings.php'

Change the permissions of the settings.php file.

- ✓ ISP's file manager select the file 'settings.php' and click on permissions.
- ✓ Add the permissions for the file to be written by the web server (i.e. web users). If system is asking you to use a numeric value to CHMOD the file you will want to use 666, this will set the file to be written by anyone.
- ✓ Save changes.

- ✓ The installation script should change the permissions of the 'settings.php' file back to read-only. At the end of initial configuration of site. It is recommended to change the permissions of settings.php to Read Only.
- ✓ Change the permissions of the /default directory.
 - ✓ ISP's file manager select the /default folder and click on permissions.
 - ✓ Add the permissions for the folder to be written by the web server (i.e web users).
 - ✓ Save change.

Create the Drupal database

- ✓ Create a new, empty database for Drupal to use.
- ✓ Add a user who has full access to this newly created database.
- ✓ Methodology to create the DB will likely depend on ISP.

Most common tool used to administer databases is "phpMyAdmin". It is possible that you will have privileges to create new databases using this tool. However, it is more likely that your ISP provides access to phpMyAdmin so that you can work with content in your database. When you create a new site, the related database must be created using the ISP's control panel.

Option 1: Create a new Database using your ISP's Control Panel

For this option to run successfully, ISP will provide a database section in account's control panel. Here you will be able to create the database, adding a username and password to access the database. The approach and steps here will vary depending on ISP, in general please specify:

- ✓ Database Name (nick name of site)
- ✓ username and
- ✓ password

Combination of username, password and DB Name will be used by the website and also for the website and DB administration.

Please note: In case databases are hosted on a separate server than the webserver. In this case, please specify the alias or IP of the server where the DB is hosted. This information can be obtained by DBA.

Option 2: Create a new Database using phpMyAdmin

- ✓ Open phpMyAdmin from Control Panel.
- ✓ In Create new database field, type database name to use for new Drupal database and then click Create.
- ✓ Click the Privileges tab.
- ✓ Click Add a new User.
- ✓ In User name field, type the username want to add (this will be the username for site to access the database, not your own username).

Certified Open Source CMS (Drupal) Professional

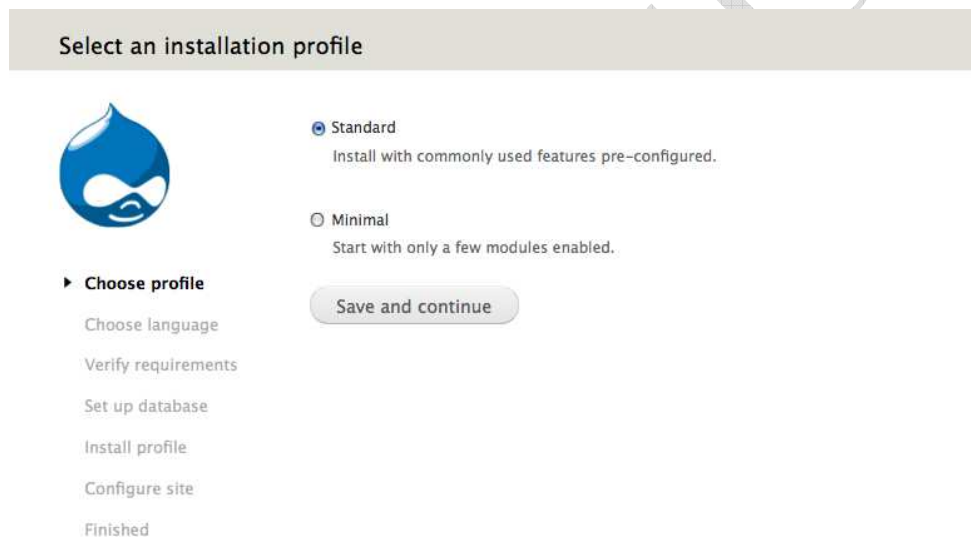
- ✓ In drop-down menu beside the Host field, select Local.
- ✓ In Password and Re-type fields, type password to use for the new user.
- ✓ In “Database for user” section, select “Grant all privileges” on the database you just created.
- ✓ In “Global privileges” section, leave all of the global privileges checkboxes unselected.
- ✓ Click Go.

Work on ISP control panel is done now. Now use a web browser to configure your new site.

Run the installation script

To run the Drupal installation script

Navigate to the base URL of website using web browser, e.g. <http://somesite.com>. You should see the Drupal installation page. The installation wizard will guide through the process of setting up Drupal website.

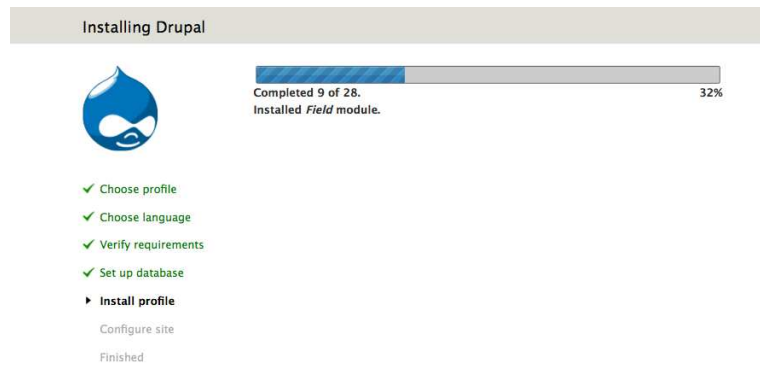


On completing the above step, if you receive an error “Parse error: syntax error, unexpected '{' in .../includes/bootstrap.inc on line 679” it may be because website is being installed on a server that is running an old version of PHP. Drupal 7 requires at least PHP 5.2.4.

- ✓ On the Select an installation profile page, select Standard. Click Save and continue.
- ✓ On the Choose language page, select English. Click Save and continue.
- ✓ On the Database configuration page, select the type of database that you are using. Type the database name, database username and database password (the same used to set up database). Please note: If your database host is located on a different server, if database server is listening to a non-standard port, or if more than one application will be sharing this database then configure Drupal options for this by clicking the advanced options link.

Certified Open Source CMS (Drupal) Professional

- ✓ Click Save and continue. The Installing Drupal page is displayed.



On the Configure site page:

In the Site name field:

- ✓ type the name you want to give your site.
- ✓ In the Site email address field, type the email address from which the automated emails will be sent.

In the site maintenance account section:

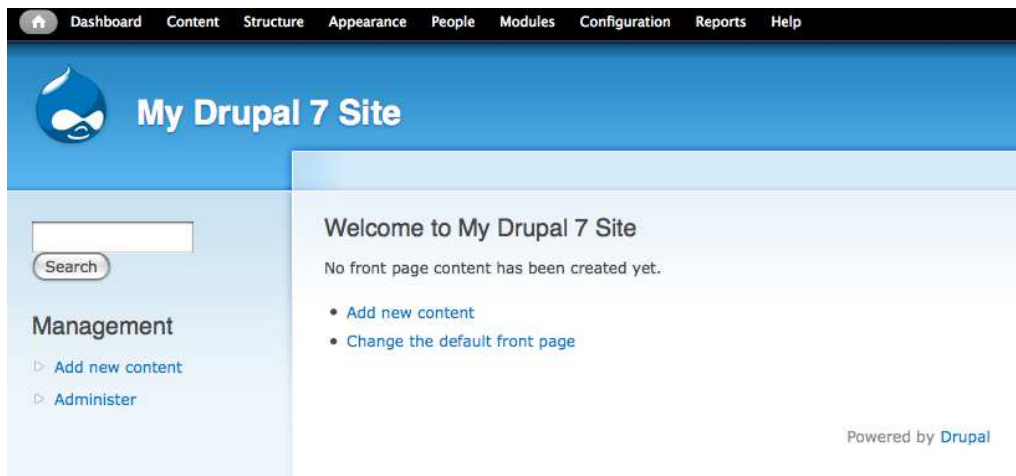
- ✓ type a username,
- ✓ email address and
- ✓ password to use for the maintenance account.

In the server settings section:

- ✓ select a country from the list
- ✓ select a time zone.

If required,

- ✓ select Check for updates automatically and Receive email notifications (recommended to keep your site's security up to date).
- ✓ Click Save and continue.
- ✓ On the Drupal installation complete page, click the Visit your new site link.



New installed Drupal homepage, as shown in the screenshot above will appear. The administrator account is automatically logged in, and if you chose the "Standard" install, the black administration toolbar will be displayed across the top of the page.

Read more about running the installation script at drupal.org.

Installation is complete

Post installation steps will be discussed later in the Post Installation section.

Command line Installation for developers

This page provides a summary of the command line instructions for installing Drupal on a typical UNIX/Linux web server. For any kind of details or trouble shooting, visit [drupal community](http://drupal.org).

Download and extract files

Download Drupal from <http://drupal.org/project/drupal> and extract the download file to a folder on your hard drive. (Replace the 'xx' in drupal-x.x.tar.gz with the actual version you've downloaded.)

```
wget http://drupal.org/files/projects/drupal-x.x.tar.gz
tar -zxvf drupal-x.x.tar.gz
```

Move the files to a directory within your web server's document root or your public HTML directory using the following command, substituting drupal-x.x with the actual version number.

```
mv drupal-x.x /var/www/
```

You can install and use Drupal in other languages by downloading translations from <http://drupal.org/project/translations>. Extract the files into the Drupal directory (e.g. /var/www/drupal-x.x)

Create the configuration file and grant permissions

In the sites/default directory, copy the default.settings.php file and rename the copied file to settings.php.

Certified Open Source CMS (Drupal) Professional

```
cp sites/default/default.settings.php sites/default/settings.php
```

Give the web server write privileges (666 or u=rw,g=rw,o=rw) to the configuration file.

```
chmod a+w sites/default/settings.php
```

Give the web server write privileges to the sites/default directory.

```
chmod a+w sites/default
```

Create the Drupal database

To complete the installation, you must create an empty database and a user with full access the newly created database for Drupal to use. You can do this by command line, or through phpMyAdmin or another database client.

```
mysqladmin -u username -p create databasename
```

Where 'username' is a MySQL user which has the CREATE and GRANT privileges. MySQL will prompt for the 'username' database password. Next you must set the access database rights. Log in to MySQL:

```
mysql -u username -p
```

At the MySQL prompt, enter:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE  
EMPORARY TABLES, LOCK TABLES  
ON databasename.*  
TO 'username'@'localhost' IDENTIFIED BY 'password';
```

Where

'databasename' is databasename

'username' is the username of your MySQL account

'localhost' is the server name used to access MySQL

'password' is the password required for that username

If successful, MySQL will reply with “Query OK, 0 rows affected”.

Run the installation script

You are now ready to run the installation script. Point your browser to the base URL of your website (e.g. <http://www.somesite.com>, <http://www.somesite.com/drupal> or <http://localhost/drupal>).

The installation wizard will guide you through several screens to set up the database, add the first user account, and provide basic website settings. Follow the wizard to finalize the installation as explained in previous section.

Installing the profile

- ✓ Start by downloading the installation and profile and unpacking it.
- ✓ Copy the entire directory that you just unpacked into the profiles directory of your core installation. For example, if you have core installed at /home/www/htdocs and you downloaded

Certified Open Source CMS (Drupal) Professional

and unpacked an installation profile called "fooprofile", copy the entire "fooprofile" directory into /home/www/htdocs/profiles/fooprofile.

- ✓ Read the documentation on the installation pack you downloaded. This can either be found on the profile's project page or in the README.TXT that came with the download. It should contain installation instructions such as listing the required extra modules and/or themes that you will have to download.
- ✓ Download these modules and themes and put them in the right directory. This is usually sites/all/modules and sites/all/themes.
- ✓ Go to your site's install.php and select which installation profile to install.

www.vskills.in

Self Assessment Exercise

Q.1 Which file is extracted and should be changed after the TAR file extraction at first?

- a. Settings.php
- b. Change default.settings.php to settings.php
- c. None of the above
- d. Both b and a

Q.2 Permission on Settings.php should be ?

- a. 666
- b. 444
- c. 555
- d. 222

Q.3 chmod command performs what actions?

- a. Change permissions on the file
- b. Change executing behavior
- c. Change creation date of file
- d. Change ownership of the file

Q.4 DB creation for Drupal requires ?

- a. Database name
- b. User name
- c. Password
- d. Full privileges to user on DB
- e. All of the above

Q.5 Which permissions on the DB are required by the user?

- a. SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER
- b. SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES
- c. CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES
- d. SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES

1.16 Post Installation

Steps those are necessary to perform before getting started on drupal and after installation.

- ✓ **Setup cron job** to perform following
 - ✓ Search module's indexing of your website's content
 - ✓ Aggregator module's retrieval of feeds
 - ✓ Ping module's notification of other sites of updates
 - ✓ System module's routine maintenance tasks, archiving logs, all depend a properly configured cron job.

Cron is a program that executes commands at specified intervals. These commands are called "cron jobs." Cron is available on Unix, Linux and Mac servers. Windows servers use a Scheduled Task to execute commands.

Frequency of the cron jobs will depend on the usage of the site, on a normal site it can be once a day where as for the more active sites it will be more often, this required to be scheduled properly in order to ensure smooth running of the website.

The cron command

In the following example, the crontab command shown below will activate the cron tasks automatically on the hour:

```
0 * * * * wget -O - -q -t 1 http://www.somesite.com/cron.php
```

In the above sample, the 0 * * * * represents when the task should happen. The first figure represents minutes - in this case, on the "zero" minute, or top of the hour. (If the number were, say, 10, then the action would take place at 10 minutes past the hour.) The other figures represent, respectively, hour, day, month and day of the week. A * is a wildcard, meaning "every time." It is recommended to go through Man pages of the OS the cron job is being created. The rest of the line basically tells the server to "ping" the url <http://www.somesite.com/cron.php>.

NOTE: The -O is "dash capital letter O", not "dash zero". Yes, the capital O is important. Lowercase o won't work

Here is a diagram of the general crontab syntax, for illustration:

```
# +----- minute (0 - 59)
# | +----- hour (0 - 23)
# | | +----- day of month (1 - 31)
# | | | +----- month (1 - 12)
# | | | | +---- day of week (0 - 7) (Sunday=0 or 7)
# | | | | |
* * * * * command to be executed
```

Thus, the cron command example above means "ping <http://www.somesite.com/cron.php> at the zero minute on every hour of every day of every month of every day of the week."

Drupal and Cron jobs

Every Drupal install requires regular actions to handle maintenance tasks such as cleaning up log files and checking for updates. Cron.php is the file that Drupal uses to run the maintenance process.

For instance, if your site were `www.somesite.com`, loading the URL `http://www.somesite.com/cron.php` in your browser would run the maintenance. This page is automatically set up when you install Drupal. Simply loading the URL will run the maintenance. Nothing more is required.

How to set up a cron job

Cron jobs are scheduled by editing the crontab. A crontab is an OS text file that contains the commands to be run with a frequency, as explained above. This file can be created and edited either through the command line interface, or, if you manage your website through a web-based control panel such as cpanel or Plesk, you will use the web interface.

In order to schedule your cron job in Plesk for Linux, go to Settings > Scheduled Tasks in the General Group. Select the system user account on whose behalf the task will be executed. Click "Schedule New Task". Specify when to run your command. You can schedule the time using the UNIX crontab entry form.

Otherwise, check with your hosting company for detailed instructions if you are using a web-based control panel.

To edit a crontab through the command line, type:

```
crontab -e
Add ONE of the following lines:
45 * * * * /usr/bin/lynx -source http://www.somesite.com/cron.php
45 * * * * /usr/bin/wget -O - -q -t 1 http://www.somesite.com/cron.php
45 * * * * curl --silent --compressed http://www.somesite.com/cron.php
```

This would have a lynx, wget, or curl visit your cron page 45 minutes after every hour.

Three options are provided in case either wget, lynx or curl are not installed on the server.

There are many ways to configure a cron job. If you have full access to crontab features, you should be able to simply paste in one of the above example commands - be sure to replace "somesite.com" with your own web domain.

If you're on shared hosting, you should be able to find cron job configuration somewhere in your hosting control panel. Some hosts even have cron "wizards" that walk you through the cron configuration, making it much easier if cron is new to you.

On a Windows system you can accomplish the same thing with scheduled tasks to launch Internet Explorer pointed to the URL.

Security notes

It is possible to run cron.php directly via `scripts/drupal.sh` with Drupal 6. Drupal.sh allows a Drupal page to be executed from a shell script. To do so, add the following cron job to run as the

Apache user

```
/full/path/to/drupal.sh --root /full/path/to/site/root/ http://default/cron.php
```

Note that `http://default/cron.php` is NOT shown as an example, it should be used as is, without changes.

Taking this approach allows `cron.php` to be blocked to prevent remote access.

To block remote access to `cron.php`, in the server's `.htaccess` file or `vhost` configuration file add this section

```
<Files "cron.php">
  Order Deny,Allow
  Deny from all
  Allow from localhost
  Allow from 127.0.0.1
  Allow from xx.xx.xx.xx <-- your IP address
</Files>
```

If you take this approach and use `drupal.sh` to call `cron.php`, it is probably best not to use the root user to run the cron job. A non-privileged user account, or the Apache account user, for example `http-service` or `www-data`, is a better choice. To do so, call `crontab -e` when logged in as a non-privileged user, or for the Apache account on a Debian server, for example, you can add a user parameter

```
sudo crontab -e -u www-data
```

The downside to this method is that any URLs generated by cron jobs using this method will not be properly formed, starting with `"http://default/"`.

Cron Jobs with Multiple sites

If you run many sites, you can use this tip to make managing your cron jobs easier. To minimize the clutter, create a `/etc/cron.5min` directory and have `crontab` read this directory every five minutes.

```
*/5 * * * * root run-parts /etc/cron.5min
```

Then place multiple individual files into the `/etc/cron.5min` directory, one for each site. The files can be named `"site1"`, `"site2"`, etc. -- note that `run-parts` may fail to detect files which contain a dot (`.`) in their name. To make sure that all of your files are visible to cron, type this at a shell prompt

```
$ sudo run-parts --test /etc/cron.5min
```

and make sure that all of your files are listed.

Each of the files in `/etc/cron.5min` should contain one line:

```
/usr/bin/lynx -source http://(full site URL)/cron.php > /dev/null 2>&1
```

or, alternatively, one of the `curl` or `wget` commands specified above.

If this doesn't work, try putting another line at the start of each file

```
#!/bin/sh
```

and make sure that the files are executable by doing

```
$ sudo chmod u+x /etc/cron.5min/*
```

- ✓ **Setup to get the Clean URL's** - by default the URL's of the dynamic pages or static pages will be something like “ <http://www.somesite.com/?q=node/100>” this is not a search engine or user friendly URL. Search engines will not index such kind of pages. To Setup the clean URL, for the most common server setup, which is an Apache web server running on some flavor of Unix/Linux, with the mod_rewrite Apache module configured and mod_rewrite enabled in httpd.conf configuration file. If this is a shared server then there might not be a permission to modify the httpd.conf file, in which case this has to be programmatically handled.

Enabling Clean URL in Drupal

Clean URLs configuration page (Administer > Configuration > Search and metadata), show the results, and allow you to save configuration.

You can enable or disable it at a later time by following these steps

- ✓ Navigate to the Clean URLs configuration page (Administer > Configuration > Search and metadata)
- ✓ Wait for the automated Clean URLs test to run.
- ✓ Check or uncheck the Enable clean URLs checkbox
- ✓ Click "Save configuration"

Even if Clean URLs are successfully enabled at install-time, if you have a dedicated server you may still want to follow the steps (below) to enable the more efficient httpd.conf rewrite method for clean URLs. If you choose to do that, you might want to turn off Clean URLs while you are working on the server.

Clean URL on dedicated Server using httpd.conf

- ✓ Enable mod_rewrite for Apache. You can talk to your web host or consult the Apache documentation for mod_rewrite to get more information on how to do this. At a minimum, this will involve making sure that mod_rewrite is enabled for your installation of Apache.
- ✓ To test if mod_rewrite is available in Apache2, you can type the following at a command prompt, to list all installed Apache modules `apache2ctl -M`
- ✓ On some systems this command may be `apachectl -M`
- ✓ In the output, check to see if the `rewrite_module` is included in the list of modules.
- ✓ If the rewrite module is not in the list, it will have to be either compiled-in or made available as a loadable module. Generally speaking, you can tell Apache to load the module by including
- ✓ `LoadModule rewrite_module modules/mod_rewrite.so`

Certified Open Source CMS (Drupal) Professional

- ✓ AddModule mod_rewrite.c in your Apache configuration file. Be sure to uncomment AddModule mod_rewrite.c, if it is in your configuration file but has been commented out. The following may work to enable the module without editing any files
- ✓ a2enmod rewrite

Note that these approaches may not work for all combinations of operating system and Apache server – consult the Apache documentation that came with your Apache software for the correct syntax.

Remember to restart Apache for the new configuration to take effect.

The next step is to locate the appropriate Apache configuration file for your site. Depending on your server configuration, the appropriate Apache configuration file could be httpd.conf, a virtual-host-specific file (vhost.conf), a specific site file (e.g. "default"), or apache2.conf. They are usually located in /etc/httpd/conf, /etc/apache2, or a sub-directory; if not, try the command:

```
find /etc -name httpd*
```

to find the file if it is located elsewhere in your file system.

If you do not have write permissions to these files, and Clean URLs are not working out-of-the-box for you, you may have to ask your systems administrator or hosting provider for help. You may still be able to read these configuration files to troubleshoot a little however.

The next step is to copy or include the Drupal-specific settings directly into your configuration file. There are instructions here for how to include the Drupal directives in your configuration file. Consult the .htaccess file in Drupal page for examples of rules, such as the following

```
<Directory /var/www/somesite.com>
  RewriteEngine on
  RewriteBase /
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
</Directory>
```

Make sure that you are looking at the .htaccess file for your major version of Drupal (i.e., 7.x, 6.x, 5.x).

Note: If you do not want to put the rewrite rules in your Apache configuration file, you can still simply use the Drupal .htaccess file (as you would if you were on shared hosting). You will need to have the AllowOverride directive set in your Apache configuration file (this will allow local .htaccess overrides on your site)

- ✓ AllowOverride All
- ✓ AccessFileName .htaccess

Read Behind the scenes with Apache's .htaccess for a thorough review of .htaccess. You may also find it helpful to view samples of Apache 2 AllowOverride directives.

Certified Open Source CMS (Drupal) Professional

Server configuration for Clean URLs on a shared server, with .htaccess

The standard Drupal installation contains a sample .htaccess file which should be sufficient to get Clean URLs running. Do not miss to copy this file, as it will be hidden in the normal view because this file is having dot in front of the file name (use ls-`alrtg` to get the listing). Before enabling Clean URLs, make sure installed location contains this file.

Still if the Clean URL's don't work, contact the hosting provider.

Make sure that .htaccess file usage

Specify explicitly in Apache configuration to use .htaccess file. This is off by default, though most hosts will have turned it on. To specify this use AllowOverride All directive .htaccess start working.

To check .htaccess read, temporarily add some garbage string to the file. Site will immediately start returning a "500 Server Error" when you load a page from that directory due to this wrong configuration. (Remove the garbage string immediately after this litmus test.)

If you do this, and your site does not break - then .htaccess is being ignored and you will not be able to use clean URLs until you get support from your hosts, or enable it in httpd.conf as described above. Some hosts allow you to enable this option through their site management control panel, so look there first.

RewriteBase setting

The main configuration option which may need to be changed for your site is the RewriteBase. This can be specified in the Drupal .htaccess file or in the httpd.conf file, depending on where you are putting the Drupal rewrite directives. By default, the RewriteBase setting is commented out of the Drupal .htaccess file, and that works well for many configurations.

If you are having trouble getting Clean URLs to work, you may need to change this setting. For example, if your Apache DocumentRoot is /var/www/ (i.e., /var/www/index.html is what is displayed when you point your browser at http://www.somesite.com/) and your Drupal installation is installed in the subdirectory /var/www/mysite/, then the RewriteBase could be set to

```
RewriteBase /mysite
```

In some configurations, setting

```
RewriteBase /
```

 will allow clean URLs to work.

```
$base_url
```

You may need to manually set the \$base_url variable in the settings.php file if not already set. It's currently known that servers running FastCGI can run into problems if the base_url variable is left commented out.

Multi-site

RewriteBase works when your Drupal installation serves only one site, or when all the sites it serves are in the same subdirectory of their domains. For example,

```
RewriteBase /
```

will work for the following sites

```
www.vskills.in
```

```
http://www.somesite.com/  
http://www.somesite2.com/  
http://www.somesite3.com/
```

```
RewriteBase /mysite
```

will work for the following sites:

```
http://www.somesite.com/mysite  
http://www.somesite2.com/mysite  
http://www.somesite3.com/mysite
```

However, if your sites are in different subdirectories, RewriteBase will not work. You will need to create a special rule for each subdirectory. For example, your Drupal installation may serve the following sites

```
http://www.somesite.com/  
http://www.somesite.com/mysite
```

In order to enable clean URLs for both sites, you will need to add

```
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteCond %{REQUEST_URI} ^/mysite/(.*)$  
RewriteRule ^(.*)$ /mysite/index.php?q=$1 [L,QSA] before the existing rewrite rules.
```

Location of index.php

For some server configurations, another change to the Drupal .htaccess file may be necessary. Find a line that looks like this, near the end of your Drupal .htaccess file:

```
RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
```

You may need to replace index.php with the URL path to your Drupal installation's index.php file. For instance, if your site's home page URL is http://somesite.com/subdir/, you might need to use /subdir/index.php instead of index.php. If your site's home page URL is http://somesite.com/, you might need to use /index.php instead of index.php. This is necessary on some, but not all server configurations.

Files Directory

In most cases, Drupal would create the files directory for you. If Drupal can't create the directory, follow the instructions below.

After installing Drupal, you will need to ensure the directory files can be accessed by your Web server. If it does not have the correct permissions you may get an error message stating that "sites/default/files does not exist ..."

Here's how

✓ In the folder sites/default create a new directory called files.

Grant read, write and execute permissions on the new directory to the Web server.

Most FTP programs will allow you to create the new directory and set its permissions.

Certified Open Source CMS (Drupal) Professional

Once you've completed these steps confirm the appropriate permissions have been assigned by navigating to Administer -> Reports -> Status report.

Scroll to the section "File system" and confirm your settings are now correct.

Check Apache-based Web Servers Settings

If the directory files is not "owned" by the Web server there are two ways to proceed. These instructions assume you have shell access to the Web server. If you do not have shell access, please contact your hosting provider for additional support.

Ideally you will be able to change the "owner" of the directory files to match the name of the Web server. On a Unix or Linux-based server you may verify "who" the Web server is running as by issuing the following commands:

```
$ ps aux | grep apache # for Apache 2.x
$ ps aux | grep httpd # for Apache 1.x
```

Depending on your Web server one of these commands will return a series of lines like this:

```
www-data 13612 0.1 0.9 50640 20340 ? S 12:29 0:08 /usr/sbin/apache2 -k start
```

The first column in the returned text is the "user" that your Web server is operating as. In this case the user is www-data. To make your files directory writeable by the Web server you need to reassign its ownership using the following command:

- ✓ \$ chown -R www-data sites/default/files
- ✓ \$ chmod -R 0700 sites/default/files

If you do not have sufficient permissions on your server to change the ownership of the folder, the next best option is to provide group-write access to the folder with the following command:

- ✓ \$ chmod -R 0770 sites/default/files

You should also be able to adjust the directory's permissions using an FTP program as outlined above.

Once you've completed these steps confirm the appropriate permissions have been assigned by navigating to Administer -> Reports -> Status report. Scroll to the section "File system" and confirm your settings are now correct.

Running multiple sites from same code base (normally known as portlets)

Running more than one Drupal site, you can simplify management and upgrading of your sites by using the multi-site feature. Multi-site allows you to share a single Drupal installation (including core code, contributed modules, and themes) among several sites.

This is particularly useful for managing the code since each upgrade only needs to be done once. Each site will have its own database and its own configuration settings, so each site will have its own content, settings, enabled modules, and enabled theme. However, the sites are sharing a code base and web document root, so there may be security concerns.

How to do it?

To create a new site using a shared Drupal code base you must complete the following steps:

- ✓ Create a new database for the site.
- ✓ Create a new subdirectory of the 'sites' directory with the name of new site, this explained below.
- ✓ Copy the file sites/default/default.settings.php into the subdirectory created in the previous step. Rename the new file to settings.php.
- ✓ Adjust the permissions of the new site directory, and grant write permissions on the configuration file as explained above.
- ✓ Make symbolic links if you are using a subdirectory such as somesite.com/subdir and not a subdomain such as sub.somesite.com, this is explained below.
- ✓ In a Web browser, navigate to the URL of the new site and continue with the standard Drupal installation procedure.

It may also be necessary to modify your Web server's configuration file, httpd.conf so that Drupal will override Apache's settings. This is true for all installations of Drupal and is not specific to the multi-site install.

Process details for Domains, URLs, and sites subdirectory name

The new subdirectory of the sites directory has a name that is constructed from the site's URL. For example, the configuration for www.somesite.com would be in sites/somesite.com/settings.php. Do not include 'www' as part of the directory name.

Drupal will use the same sites/somesite.com directory for any subdomain of somesite.com, including www, unless there is an alternative, matching subdomain sites subdirectory. For instance, URL http://sub.somesite.com would be served from sites/sub.somesite.com, if it exists.

For a subdirectory URL, such as http://somesite.com/subdir, name the sites subdirectory as follows: sites/somesite.com.subdir – explained below on how to get subdirectory multi-site working.

If installing on a non-standard port, the port number is treated as the first part of the subdomain. For example, http://www.somesite.com:8080 could be loaded from sites/8080.somesite.com. If that directory doesn't exist, Drupal would then look for sites/somesite.com, just like a real subdomain.

Site-specific modules and themes

Each site configuration can have its own site-specific modules and themes in addition to those installed in the standard 'modules' and 'themes' directories. To use site-specific modules or themes, simply create a 'modules' or 'themes' directory within the site configuration directory. For example, if sub.somesite.com has a custom theme and a custom module that should not be accessible to other sites, the setup would look like this

```
sites/sub.somesite.com/settings.php
sites/sub.somesite.com/themes/custom_theme
```



```
sites/sub.somesite.com/modules/custom_module
```

Document root

One area of frequent confusion is that in a Drupal multisite installation the webserver document root is the same for all sites. For example with the following three sites: somesite.com, sub.somesite.com and somesite.com/site3 there will be a single Drupal directory and all sites will be calling the same index.php file.

Some webhosts automatically create a new directory (i.e. somesite.com) when creating a new domain or subdomain. In this case it is necessary to make it into a symbolic link to the main Drupal directory, or better yet when creating the domain or subdomain, set it to use the same document root as the site where you have Drupal installed.

Subdirectory multi-site

If you are attempting to get Drupal multi-site working using subdirectory URLs rather than subdomain or different domain URLs, you may encounter problems. You'll start out by making a directory such as sites/somesite.com.subdir, and putting a settings.php file there. If this works for you, great! But it probably will not, until you make a symbolic link that tells your web server that the document root for http://somesite.com/subdir is the same as the document root for http://somesite.com. To do this, go to the somesite.com document root and type:

```
ln -s . subdir
```

(substituting the actual name of the subdirectory you want to make work).

If your codebase itself is in a subdirectory, then link your new site to the directory:

```
ln -s drupaldir subdir
```

This symbolic link is enough to resolve issues with subdirectory multi-site installations on at least some web hosts. But if you still encounter problems, such as issues getting clean URLs to work (especially if you had to edit the .htaccess file to get clean URLs working on the main somesite.com domain site).

Note that these problems do not usually occur for multi-site installations using a subdomain, so you might want to try that if you cannot get subdirectory multi-site to work. (However, subdomains are not good for search engine rankings!)

Localhost alias for local workstation

On many systems it is possible to create entries in a "hosts" file to create aliases for the localhost name for a local workstation. By creating aliases for localhost it is possible to create names such as localdev1.somesite.com and localdev2.somesite.com, both for the local computer.

If on the other hand you use subdirectories in your local web root, create a symbolic link like this:

```
ln -s drupaldir subdir
```

and name your site folder localhost.subdir.

Domain name changes

Once a site is in production in a particular subdirectory under the sites directory, the subdirectory should not be renamed, even if the Web site URL changes. This is because several database tables (for example: system and files) include references to "sites/www.mydomain.com." Instead of renaming the sites directory, you can create a symlink to the new URL from the old one. Navigate to the sites directory and then use the following command:
\$ ln -s /path/to/drupal/sites/old.domainname.com new.domainname.com

Security Concerns

You might want to reconsider using Drupal's multi-site configuration, in situations where the administrators for all of the sites being run from the same code base are not either the same person or a small group with a high level of mutual trust. The reason is that anyone with full administrative privileges on a Drupal site can execute arbitrary PHP code on that site through various means (even without FTP access to the site), and that arbitrary PHP code could be used from one site to affect another site, if the two sites are in the same HTTP document root and sharing the same Drupal code.

Multi-site with single codebase, different content database, shared user database, shared sign-on

- ✓ Make a copy of /sites/default to /sites/site1 and another to /sites/site2. You may add modules and themes directories in these new directories in the usual way for site-specific configurations. A files directory should be there already, if it is not there then create that. Rename /sites/site1/default.settings.php to settings.php and /sites/site2/default.settings.php to settings.php (directories should have read write permissions).
- ✓ Add these lines to your hosts files:
 - ✓ site1
 - ✓ site2
- ✓ Create three empty MySQL databases:
- ✓ drupal_site1
drupal_site2
drupal_shared_tables
- ✓ Go to http://site1 and run through the install process in the normal way, using the database name 'drupal_site1'.
- ✓ In MySQL (please download the MYSQL web client) Administrator create a new back-up project of the 'drupal_site1' database containing `_only_` the tables to share, save it and run it. Save .sql file on the hard disk.
- ✓ Check Drupal tables for sharing users are as follows

```
authmap
profile_fields
profile_values
rolesessions
users
```

Certified Open Source CMS (Drupal) Professional

- ✓ The absolute basic essential tables (if you don't want to share any data except for the users and sessions, no profile data, roles, etc.) are

```
sessions
users
```

- ✓ To Open ID authentication enabled also need to share the openid_associations table.
- ✓ (This is not a definitive list. But other features like taxonomy, etc. can be shared. Each site has its own permissions matrix - but share one across all sites.)
- ✓ Edit the .sql script created and update the database name within to 'drupal_shared_tables'.
- ✓ Using the Restore interface, run the .sql script. Now have a database called 'drupal_shared_tables' containing only the list of tables above.
- ✓ Go to http://site2 and run through the install process in the normal way, using the database name 'drupal_site2'.
- ✓ Go back to original 'drupal_site1' database and drop the same list of tables from the schema. Perform same for the 'drupal_site2' database.
- ✓ Databases are ready now.
- ✓ Browse to your Drupal application and edit /sites/site1/settings.php. Change these lines:

```
<?php
$db_url = 'mysql://user:password@localhost/drupal_site1';
$db_prefix = ";
?>
To:
<?php
$db_url = 'mysql://user:password@localhost/drupal_site1';
$db_prefix = array(
  'default' => ",
  'authmap' => 'drupal_shared_tables.',
  'profile_fields' => 'drupal_shared_tables.',
  'profile_values' => 'drupal_shared_tables.',
  'role' => 'drupal_shared_tables.',
  'sessions' => 'drupal_shared_tables.',
  'users' => 'drupal_shared_tables.',
);
?>
```

Note the all important trailing dot on the end of drupal_shared_tables - This is very much required! This is pre-pending table name info to Drupal's default table name, so the dot is required or Drupal will try to select

drupal_site_1.drupal_shared_tablesauthmap

Instead of the desired

drupal_shared_tables.authmap

- ✓ Do the same for `/sites/site2/settings.php`, but with `$db_url` set to `drupal_site2`.
- ✓ Now login as the super user on both sites separately (`http://site1/user` or `http://site2/user`)
- ✓ Using the credentials you entered when you installed site1, give them both different settings, modules, permissions and content but use the same user credentials. Moreover, if you create a user on site1, you will see them in users in site2, but with an entirely independent set of permissions you can decide on a site level.
- ✓ Now for the shared logins. Download and unzip the Single Sign On module in to `/sites/all/modules`.
- ✓ Enable the Single Sign On Controller module on the site you want to use as the master for logins (the controller) - in this case, site1. Enable the Single Sign On Client module on all other sites, for this example just site2.
- ✓ Then IMMEDIATELY add the following line to `settings.php` for both sites or you will get PHP errors:

```
<?php
# Single Sign On settings:
$config['session_inc'] = 'sites/all/modules/sso/session.singlesignon.inc';
?>
```

- ✓ Copy and paste the Controller information on site1 (found at `admin/settings/singlesignon-controller`) over to the Client settings on site2 (go to `admin/settings/singlesignon-client`). Save settings on site2 and logout of both sites.
- ✓ Go to `http://site1/user` and login. Now go to `http://site2`. You should stay logged in. Logout again and see you are logged out in both locations. Reverse it so you login on site2 and visit site1.

1.17 Install Drupal with multi lingual support

Not all languages are available for every version of Drupal, but if the one required in the site is available, follow these steps to install Drupal in your language:

- ✓ Download the language file from the translation server.
- ✓ Create a directory named translations inside the directory of the profile you will be using. This is most likely inside `profiles/standard`.
- ✓ Move the downloaded `.po` file inside the translations folder you just created.
- ✓ Install Drupal as per usual by visiting `www.somesite.com/install.php`.

Automating installation and updates

Once you start to add modules and a theme to the site, downloading and keeping translations updated manually will be very complex and tiring. It is suggested that instead you start off with the Localized Drupal installation profile or adapt the Localization update module to keep yourself updated.

1.18 Install Modules and Themes

This directory should be used to place downloaded and custom modules and themes which are common to all sites. This will allow you to more easily update Drupal core files. These modules and themes should be placed in subdirectories called modules and themes as follows:

sites/all/modules
sites/all/themes

Installing Modules Instructions

Regularly review and install [available](#) ↓

[+ Install new module](#)

OTHER

ENABLED	NAME
---------	------

- ✓ **Use the Upgrade manager.** The Upgrade manager is new to Drupal 7 and gives administrators the ability to upload modules and theme directly to the site. Navigate to (Administer > Modules) and click 'Install new modules'.

Install from a URL

For example: <http://ftp.drupal.org/files/projects/name.tar.gz>

Or

Upload a module or theme archive to install

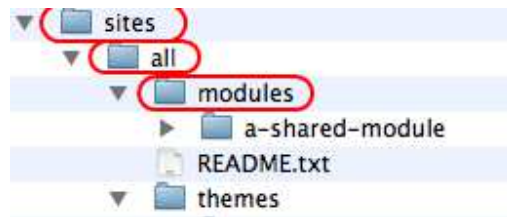
For example: [name.tar.gz](#) from your local computer

Select the URL for the distribution you want to install.

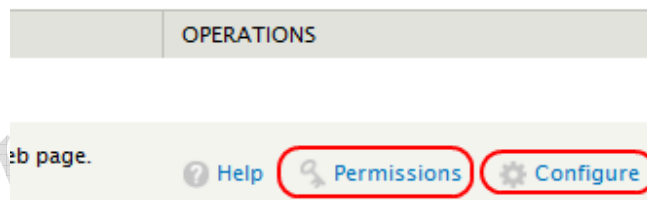
- ✓ **Download the module.** The module version must be compatible with your version of Drupal. Note that "Development snapshots" are modules that are in an active stage of development. They may be written for a previous/current/future version of Drupal, and they are considered unstable and should be handled with care.
- ✓ **Extract the files.** When you first get the module, it will appear in a compressed file format such as 'tar.gz'. On Windows, use a program like 7-zip to extract it. On the Mac, you can use Stuffit Expander. For *nix systems, use the command line

```
tar -zxvf modulename-drupalversionnumber.tar.gz
```

- ✓ You should see a list of files extracted into a folder.



- ✓ **Upload the folder.** Transfer your files with SFTP or FTP to the desired modules folder in your Drupal installation. The modules folder at the top level of your Drupal installation is reserved for Drupal core modules (the ones that come with the original download of Drupal). So, you should create a sites/all/modules directory and put uploaded modules there. If you are running a multi-site installation of Drupal, you can create a modules folder under sites/my.site.folder and put modules there that are specific to a particular site in your installation. Modules that will be shared between all sites should be placed in sites/all/modules.
- ✓ **Read the directions.** If the module has an installation file (usually INSTALL.txt and/or README.txt), read it for specific instructions. There are modules that require special treatment, and even modules that depend on other downloaded files to function properly. Sometimes the README file has no .txt extension. When you try to double-click on it, your computer doesn't know what program to use. In that case, open your favorite text editor first, and then open the file using the editor's 'file open' command.
- ✓ **Enable the module.** Navigate to (*Administer > Modules*). Check the 'Enabled' box next to the module and then click the 'Save Configuration' button at the bottom. NOTE: If you are upgrading an existing module you'll need to browse to your update page at <http://www.somesite.com/update.php> and click on 'run the database upgrade script' *after making a backup of both your database and 'sites' folder*.



- ✓ **Set up permissions.** Some modules will require you change permissions to get them working. Permissions information may be in the instructions that came with the module. Navigate to (*Administer > Modules >*) and click on the 'Permissions' button for the desired module. Users with the "Administer permissions" permission can also do this on the Permissions page (*Administer > People > Permissions*). Scroll down to see if the module appears in the list and, if it does, give the appropriate permissions to desired roles.
- ✓ **Adjust settings.** Most modules will have some type of settings page. It will vary from module to module but if not described in the README.txt file it will usually be located by navigating to (*Administer > Modules*) and clicking on the 'Configure'. If you have trouble locating a settings page for a module, navigate to the (*Dashboard > By Module tab*).
- ✓ If you run into problems, search the module's issue queue and the forums. If your problem hasn't already been addressed, post a question or issue and someone will try to help you out.

Installing Themes

- ✓ **Download the theme.** You can find themes on <http://drupal.org/project/themes>, as well as some external sites. Make sure the version of the theme matches your version of Drupal. Note that themes labeled "CVS" or "DEV" are in a development stage. They may be written for a previous/current/future version of Drupal, and they are considered unstable and should be handled with care.
- ✓ **Extract the files.** When you first get the theme, it will appear in a compressed file format such as 'tar.gz'. On Windows, use a program like 7-Zip to extract it. On the Mac, you can use Stuffit Expander. To extract the file using the Unix command line:

```
tar -zxvf themename-drupalversionnumber.tar.gz
```

You should see a list of files extracted into a folder.

- ✓ **Upload the folder.** FTP/Copy/SCP your files to the desired themes folder in your Drupal installation. Since the themes folder at the top level of Drupal is typically reserved for Drupal core themes, you should create a sites/all/themes/ directory and put uploaded themes there. If you are running a multi-site installation of Drupal, you can create a themes folder under sites/my.site.folder and put themes there that are specific to a particular site in your installation. Themes that will be shared between all sites should be placed in sites/all/themes.
- ✓ **Read the directions.** If the theme has an installation file (usually INSTALL.txt and/or README.txt), read it for specific instructions. There are themes that require special treatment to function properly.
- ✓ **Enable the theme.** Go to administer > site building > themes. Check the 'Enabled' box next to the theme.
- ✓ **Optional:** Make it the active, default theme. Check the 'default' box to make this the chosen theme for your site. Enabled alone will allow users to select the theme, if you have allowed that permission.
- ✓ Click the 'Save Configuration' button at the bottom

Modules and Themes Version

You can see the versions of your installed modules and themes on the modules and themes administration screens. In Drupal 7, go to Administration >> Modules or Administration >> Appearance, and you will see a column listing the version number. (In Drupal 6, the equivalent pages can be found at Administer >> Site building >> Modules and Administer >> Site building >> Themes.)

1.19 Migrating to Drupal from existing CMS or CFM

Migration of site from any CMS to Drupal is a big task depending on following factors:

- ✓ Current CMS used
- ✓ Current Language used
- ✓ Current OS and platforms

Certified Open Source CMS (Drupal) Professional

- ✓ Current DB used
- ✓ Size of the data
- ✓ Current user roles and permissions defined
- ✓ Any Regional or multi lingual support used.
- ✓ What character set is being used.

www.vskills.in

Self Assessment Exercise

Q.1 Cron Jobs in Drupal are required for?

- a Checking health of the system
- b Aggregating Content
- c Indexing Content
- d All of the above

Q.2 The default URL of the site is created by using

- a Default.php
- b Index.php
- c Webservice
- d All of the above

Q.3 Which file is most important in creating the URL for multisite deployment

- a Index.php
- b Default.php
- c Settings.php
- d None of the above

Q.4 Why are themes required?

- a To provide skin to the website
- b Work with templates
- c Create the New Templates
- d All of the above

Q.5 Themes and Modules are installed at directory?

- a Site/themes
- b Site/module
- c Site/modules and site/themes
- d site

Answers - 1-d, 2-b, 3-d, 4-d, 5-c