shopsite

Linux/UNIX Install Help

This document and the contents thereof are property of ShopSite, inc. and are not to be modified or redistributed without express written permission.

SHOPSITE, INC., PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event shall ShopSite be liable for any loss of profits, loss of business, loss of use of data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind, even if ShopSite has been advised of the possibility of such damages arising from this publication. ShopSite may revise this publication from time to time without notice.

Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Installing ShopSite on Linux/UNIX Servers

ShopSite is easy to install on your Linux or UNIX Web server; the only tools required are a Web browser and FTP or shell (SSH) access to your Web server.

Note:

Most ShopSite hosting partners will install ShopSite on your server for you. You should contact your ShopSite partner if you have questions about your ShopSite installation.

These instructions are for installing new ShopSite stores on Linux/UNIX servers.[1] See the Upgrade Help for instructions to upgrade an existing ShopSite installation. There are different instructions for Installing or Upgrading ShopSite on Microsoft Windows Servers.

Warning:

Whenever connecting to your server for administrative purposes (such as installing or upgrading ShopSite), you should use a secure connection. Telnet and standard FTP are both unencrypted connections, and could be intercepted. Always use a secure connection, such as SSH for shell connections, and FTP over SSH or SFTP for file transfers.

Overview

The process of installing ShopSite includes the following phases:

1. Preparing to Install
2. Obtain Installation Files
3. Upload Installation Files
4. Run Installation Program
5. Troubleshooting

In order to keep the instructions as clear as possible, examples that follow assume you are installing a single[2] ShopSite store on the main domain name of your Web server, and that your Web server files are located within the `/www` directory on your server. If you are installing your ShopSite store in a different location, you will need to use the appropriate paths for your Web server.

Preparing to Install

If you are running a properly configured, compatible Web server, ShopSite usually does not require any special Web server configuration. Before you get started, however, you should be aware of the following:

- ShopSite runs on a variety of Operating Systems and Web servers. See the System Requirements Help to make sure you have a compatible Web server.
- Your Web server must be using NCSA compatible user authentication. Most Web servers, including Apache and Microsoft IIS, should be already configured to do this.
- In order to comply with PCI Security Requirements, you **Must** have SSL (secure connections) configured and working on your Web server, and you must configure ShopSite to use SSL.

Prior to beginning the installation, you should plan out where you will install the different ShopSite directories[2]. You do not need to create any of these directories at this time, but you should decide where they will be and make sure they will have the required access. ShopSite will let you put the directories wherever you want, and you can name them however you wish, as long as the directories have the appropriate access rights for the Web server[3]. You will be prompted to specify the locations for these directories as part of the installation process.

- **CGI Directories**
  ShopSite uses numerous CGI programs, which must be allowed to run from a Web browser. Your Web server may already have a `cgi-bin` directory where you can put your ShopSite CGI directories, or you can configure your Web server to allow CGIs to run in the directories where the ShopSite CGIs will be. In order to run CGI programs, the Web server must also have `read` and `write` permissions for these directories and all their contents, and `execute` permission for the directories and any CGI programs in them[3]. ShopSite uses the following CGI directories:
  - Install Directory - This is the directory where you will upload and run the installation files. This directory is only required while you are installing or upgrading ShopSite. Once you have ShopSite running properly, it is a good idea to restrict Web access to or remove the install directory.

    Example FTP/Shell Path: `/www/cgi-bin/shopsite-install`
    Example URL[4]: `http://www.mystore.com/cgi-bin/shopsite-install`

- Shopping Cart Directory - This is where the shopping cart CGI files will be stored.

  Example FTP/Shell Path: `/www/cgi-bin/sc`
  Example URL[4]: `http://www.mystore.com/cgi-bin/sc`

- ShopSite Directory - This is where the merchant back office CGI files will be stored.

  Example FTP/Shell Path: `/www/cgi-bin/ss`
  Example URL[4]: `http://www.mystore.com/cgi-bin/ss`

- **HTML Directories**

In order to access the static HTML elements of your store, the Web server must have `read` and `write` access to these directories and their contents. The directory itself also needs to have `execute` permissions for the Web server[3]. ShopSite stores static HTML content in the following directories:

- Store Pages Directory - this is the directory where ShopSite will put the store pages it generates. It may also be referred to as your *HTML Output Directory*. Most merchants will want this to be the same directory as their domain name's root URL. The URL of this directory is included in the store auth file and can not be changed without obtaining a new auth file.

  Example FTP/Shell Path: `/www/htdocs`
  Example URL[4]: `http://www.mystore.com`

- Store Media Directory - this is the directory which contains the images your store uses.

It is a non-configurable sub-directory of the Store Pages directory. While you can not configure this directory, you should consider it when planning the location of your Store Pages directory. You can upload images to this directory in the back office, but you may also want to have FTP access to this directory if you will be uploading a large number of images.

Example FTP/Shell Path: `/www/htdocs/media`
Example URL[4]: `http://www.mystore.com/media`

- ShopSite Images Directory - this is the location where ShopSite stores images used by

the back office and by some of the built-in themes.

Example FTP/Shell Path: `/www/htdocs/shopsite-images`
Example URL[4]: `http://www.mystore.com/shopsite-images`

- **Data Directory**

The store data directory is where the databases and store configuration files are located. This directory must have `read` and `write` permissions for the Web Server[3], but ***it should not be accessible from a Web browser***. This means it must not be inside any directory that can be accessed from a Web browser (such as the `htdocs` or `cgi-bin` directories). This is an *essential* requirement for PCI Security Requirements.

Example FTP/Shell Path: `/www/data`

Not Accessible using a URL

Obtaining Installation Files

Once you are ready to install ShopSite, you need to obtain the installation files. Each ShopSite store requires a unique Auth file in order to run. See the Auth File Help for information about what an Auth file is and how to obtain one. Make sure you save your Auth file in a location where you will be able to find it later. **Be sure to use ASCII (text file) format** any time you use FTP to transfer your Auth file.

The ShopSite program files must be installed on the Web server.[2] ShopSite partners can obtain the program installation files directly from ShopSite by following the instructions below. You will need the partner ftp username and password you were given when you signed the partner agreement. Make sure you have them ready before you start.

FTP to the ShopSite Support FTP server and download the installation files using one of the following methods:

- **Using an FTP client on your local computer:**
  1. Open your FTP client program and open the *Connect* dialog.
  2. Enter `support.shopsite.com` as the **Hostname**.
  3. When you are prompted, enter the username and password you were provided.
  4. On the *Server*, change to the directory that correlates to your Web server operating system.
  5. On your *local computer*, change to the directory where you want to put the installation files. You may want to use the same location where you put your Auth file.
  6. *Use* ***Binary (BIN)*** *mode* to copy the installation file to your local computer. The file to download will be named `shopsite-`*`NN-OperatingSystem`*`.zip` (where *NN* is the ShopSite version and *OperatingSystem* is your Web server operating system).
  7. If there is a `patch` directory in the operating system directory, you will also need to copy any files in that directory to a sub-directory on your local computer *using* ***Binary (BIN)*** *transfer mode*.
- **Using a Web browser on your local computer:**
  1. Open your Web browser and enter the following URL into the location bar. You will need to replace *MyUserName* and *MyPassword* with the username and password you were provided:
     `ftp://`*`MyUserName`*`:`*`MyPassword`*`@support.shopsite.com/`
  2. Click on the name of the directory that correlates to your Web server operating system.
  3. Click on the file named `shopsite-`*`NN-OperatingSystem`*`.zip` (where *NN* is

the ShopSite version and *OperatingSystem* is your Web server operating system).

4. When prompted, select a location to save the file on your local computer. You may want to use the same location where you put your Auth file.
5. If there is a `patch` directory in the operating system directory, you will also need to save any files in that directory to a sub-directory on your local computer.

- **Using an FTP client on your Web server:**
  1. Use SSH to connect to your Web server, then FTP to `support.shopsite.com`.
  2. When you are prompted, enter the username and password you were provided.
  3. On the ShopSite support server, change to the directory that correlates to your Web server operating system.
  4. Change to the directory on your Web server where you want to store the installation file. It is a good idea to use a different directory than the one you will unzip the archive into, so you don't accidentally overwrite anything important.
  5. *Use* ***Binary (BIN)*** *mode* to copy the installation file to your Web server. The file to download will be named `shopsite-`*`NN-OperatingSystem`*`.zip` (where *NN* is the ShopSite version and *OperatingSystem* is your Web server operating system).
  6. If there is a `patch` directory in the operating system directory, you will also need to copy any files in that directory to a sub-directory on your Web server *using* ***Binary (BIN)*** *transfer mode*.
  7. Unless otherwise indicated, you will still need to follow the **Upload Installation Files** steps below.

Upload Installation Files

Once you have all the files required to install ShopSite, you need to create the install CGI directory on your Web server and copy the install files into that directory.

1. Create the directory from which you will run the Install CGIs. This is the Install Directory you planned out during the preparation stage. Make sure the Web server has the correct permissions and ownership for the directory[3], and that the directory is in a location where you can run CGIs from a Web browser.
2. Copy the store Auth file to the Install Directory on your Web server. **Be sure to use ASCII (text file) format** any time you use FTP to transfer your Auth file. If your Auth file has a different name, change the file name to `store.auth`. You must have your Auth file in the install directory before you can install or use ShopSite.

3. If you do not have SSH access on your server, you will need to unzip the install file before uploading it to your server.
4. Copy the install file (or all the files you extracted from the install file) to the Install Directory on your Web server. *If you are using FTP, make sure to use **Binary (BIN)** transfer mode*.
5. If you have not already unzipped the install file, unzip it into the Install Directory on your Web server.
6. Check to make sure all your files have correct permissions and ownership[3].
7. If you need to conserve disk space, you may remove the install file (the `.zip` file only) at this time.

If you have any patch files, you will apply them *after* you have completed installing ShopSite.

Run Installation Program

Once all the required files are in your Install Directory and have appropriate permissions and ownership[3], you can run the Installation Program from your Web browser. To start, enter the Install Directory URL followed by `/wwwinstall.cgi` into the location bar of your Web browser, then press **Enter**.

- Example Install URL: `http://www.mystore.com/cgi-bin/shopsite-install/wwwinstall.cgi`

When the Install Program loads in your Web browser, you should see **ShopSite Install Menu** displayed at the top of the screen. See Troubleshooting if you do not see this screen.

Select the radio button to **Create a new ShopSite installation (with first store)** and click **Continue**.

The next screen is where you will enter the settings, including directory locations and URLs, for the store. The **Install Help** button at the top of the screen will open this help page in a new browser window. You will be required to enter the following settings:

ShopSite directory	This is the FTP/Shell path of the ShopSite Directory.
URL of ShopSite CGI's	This is the URL of the ShopSite Directory.
Shopping Cart directory	This is the FTP/Shell path of the Shopping Cart Directory.
URL of Shopping Cart CGI's	This is the URL of the Shopping Cart Directory.

HTML Output directory (associated URL is defined in AUTH file)	This is the FTP/Shell path of the Store Pages Directory.
ShopSite Images directory	This is the FTP/Shell path of the ShopSite Images Directory.
URL of ShopSite Images	This is the URL of the ShopSite Images Directory.
Data directory	This is the FTP/Shell path of the Data Directory.
Location of sendmail executable	This is the FTP/Shell path for `sendmail` on your server.
Location of tar executable	This is the FTP/Shell path for the `tar` utility on your server.
Password for *StoreID*	Enter the password that will be initially used to access the ShopSite back office. The merchant can change the password by going to **Utilities** > **Change Password** after logging in.
Merchant email address	Enter the contact e-mail address for the merchant. The merchant can change this later by going to **Preferences** > **Hosting Service** after logging in.
Locale	Select the locale for ShopSite to use. The merchant can change this later by going to **Preferences** > **Locale** after logging in.
Currency	Select the primary currency for ShopSite to use. The merchant can change this later by going to **Preferences** > **Locale** after logging in.

If you have not already created the directories ShopSite will use, check the box for **Do you want Install to create any directories that do not already exist?**. If you set up your permissions properly, ShopSite should be able to create these directories.

Click **OK** to install ShopSite using the settings you specified. The Installation Progress screen will display information about the installation, with any warnings or errors displayed in red. If the installation completes successfully, you will see **The update has been completed** in large text near the end of the screen. ShopSite will rename the `store.auth` file to `store.bak` when the install completes to prevent anyone from running the Install Program again.

Once the installation is complete, you will need to install any patches that you have. Patches are usually a newer version of a CGI file included in the original installation. To install a patch file do the following for *each* patch file:

1. Using FTP or SSH, locate the existing CGI with the same name of the patch file.
2. Back up the existing CGI file by renaming it or copying it to a new location (for example, you could change `order.cgi` to `order.cgi.old`).

3. Copy the patch file to the directory where the existing CGI was. *If you are using FTP, make sure to use **Binary (BIN)** transfer mode*.
4. Make sure the patch file has the correct ownership and permissions[3] for the Web server.

Once you have confirmed that ShopSite is installed and working, you may remove the Install Directory and its contents in order to free up disk space. If you do not delete the directory, you should at least change permissions[3] for the Install Directory so it is no longer accessible from the Web, in order to prevent anyone from accessing the Installation Program without permission.

Troubleshooting

If you encountered any problems installing ShopSite, look at what errors you got, and where you got them, then check this list of troubleshooting tips to see if your problems were caused by any of these common mistakes:

1. **Are all your file permissions correct?** Incorrect Permissions[3] are the most frequent cause of problems installing ShopSite. In order to install ShopSite, you must grant your Web server the permission to view, modify, and run the files used in the installation. The File Permissions Explained page provides a more detailed explanation of file ownership and permissions.
2. **Did you use ASCII format to transfer your Auth file?** - Any time you use FTP to transfer your Auth file, you should use the ASCII (text file) format. If you transferred the file using BIN format, the end-of-line characters in the file may not match the ones used by your operating system, causing the file to appear corrupted to the Web server. **Always use ASCII format to transfer your auth file via FTP**.
3. **Does your Web server allow CGIs to run?** The ShopSite install directory needs to be in a location where you can run CGI programs from a Web browser, such as the *cgi-bin* directory. You may need to change the location of your install directory, or make configuration changes to your Web server to allow you to run CGI programs in your ShopSite CGI directories. Contact your hosting provider for more information about configuring your Web server.
4. **Did you use binary transfer mode to upload the install files?** The install zip file and the files within it are binary (encoded) files. If you upload or download a binary file using ascii (plain text) transfer mode, the files will be corrupted. Make sure you use binary transfer mode when you download the files from ShopSite and when you upload them to your Web server.
5. **Are you using the correct Store URL and ID?** When your *store.auth* file was generated, you specified a Store URL and a store ID to use for your store. The URL and ID are used to help uniquely identify your store, and are encoded in your *store.auth* file. If you want to use a different URL or store ID, you will need to obtain a new auth file. See ShopSite Auth File Information for details on obtaining and replacing an auth file.

6. **Is your *Store.auth* file in the install directory?** The ShopSite install program requires your *store.auth* file to be in the same directory as the install files. If your auth file is in a different location, or is not named *store.auth*, you will need to correct that before the install program will run. After completing the installation, your auth file is renamed *store.bak* in order to prevent the install program from being run without your authorization.

Footnotes:

1. If you are a ShopSite reseller using the scripted installation utility, see the Scripted Install help.

2. If you are installing multiple ShopSite stores on a single server, you can save space by having all the stores use the same CGI and ShopSite Images directories. Each store must have its own HTML, Media, and Data directories, as well as a unique license (auth file). For more information on installing multiple stores on a single server, see the Creating a ShopSite Mall instructions.

3. Linux and UNIX use file permissions to control what level of access a program (or the person using a program) has to a file or directory on the server. In order to install or run ShopSite, you must ensure the Web server has the correct ownership and permissions to run the install CGI and access the other install files. The ShopSite install files and the directory that contains them should belong to the user and group the Web server requires (usually, this is the user and group the Web server runs as). You will also need to grant the user and group *read* and *write* permission for the directory and files, and *execute* permission for the directory and the install CGI. See file permissions explained for more detailed information.

4. Sticklers for proper formatting of URLs will have noticed that when indicating a URL for a directory (rather than a specific file), the examples do not include a trailing slash at the end of the URL. The reason for this is because ShopSite uses the URLs from the configuration to create URLs to specific files, and will attach a slash before the path to the file. If you include a trailing slash in the URLs when you configure ShopSite, your URLs will get an extraneous slash in them such as `http://www.mystore.com/cgi-bin/sc//order.cgi`. While this normally will not cause any problems using ShopSite, it is not the proper way to format URLs either.

Upgrading ShopSite on Linux/UNIX Servers

New versions of ShopSite are released periodically to provide merchants with new features or improvements on existing features. ShopSite Starter or Manager merchants may also want to upgrade service levels to get access to additional features. Upgrading ShopSite on Linux or UNIX servers is easy; the only tools required are a Web browser and FTP or shell (SSH) access to your Web server.

Note:

Many ShopSite resellers and Web hosting providers will automatically upgrade your ShopSite store for you when a new version becomes avaialble. You should contact the reseller or hosting provider if you have questions about upgrading your ShopSite store.

These instructions are for upgrading ShopSite stores on Linux/UNIX servers[1]. See the Installing ShopSite Help for instructions to install a new ShopSite store. There are different instructions for Installing or Upgrading ShopSite on Microsoft Windows Servers.

Warning:

Whenever connecting to your server for administrative purposes (such as installing or upgrading ShopSite), you should use a secure connection. Telnet and standard FTP are both unencrypted connections, and could be intercepted. Always use a secure connection, such as SSH for shell connections, and FTP over SSH or SFTP for file transfers.

Overview

ShopSite can be upgraded to newer versions or to higher service levels. A **version upgrade** means to upgrade is an upgrade to a more recent release of ShopSite with new or improved features. Version upgrades can either be full versions (ShopSite 7.x to ShopSite 8.x) or point releases (ShopSite 8.0 and ShopSite 8.2). A **service level upgrade** is an upgrade to a greater feature set (ShopSite Manager to ShopSite Pro) within the same version.[2]

- Upgrading to a Newer Version of ShopSite
- Upgrading to a Higher Service Level

Upgrading to a Newer Version of ShopSite

New versions of ShopSite are released periodically to provide additional features or improvements on existing features. When a new version includes significant improvements over the previous version, the major version number is incremented (e. g. 9.x to v10). This is called a **full version upgrade**. Between full releases, less substantial improvements are released as a **point release upgrade**, and the minor version number is incremented (e.g. 8.0 to 8.2).

The process of upgrading to a new version of ShopSite is the same, regardless of whether it is a full or point release upgrade. To upgrade to a newer version of ShopSite, do the following:

1. **Backup your existing store.** You should always make sure any important information has been backed up before making any changes to your store. All store-specific information, including your auth file, is kept in your store's **data directory**. It is a good idea to store a backup copy of your data directory in a compressed archive on your local computer.
2. Obtain the ShopSite program installation files. You do not need a new auth file to upgrade your store, but you will need the program files for the version of ShopSite you are upgrading to. See Obtaining Installation Files in the install help for detailed instructions on obtaining the new ShopSite installation files.
3. Upload the installation files to your server. This is the same procedure outlined in Upload Installation Files in the install help. Your may need to copy the *storeid*.auth file from your data directory and save it as store.auth in your install CGI directory.[3]
4. Run wwwinstall.cgi in your Web browser. This is the same program used to install new stores, and is located in your store's Install Directory.
   - Example Install URL: `http://www.mystore.com/cgi-bin/shopsite-install/wwwinstall.cgi`
5. Select the radio button to **update**[4] your store and click **Continue**.
6. Confirm the paths ShopSite should be using for the upgrade, then select whether or not ShopSite should create a backup copy of your store's data directory before performing the upgrade. Click **START UPDATE** to continue.
7. If you selected not to have ShopSite backup your store content, you will get a warning message. If you have not already created a backup, you should check the box to have ShopSite create a backup. Click **CHECK PERMISSIONS** to continue.
8. ShopSite will check file permissions for the content that will be upgraded.[5] If any locations do not pass the check, you will need to change the permissions on that directory before continuing. Your Web server needs to have read, write, and execute permissions to all ShopSite content in order to upgrade successfully. For more information, see Linux and UNIX File Permissions Explained. After fixing any permissions problems, click **CHECK PERMISSIONS** to have ShopSite test file permissions again.
9. Once all your directories pass the permissions check, click **CONTINUE UPDATING** to complete the upgrade. ShopSite will then install the upgraded content and program files.

After the installation completes, you should log in to your ShopSite back office and confirm that the upgrade completed without errors.

10. After running the upgrade, you may have patch files to install. Follow the steps to install patches outlined in the installation help.

If you encounter any problems during the upgrade process, see the Troubleshooting section of the installation help.

Upgrading to a Higher Service Level

ShopSite is available in different service levels, with lower levels offering fewer features for a reduced cost. For example, ShopSite Pro offers several features to help you draw customers to your store that are not available in a less expensive ShopSite Manager store. See the Feature Checklist to compare featurse available in different service levels of the current version of ShopSite.

If you originally purchased a lower service level version of ShopSite but want to start using the additional features available in a higher service level, you can upgrade your current store without any hassles. All you have to do is install a new auth file to allow your store to run at the new service level:

1. **Backup your current auth file.** If something is wrong with your new auth file, you may need to restore the old one until you can get your new auth file replaced. Your auth file is located in your store's data directory, and is named `*storeid*.auth` (where *storeid* is the username you use to sign in to your ShopSite back office). You can make a copy of the file with a different name (`*storeide*.auth.old`, for example) in the same directory, or on your local PC.
2. Obtain a new auth file from your ShopSite reseller or hosting provider. The reseller who provided you with your original auth file can provide you with a new one for your new service level.
3. Save the new auth file as `*storeid*.auth` (where *storeid* is the username you use to sign in to your ShopSite back office) in your store's data directory. Make sure the file name matches the storeid of your store, or ShopSite will not run.
4. Log in to your ShopSite back office and check that the new auth file is installed correctly. The store version and service level are indicated at the bottom of the main back office screen, followed by your storeid. If you have any problems, see the troubleshooting section of the Auth File information page.

For more information about your auth file, see the ShopSite Auth File Information page.

Footnotes:

1. If you are a ShopSite reseller using the scripted installation utility, see the Upgrading instructions in the Scripted Install help.

2. In order to avoid unnecessary complications, you should not atttempt to upgrade version numbers and service levels at the same time. If you want to do both, start by upgrading versions, then upgrade your service level afterwards. This will make troubleshooting upgrade problems easier, and cost you less.

3. If you did not delete your install CGI directory, your auth file is already there with the name `store.bak`, and you can simply rename it `store.auth` to upgrade your store.

4. The option to install a new store *will* wipe out an existing store using the same auth file. If you want to keep settings and content from an existing store, *make sure* to use the option to upgrade your store.

5. If you are upgrading a mall with more than one store in it, you will get an additional screen before the file permissions check which mentions some information you should be aware of. You will need to click **CONTINUE** to proceed to the file permissions check.

ShopSite System Requirements

ShopSite shopping cart software is designed to work in a variety of common Web hosting environments. You can get ShopSite for Web servers running on Linux, FreeBSD, Solaris, and Windows Server operating systems.

In addition to the ShopSite program requirements indicated below, you will require disk space for your store content. The amount of disk space required by an individual store depends on several factors, such as the number of products and orders and the size and quantity of images you use. As a general guideline, we suggest having at least 2MB of disk space set aside for your store, but you may need more.

You will need to have administrative rights on your Web server to install ShopSite. In order for ShopSite to run without problems, your server should meet at least these minimum requirements, depending on your operating system:

- ***Linux***
  - Kernel 2.2 or newer with glibc 2.1.3
  - Apache or compatible Web server[1]
  - SSL support is required for PCI compliance
  - Sendmail or compatible SMTP server
  - 1Ghz or better Intel/x86 Compatible CPU
  - 256MB RAM
  - 40.4MB Hard Drive space for program files
  - 40.4MB Hard Drive space for installation files[2]
- ***FreeBSD***
  - Version 4.3 or newer. Native version 6.x support.
  - Version 5.x requires Version 4.x compatibility libraries
  - Apache or compatible Web server[1]
  - SSL support is required for PCI compliance
  - Sendmail or compatible SMTP server
  - 1Ghz or better Intel/x86 Compatible CPU
  - 256MB RAM
  - 40MB Hard Drive space for program files
  - 40MB Hard Drive space for installation files[2]
- ***Solaris***
  - Version 2.6 for SPARC or newer[3]

  - 64-bit systems require some 32-bit libraries
  - Apache or compatible Web server[1]
  - SSL support is required for PCI compliance
  - Sendmail or compatible SMTP server
  - Sun SPARC CPU
  - 256MB RAM
  - 46.2MB Hard Drive space for program files
  - 46.2MB Hard Drive space for installation files[2]

- ***Windows Server***
  - Windows 2003 Server or 2003 Server
  - Internet Information Server (IIS) 5.0 or newer[1]
  - NTFS File System
  - Sendmail or compatible SMTP server
  - 1.6Ghz or better Intel/x86 compatible CPU
  - 512MB RAM
  - ~50MB Hard Drive space for program files
  - ~50MB Hard Drive space for installation files[2]

Whichever operating system you are hosted on, it is an extremely good idea to make sure your operating system and other software is up to date. Software updates frequently include security and reliability enhancements, which can significantly contribute to security and stability on your Web server. Contact your system administrator or hosting provider if you have questions about keeping your server up to date.

Footnotes:

1. ShopSite will work with most NCSA compatible Web servers such as Apache and IIS. Your Web server must be configured to use standard NCSA-style authentication, and should have SSL capability.

2. Hard Drive space required for installation is in addition to the space required for the program files, but is *only* required for the installation. If you have limited available space, you may remove the installation files after ShopSite has been installed.

3. If you are running Solaris 2.6, you must have the complete 2.6 recommended patch cluster installed, including Sun Patch 107733-10.

ShopSite Auth File Information

A ShopSite auth file is the digital equivalent to the keys of your store; it is used to open your store, and to help protect your store and the information your customers entrust you with. You can not install or run ShopSite without a valid auth file, which can only be used with your store.

When you obtain a ShopSite store license from a ShopSite reseller, such as your hosting provider, a unique auth file is generated. The auth file contains information specific to your store, such as your **storeid** (the username you use to log in to your ShopSite back office), a unique serial number, the store URL, and the service level for your store. This information is stored in an encrypted format in your auth file for your security.

Be sure to use ASCII (text file) format any time you use FTP to transfer your Auth file.

In order to install or upgrade ShopSite, you must put the auth file in a location where the install programs can find it. On Linux and UNIX servers, the auth file must be in the install directory, with the name `store.auth`. Windows-based installations will prompt you for the auth file during the installation process.

Once ShopSite has been installed, your auth file is stored in your store's data directory.[1] with the name *`storeid`*`.auth` (where *storeid* is the username you use to log in to the ShopSite back office).

Replacing an Auth File

There are several possible situations where you would be required to replace your store's auth file with a new one. Your auth file contains various information used to uniquely identify your store, so you will require a new auth file any time that information changes. This includes changes to your store URL or the service level of your store. You should *not* change the storeid of an existing store.[2] You may also need to obtain a new auth file if you are moving your store to a server running a different operating system.

Your ShopSite reseller can provide you with a new auth file if you need to replace it. When you request a new auth file, your ShopSite reseller will need the following information (see the **Store Settings** and **Product Information** sections of your **Preferences** > **Hosting Service** screen for this information):

- **storeid** - the username you use to log in to your ShopSite back office (This is the `ID` under **Store Settings**).
- **store URL** - the URL of your ShopSite store directory (Be sure to use the `Store URL` in the **Store Settings** section, *not* the `"My Store" URL` in the **Merchant Settings** section).
- **serial number** - the unique serial number for your store (listed in the **Product Information** section).
- **service level** - the feature level of your store (the `Product` in the **Product**

Information section).

After you have obtained a new auth file, you can install it by doing the following:

1. **Backup your current auth file.** If something is wrong with your new auth file, you may need to restore the old one until you can get your new auth file replaced. Your auth file is located in your store's data directory1, and is named `storeid.auth` (where *storeid* is the username you use to sign in to your ShopSite back office). You can make a copy of the file with a different name (`storeid.auth.old`, for example) in the same directory, or on your local PC. **Be sure to use ASCII (text file) format** any time you use FTP to transfer your Auth file.
2. Save the new auth file as `storeid.auth` (where *storeid* is the username you use to sign in to your ShopSite back office) in your store's data directory1. Make sure the file name matches the storeid of your store, or ShopSite will not run.
3. Log in to your ShopSite back office and check that the new auth file is installed correctly. The store version and service level are indicated at the bottom of the main back office screen, followed by your storeid. If you have any problems, see the troubleshooting section below.

Troubleshooting Auth File Problems

If you are having trouble installing or upgrading ShopSite, or if you are unable to access ShopSite after installing or upgrading ShopSite, your auth file could be part of what is causing your problems. The following are a few things to be aware of with regard to your auth file:

1. **Did you use ASCII format to transfer your Auth file?** - Any time you use FTP to transfer your Auth file, you should use the ASCII (text file) format. If you transferred the file using BIN format, the end-of-line characters in the file may not match the ones used by your operating system, causing the file to appear corrupted to the Web server. **Always use ASCII format to transfer your auth file via FTP**.
2. **Is your Auth file in the correct location?** - If you are installing or upgrading ShopSite on a Linux or UNIX server, you must place a copy of your auth file in the install directory, and name the file `store.auth`. Once the store is running, the auth file must be located in the data directory with the name `storeid.auth` (where *storeid* is the username you use to log in to the ShopSite back office).
3. **Can your Web server read your auth file?** - Your Web server must have permission to read the auth file in order to run or install ShopSite. See Linux/UNIX Permissions or Windows Permissions for more information.
4. **Does your auth file match your settings?** - Your auth file includes the storeid, store URL, and other information specific to your store. If you have changed any of these settings, you will need to get a new auth file with the new information in it. Contact your ShopSite reseller for more information.

Footnotes:

1. The data directory is where the information unique to your store is kept. For more information see Data directory for Linux/UNIX Installs or Data Directory for Windows Installs help for more information.

2. The storeid and serial number are used as part of a system that associates shopping cart and order information with *only* your store and encrypts the information so it can not be viewed by anyone else. If you were to change your storeid or serial number, you would no longer be able to view or process orders placed prior to the change.

Using a Secure Server with ShopSite

ShopSite is designed to use a secure Web server when transmitting sensitive data, such as payment and order information. In order for this to work, ShopSite must be installed on a Web server with SSL and an SSL certificate, and you must configure ShopSite to use the secure server information in the ShopSite back office **Preferences > Hosting Services** screen.

In order to avoid getting errors or warnings when using ShopSite with SSL, you should use the same domain name in your ShopSite secure URL as the domain name of the server's SSL certificate. The secure domain name and the store domain name do not have to be the same, but the secure URL should include the `https://` prefix (note the "s" at the end) instead of the insecure `http://` prefix.

If your store is sub-hosted on a server with other domains, you may need to indicate a path in addition to the secure domain name in the ShopSite back office. For example, your store may have the domain name `mystore.com`, but it may actually be sub-hosted in the `mystore` sub-directory of a server with the domain name `mysecureserver.com`. In this case, the **Store URL** would be `http://mystore.com/`, but the **Secure URL** would be `https://mysecureserver.com/mystore/`. Your hosting provider should be able to help you determine the correct secure URLs for your store, if you are unsure.

Linux and UNIX File Permissions Explained

The Linux and UNIX operating systems use a system called **file permissions** to manage access to content stored on the computer. File permissions provide security for the files, preventing unauthorized access or changes to the files.

> It may help you to think of the computer's hard drive as a series of file cabinets. File permissions are like keys to the cabinets, controlling who can open the drawers and work with the files. The directories on the computer are like the individual drawers and folders in the cabinets. If you have many people who need to use the files in the cabinets, you can set aside individual areas for specific people, and you may also have areas shared by several people. To help avoid mistakes, and to prevent anyone from seeing things they shouldn't, each person could be only given keys to cabinets he or she needs to be able to access.

In Linux and UNIX, anything stored on the computer's hard drive counts as a file, whether it's a program, a directory, or a regular file. Each file has its own file permissions which are used to tell the operating system what type of file it is, as well as who can access or modify it.[1]

This document includes the following sections providing more information on file permissions:

Permissions in Detail

There are three **classes** of file permissions which control the degree of access for each file. The first class is the **owner**, meaning the user account to which the file belongs.[2] The second class is the **group**, which makes it possible for specific other users on the system to have access to the file. The final class is for **everyone** with a user account on the system.

For each class, the owner of the file can specify three levels of access rights for the file. The first is **read** permission, the ability to see the contents of the file. Second is **write** permission, the ability to modify the contents of the file. The final level is the **execute** permission, which is used to control the ability to run a program file.

The following is an example of a directory listing displaying file permissions on a typical Linux system (run `ls -al` from a command prompt):

```
drwxrwxr--    5 apache      apache        4096 Feb 28 15:19 .
drwxrwxr--    3 myusername  myusername    4096 Feb 28 15:17 ..
```

```
-rw-rw-r--    1 apache      apache        33298 Feb 28 15:49 index.html
drwxrwxr--    7 apache      apache         4096 Feb 28 15:32 media
-rwxr-xr--    1 myusername  myusername      178 Feb 28 15:21 rmversion
```

Each row in the example displays information about the file identified in the final column of the row.[3] The file permissions are displayed in a format called the **file mode** in the first column. The third column displays the name of the user account that owns the file, and the fourth column displays the name of the group the file belongs to.

The file mode contains ten characters which indicate the file type and permissions for each of the three classes. The file permissions for each class are indicated in sets of three characters, starting with the second character[4]. The first set indicates permissions for the owner of the file, the second set for the group, and the third set for everyone. The first character of each set will be an **r** if read permission is granted. The second character in the set uses a **w** to indicate write permission has been granted. The third character in the set uses an **x** to indicate that execute permission has been granted. If a **-** (**dash**) appears in any position, the permission that position indicates is not granted.

> In the example above, you can see that the mode for *index.html* is `-rw-rw-r--`. The user permissions (positions 2, 3, and 4) are `rw-`, indicating the user has read and write permissions, but not execute. The group permissions (positions 5, 6, and 7) are the same as the user. The permissions for everyone (positions 8, 9, and 10) are `r--` meaning that only read access is allowed.

Permissions and Programs

Because a program is a file, the program is subject to the file permissions granted to it. In order to run a program file, a user must have permission to read and execute the program file. If the user does not own the file, but is a member of the group the file belongs to, the file must have group read and execute permissions. To run a file that does not belong to the user, or to a group the user is a member of, the file must have read and execute permissions for everyone.

A running program is owned by the user who ran the program (*not* the user who owns the program file), and has the same rights that user has. This means that the program has permission to read, write, or execute the same files as the user who ran the program. A running program belongs to the group the program file belongs to.

A server is a type of program designed to grant remote users the ability to access files on the computer over the internet. Just as with any other program, the access a server has is limited by the file permissions granted the user the server is running as. A wise system administrator will create special user accounts to run servers in order to restrict what files the server has access to.

Web servers can provide the general public access to files on the computer the server is running on. This can be useful for providing information and services such as an online store, but requires special care to avoid allowing people to access files they should not.[5] In order to view an HTML page in a Web browser, the special account used to run the Web server (common names include *apache*, *web*, or *nobody*) must have permission to read that file. The safest way to give the Web server access to a file is for the file to belong to the same special user account and group that runs the Web server and restrict what permissions the file allows. Read access should be granted to the user and group, and may be granted to everyone. Write access should only be granted to the user and group if the file will be modified by the Web server or a program (such as a CGI) run by the Web server. Execute permission should be granted to the user and group if the file is a program, such as a CGI, or a directory. It is generally not a good idea to grant write or execute permissions for everyone.

Changing File Permissions

The owner of a file can change permissions for that file. The file owner can also assign the file to any group to which the user belongs. In order to change the owner of a file, the user must have administrative (**super-user**) access on the system.[6]

Changing the Mode

The **chmod** command can be used to change the file mode, granting or revoking read, write, or execute permissions for any of the classes:

```
chmod permissions filename
```

The *permissions* can be indicated in different ways. The standard way for doing this is by using a three-digit number to specify the desired file permissions by using one digit for each of the permission classes. The numbers one through seven are used for each digit to indicate which permissions to set on the file:

```
1 - execute (--x)
2 : write (-w-)
3 : 1 + 2 (-wx)
4 : read (r--)
5 : 4 + 1 (r-x)
6 : 4 + 2 (rw-)
7 : 4 + 2 + 1 (rwx)
```

Using this system, you can set **absolute** permissions for the file, meaning that the permissions are changed to whatever you indicated, regardless of what they were before. For example, if you ran `chmod 775 myfile`, the file *myfile* would have the mode `-rwxrwxr-x`.

While using the standard system to specify file permissions is quick and uncomplicated, there are two significant limitations to the standard system. The primary limitation is that it's not intuitive or easy to remember. The second limitation is that you can only do absolute permission changes. There is a second system which, while it requires more keystrokes, is easier to remember and can be used to make **relative** file permission changes. The advantage of relative changes is that you only change the permissions you want to, leaving anything else the way it was.

To use the second system to specify file permissions, you simply have to remember the three types of classes and the three types of permissions. For each class and permission, there is a one-character memnonic code:

```
u : user class
g : group class
o : other (everyone) class
r : read permission
w : write permission
x : execute permission
```

In addition to the codes, there are three operators:

```
+ : add permissions
- : remove permissions
= : set to specified permissions
```

Using this system, you indicate which classes to change (if you do not specify a class, all will be changed), how to change them, and what permissions to change. For example, you could run `chmod +r myfile` to add read permissions for all classes to *myfile*, or you could run `chmod u=rw myfile` to give the owner of *myfile* read and write permissions, but not execute. You could also run `chmod ug+rw,o-wx myfile` to add read and write permissions for the user and group, and remove write and execute permissions for everyone. This system is especially useful when using wildcards[7] to change permissions on multiple files at a time.

Changing the Owner

The **chown** command changes the file owner:

```
chown username filename
```

The *username* must be a valid user account on the system. You can use wildcards[7] in the *filename* to change multiple files at once. Only a user with administrative access[6] can change the owner of a file.

Changing the Group

The **chgrp** command changes the group the file belongs to.

```
chgrp groupname filename
```

The *groupname* must be the name of a valid group on the system to which the user running the command belongs. You can use wildcards[7] in the *filename* to change multiple files at once. A user with administrative access[6] does not have to be a member of the target group.

Footnotes:

1. When a user accesses a file, it is done through a program, such as FTP or a shell. The program, not the user, is actually what has access based off the user account the program is running as.

2. Ownership of files is similar to employee ownership within a company. The files actually belong to the computer, but the user account has the rights and responsibility for using the file.

3. The `ls -al` command outputs seven columns of information for each file listed. The name of the information in each column is listed above the column in the example below:

```
mode        nodes owner       group         size   modified time  filename
drwxrwxr--  3     myusername  myusername    4096   Feb 28 15:17   ..
```

The **mode**, **owner**, and **group** are discussed above. The **nodes** column displays the number of files in a directory. The **size** is the file size, usually in bytes. The **modified date** is the time and date of the last time the file was changed. Finally, the **filename** is the name of the file.

4. The first character in the mode is used to indicate the type of file. A `-` (dash) indicates a normal file, a `d` indicates a directory. There are also other less common characters for other special file types.

5. Web servers can provide unauthenticated access, meaning anyone with a Web browser and an internet connection can access Web content without identification. While this is desirable for some things, allowing an anonymous user the ability to change system files, such as the Web server settings or user passwords, invites disaster. While the practice of managing file permissions and using restricted user accounts may be inconvenient, the protection it provides is well worth the effort.

6. In order to manage a server properly, there needs to be a special administrative user account with the ability to override any permission settings. This special user account is often called the **superuser** or **root**. Only a superuser has the ability to change the owner of a file. Contact your hosting provider or system administrator for help changing file ownership or handling other administrative tasks.

7. Wildcards are special characters used to represent a range of possible characters. For example, if you want to change the group of every file in a directory, you may want to use a wildcard to indicate *every file* instead of having to specify each individual filename.

There are two wildcards you will probably want to use, `*` (**asterisk**) and `?` (**question mark**). The `?` can be used to represent any single character in a string. The `*` can be used to represent any number of characters in a string. For example, `chgrp web *` would change the group for all files in the current directory, `chgrp web *.html` would change the group for all files with a name ending with *.html*. If you had *.html* and *.htm* files, you could use `chgrp web *.htm*`, but that would also change *myfile.html.old*. A better way might be to use `chgrp web *.htm?`. Another occasion to use the `?` might be if you use a number to indicate the version of a file, such as *myfile.version1.html* and *myfile.version2.html*. You could use the command `chgrp web myfile.version?.html` to change only files with a version number less than 10.

Creating a ShopSite Mall

ShopSite merchants who manage more than one online store, or shared hosting providers, can install multiple ShopSite stores on a single server as a **Mall**. A ShopSite mall is a group of multiple individual ShopSite stores that use a common set of ShopSite CGIs while keeping individual store information seaparate. This can provide a significant savings in disk space requirements, but has no effective impact on the individual stores.

Note:

Most ShopSite resellers and Web hosting providers will automatically install ShopSite on your server for you. You should contact the reseller or hosting provider if you have questions about your ShopSite installation.

Warning:

Whenever connecting to your server for administrative purposes (such as installing or upgrading ShopSite), you should use a secure connection. Telnet and standard FTP are both unencrypted connections, and could be intercepted. Always use a secure connection, such as SSH for shell connections, and FTP over SSH or SFTP for file transfers.

Overview

A ShopSite mall is not fundamentally different from a ShopSite store, as far as the resources required; you can, in fact, upgrade an existing store to a mall.

- Installing a New Mall
- Converting a Store to a Mall
- Adding Stores to a Mall
- Upgrading a Mall

The instructions below are applicable to both Linux/UNIX and Windows servers, but to avoid unnecessary confusion, the link references are only for Linux/UNIX installations. For Windows servers, the referenced information can be found on the Installing ShopSite on Windows Servers help page.

Installing a New Mall

Installing a new ShopSite mall is the same as installing a store. The initial mall installation, in fact, is done by installing the first store. The key difference is that when planning the locations for your content, you may want to use different locations and URLs to better accommodate multiple stores. The best way to set up your mall depends greatly on the way your Web server is set up, and how you will be accessing the individual stores and shared CGIs. The following are a few things you may want to consider when planning to install a new mall:

- **Secure Web Access**
  Secure access to Web sites is achieved using SSL. One limitation of SSL is that each IP address can only have one SSL certificate. If you do not want your customers to get warnings about the SSL certificate not matching the domain name, you will need to make sure that the secure Shopping Cart URL uses the same domain as the SSL certificate.
  - Many modern Web servers can have multiple IP addresses, each with its own SSL certificate. This would allow you to have a unique certificate for each domain name while still sharing the CGIs.
  - If multiple IP addresses is not an option, you can use a single domain with an SSL certificate for all the stores. Each individual storefront could have a unique domain name, but secure URLs would use a common domain name. This would not affect the appearance or functionality of the individual stores.
- **Access to Common CGIs**
  The common CGIs shared by all the stores in a mall must be accessible to each store in the mall. This means that each indivdual store and the shared CGIs must have the correct file permissions to allow the Web server access, and that each store can access the CGIs through the Web server.
  - The file system (FTP and Shell) path of the CGIs does not matter as much as the Web path of the CGIs. You could put the CGIs in the primary Web server path, or in a sub-hosted domain, as long as the Web server can see and run them.
  - If the CGIs will be accessed using each individual store's domain name, you will need to make the common CGI directory visible to each domain name. This can be done by creating Web server aliases or file system symbolic links (shortcuts) to the common CGIs for each store.
  - If the CGIs will be accessed using a single common domain name for all the stores, you will need to make sure the common domain has access to the HTML and data directories for each of the individual stores. This can be done by creating Web server aliases or file system symbolic links (shortcuts) to the data and HTML directories of each of the stores.
- **Protecting Merchant Privacy**
  Sharing CGIs does not mean sharing DIP individual merchants should not have access to the data or HTML directories of other merchants. If your merchants have FTP or Shell access to their HTML directories, you should set up the permissions so that they can not see or modify the other HTML directories of the other merchants.
  - The common CGIs do not need to be accessible to any merchants through FTP or Shell. The best location for the common CGIs is outside any individual merchant's home directory. You may put the CGIs under the root Web server, or under an individual subhost, so long as none of the merchants have FTP or Shell access to them.
  - Although each store has its own data directory, the data directories do not need to be accessible to the merchants through FTP or Shell access. You may want to congregate all the data directories in a common location

outside the home directories of all the merchants. If, for example, you put the common CGIs in the `/www/cgi-bin/` directory on the server, you might want to create a `/www/shopsite-data/` directory where each individual store's data directory would be.

- Many merchants may want or need FTP access to their HTML directories, although they should avoid making changes to any content generated by ShopSite. If you grant your merchants FTP access, you should make sure they can not access the common CGIs or the HTML directories of other stores. This requires special care when setting up file permissions to allow the Web server access to all the HTML directories.

Once you have determined where to put the various content and have made any configuration changes necessary, you are ready to begin installing your mall. Select one of the stores to start with, and follow the instructions for installing a single store. Once you have one store running, follow the instructions to add more stores to your mall.

Converting a Store to a Mall

If you already have a store installed and running on your Web server, converting it to a mall is as easy as adding another store. If the first store was not initially installed to be a mall, however, there may be some privacy and access considerations you will need to adapt for.

Review the instructions for installing a new mall above, and determine how to address the factors discussed. In most cases, you can make file permissions and Web server configuration changes to accommodate your current setup. In some cases, however, you may need to re-install the store.

To re-install an existing store in a new location on the same server:

1. **Do not** try to upgrade the store at the same time you are re-installing it. If necessary, upgrade the store first, *then* move it after you have confirmed that the upgrade was successful.
2. **Back up your existing store.** Make sure your store's **data directory** and **store pages directory** (including the **media directory**) have been backed up before you make any changes to your store. It is a good idea to store a backup copy of your store content in a compressed archive on your local computer.
3. If moving the store will change the URL of the store pages directory, you will need to acquire a new auth file for your store.
4. In order to avoid any complications, it is a good idea to disable your current store *before* installing the new one. You should make the move at a time that will not have a significant impact on your customers.
5. If you are moving the store pages directory, create the directory and copy *all* the content from the old directory to the new location.

6. Disable the old store by removing Web access to the old CGIs. You may want to wait until you know the store is working in the new location before actually removing the old CGIs.
7. Make any changes to your Web server configuration that may be required for the new store location.
8. Follow the instructions to install a new store using the new location on the server.
9. Once the new store is installed, copy *all* the contents of your old data directory over the contents of the new data directory. This will replace the new store settings with the settings from your old store.
10. Log in to the merchant back office for the new store and check that your old settings are intact. You may need to re-select your Payment Processor, although the settings for that processor should be intact.
11. If the move affected the URL of your media directory, you should **reset all attributes** from the **Images** management screen.
12. Check the **Preferences** > **Hosting Service** screen and make sure all the settings are correct for your new location.
13. If you have any hard-coded URLs anywhere in your store content that are affected by the move, update them.
14. **Publish** the store and check to make sure there are no errors or missing content on your store pages. You may also want to go through and place a test order to make sure everything is working correctly.
15. Once the new store is working, you should *delete* the old CGI directories, as well as any other content from the old store that is not being used by the new store.

Adding Stores to a Mall

Once you have a mall running with a single store, adding more stores is simple and straightforward. The steps for adding stores are the same, no matter how many more stores you add:

1. Adding stores to a mall uses the same install programs as installing a new store. If necessary, follow the instructions to obtain and upload the ShopSite installation files.
2. If necessary, upgrade the existing stores before adding new stores. Your existing stores *must be running the same version* of ShopSite as the install programs used to add new stores.
3. Obtain an auth file for the new store and copy it to the **Install CGI Directory** with the filename `store.auth`.
4. Enter the Install Directory URL followed by `/wwwinstall.cgi` into the location bar of your Web browser, then press **Enter**.

- Example Install URL: `http://www.mystore.com/cgi-bin/shopsite-install/wwwinstall.cgi`

5. Select the radio button under **Existing ShopSite** to **Add a store**, then click **Continue**.
6. The next screen is where you will enter the settings, including directory locations and URLs, for the store. See the installation instructions for new stores for an explanation of these fields. Note that any fields that can not be changed from the initial store are not available when adding a store.
7. If you have not already created the directories ShopSite will use, check the box for **Do you want Install to create any directories that do not already exist?**. If you set up your permissions properly, ShopSite should be able to create these directories.
8. Click **OK** to add the new store with the settings you specified. The progress of adding the new store to your mall will be displayed on the screen. When the store has been added, the words **Installation Completed** will be displayed in large text at the end of the screen.
9. Log in to the new store to make sure everything is working as it should. If there are any problems, see the troubleshooting information for new stores. Any problems you encounter adding a store are likely to be caused by the same things that would cause problems installing a new store.

Upgrading a Mall

Upgrading a mall follows the same process as upgrading a store. Because all stores in a mall use the same CGIs, all the stores will be upgraded to the new version, but new features may not be available for all the stores.

Any stores which will not incur additional fees to upgrade will automatically get all the new features. There are usually no additional upgrade fees for **point release upgrades** (e.g. ShopSite 8.0 to 8.3), or for store licenses leased on a monthly basis.

Stores which will incur additional fees to upgrade will *not* automatically get all the new features. They will still have the new CGIs, but will not have access to the new features until you manually enable the new features. *If you manually enable features for such stores, you will be charged the relevant upgrade fees*.

To enable new features in an upgraded store:

1. Enabling new features uses the same install programs as installing a new store. If necessary, follow the instructions to obtain and upload the ShopSite installation files.
2. You *must* use the correct install programs for the version of ShopSite your mall is currently running. If necessary, upgrade the existing stores before enabling new features for other stores.
3. In order for the install programs to run, you must copy the auth file for one of the stores to the install directory, and name the file `store.auth`.
4. Enter the Install Directory URL followed by `/wwwinstall.cgi` into the location bar of your Web browser, then press **Enter**.
    - Example Install URL: `http://www.mystore.com/cgi-bin/shopsite-install/wwwinstall.cgi`

5. Select the radio button under **Existing ShopSite** for **Mall feature administration** and click **Continue**.
6. From the drop-down menu, select the storeid for the store you want to enable new features for.
7. Select the radio button labeled **Enabled** and click **Update Store**. The selected store will have the new features enabled, and you will be charged the relevant fees.