Zelerate AllCommerce System Administrator’s Guide

Release 1.2 Beta

Preface

The *Zelerate AllCommerce System Administrator's Guide* gives you detailed information about installing and setting up Zelerate AllCommerce effectively.

Audience

Zelerate AllCommerce System Administrator's Guide provides complete information for the administrator who installs, configures, and customizes Zelerate AllCommerce. It is also a good source of information for developers who are extending the applications.

Zelerate AllCommerce System Administrator's Guide assumes that you have a basic understanding of the Linux operating system. It also assumes that you are familiar with most of the following tasks:

- Managing web sites
- Setting up servers
- Installing databases
- Installing Perl
- Defining business rules for applications
- Creating users and permissions

How This Document is Organized

- Part I: Installing and Configuring Zelerate AllCommerce provides detailed instructions on installing the product automatically with a script or manually.
- Part II: Using Command Line Tools and Utilities discusses how to import data and set up the AllCommerce installation with tools and utilities.
- Part III: Managing Permissions and Users describes how to grant and revoke user permissions and how to create and manage users.
- Part IV: AllCommerce Technical Reference provides information about directory and data structures and other details about the system.
- Part V: Appendices contains information about AllCommerce directory structures and scripts.

Each part begins with a functional overview, which describes key concepts; each section describes Zelerate AllCommerce features and how to use them; it also provides any relevant reference information.

Typographical Conventions

This document uses the following typographical conventions:

Items	Description	Convention
Code samples	Pieces of HTML or Perl code	10 point Courier font
Warnings	Critical information; necessary to ensure that product works properly.	*Warning!*
Attention notes	Important information that could save you time or something you need to remember for the task at hand.	*Attention!*

Related Information Sources

Ideally, you should consult the following information sources in addition to using this document.

Zelerate Information Sources

For the most complete information about Zelerate AllCommerce, see also the *Zelerate AllCommerce User's Guide*.

External Information Sources

http://www.opensource.org/

Zelerate Training

Zelerate offers comprehensive training to help you master Zelerate AllCommerce and use it effectively. See http://www.zelerate.com/ to find out about how to sign up for a course.

Thank You!

Thank you for using Zelerate AllCommerce documentation.

We value your input and would like to hear your comments. Drop a note to doctalk@zelerate.org and let us know of your suggestions and comments for improving our documentation. If you come across a serious error in the documentation that stalls you, or find missing or outdated information, log a documentation bug at bugs.Zelerate.org.

Table of Contents

I. Installing and Configuring Zelerate AllCommerce

This procedure will guide you through the installation procedure for Zelerate AllCommerce for Linux.

This installation procedure assumes that you have installed and configured all the needed prerequisites to run AllCommerce.

Installation Checklist

Step	Task
1	Prepare for AllCommerce installation by installing other required software. See Installing Necessary Software. See Platform-Specific Installation if you are installing on Windows NT 2000.
2	Expand the tarball. See Platform-Specific Installation if you are installing on Windows NT 2000.
3	Choose installation mode: automatic or manual.
	3a. Execute the configure.pl script if installing automatically. See Installing Automatically with a Script.
	3b. Follow the step-by-step installation process using a command line if installing manually.
4	Troubleshoot the installation if necessary.

Installation Procedure

Complete steps 1 and 2 for either automatic or manual installation.

1. Installing Necessary Software

You need to ensure that you have the following software installed first. See Platform-Specific Installation if you are installing on Windows NT.

- Perl 5.005 or higher
- A database from the list below for which you can find a database driver.
- Database drivers (i.e., DBD & DBI)
- A web server (e.g., Apache)
- libwww-perl (optional, required for cookieless shopping)

All of these prerequisites are readily available for the Linux operating system. Other operating environments, such as Unix, Solaris, FreeBSD, and Windows 2000, are also supported.

Zelerate strongly advises that you install the prerequisite software before continuing with the installation procedure.

Perl

AllCommerce requires that you have Perl installed and that it is working properly. You can find out if you have Perl installed by entering `perl -v` from a shell prompt.

If you have Perl installed, this command will return a response like:

```
This is perl, version 5.005_03 built for i386-linux
Copyright 1987-1999, Larry Wall
...
```

The version number should be of the form "5.005_xx" or "5.x.y". If you do not receive this response, or if your Perl version is not the supported version, you will have to install (or update) Perl before continuing with the AllCommerce installation.

Databases

AllCommerce requires an installed database management system. Currently, it has built-in support for MySQL. We recommend Informix, mSQL, Oracle, PostgreSQL.

Attention!

Support for these databases is currently experimental and is not stable in this release. Check http://www.zelerate.org/ for updates.

Database Drivers (DBI and DBD)

Perl will need DBI and DBD drivers in order to communicate with your selected database.

DBI is a database interface API for Perl. It allows AllCommerce to smoothly connect to a database. DBI defines an API, but it omits the implementation-dependent details for the specific database. DBD Drivers are written to bridge the gap, implementing the API for a specific independent database.

The DBI alone is not sufficient, as it is merely an interface between your Perl code and a specific database driver. Therefore, the DBI Driver needs a functional DBD Driver to communicate with the database.

A Web Server

AllCommerce requires that you have a Web server installed and working properly before installing it.

At this time AllCommerce is optimized to only run with the Apache Web server. However, there is a ported version of AllCommerce available which runs under the Microsoft IIS server. In future releases of AllCommerce the ported functionality for IIS will be included in the standard release.

Wrapper Interface for Cookieless Shopping

AllCommerce can provide cookieless browsing, using the "wrapper" interface. The wrapper interface is dependent on the `libwww` library for Perl. See Appendix E: Implementing Cookieless Shopping in the *Zelerate AllCommerce User's Guide* for more information. This step is optional; however, if you are using the wrapper interface, you need to ensure that `libwww-perl is working properly`.

Ensure that the following packages are installed:

```
Digest::MD5
```

```
HTML::Parser
MIME::Base64
URI
```

You can obtain these packages from www.linpro.no/lwp or http://www.cpan.org/.

2. Expanding the Tarball

Complete the following steps to expand the tarball:

1. Move the file `os_allcommerce.1.0.4.tar.gz` to the directory that your Web Server uses for served data. If you have Apache installed, the Apache default directory should be `/home/httpd`.
2. Untar Zelerate AllCommerce by entering:
   ```
   tar -zxvf os_allcommerce.1.0.4.tar.gz
   ```
3. Now check the directory ownership:
   ```
   ls -ld os_allcommerce

   In the case of a default Apache installation, you should see
   something like:
   drwxr-xr-x 15 nobody nobody {size} {mm dd time} os_allcommerce
   ```
4. If the directory is not owned by `nobody.nobody`, enter:
   ```
   chown -R nobody.nobody os_allcommerce
   ```
5. Use `ls` to verify that the change in ownership took effect.
   ```
   In general, you will want all files and directories to be owned
   by the account under which the Web server will be run
   (typically, this is nobody).
   ```
6. If you wish to conserve disk space, you may now remove the tarball by entering:
   ```
   rm os_allcommerce.1.0.4.tar.gz
   ```

3. Choosing the Installation Mode

If you are a non-technical user, you may want to use the automatic installation procedure. This procedure uses a script that sets up and configures various aspects of the product prompts the user for various data throughout the installation process. Manual installation requires familiarity with working with a command line.

3a. Installing Automatically with a Script

The Perl script that automatically installs Zelerate AllCommerce `is configure.pl`.

At this time, `configure.pl` works with the MySQL database (running locally) and with the Apache Web Server. If you are not using MySQL as your database for AllCommerce, you will need to install and configure AllCommerce manually. See *Installing Manually*.

Executing the Script

If you are ready to run `configure.pl` and begin the installation process, you will need to be in the `os_allcommerce` directory.

From a shell prompt, type:

```
./configure.pl
```

to execute the script and begin the installation process.

The Installation Process

`configure.pl` performs several dependency checks; it then allows you to perform a step-by-step installation of Zelerate AllCommerce. The `configure.pl` script makes sure that the prerequisites are installed and working properly.

Checking for Perl and Perl Modules

This step will perform a check to see if Perl is installed and will check to see if the libraries needed for cookieless shopping are present.

```
#####
#
# Zelerate AllCommerce Version 1.2.1
#
#####

Welcome to the Install Script for Zelerate AllCommerce.
This script will guide you through the installation
process.

Do you wish to continue with this installation? (y/n)
```

```
#####
#
# Perl Setup
#
#####

Checking Perl version ... using Version 5.00503 ... ok.

libwww-perl and other needed libraries from CPAN are not
installed for cookieless shopping, please refer to the
install notes for more details.
```

If you have not installed all the necessary prerequisite software packages, such as Perl or the needed Perl library for cookieless shopping, you will see a message explaining what is missing. For more information on the prerequisites needed to run AllCommerce, see Installing Necessary Software.

Checking for DBI and DBD Drivers

This step performs a check to make sure that DBI (including the needed DBD drivers) are installed and working properly.

```
#####
#
# DBI Queries
#
#####

Checking for available DBI drivers ... done.
Listing each DBI driver with recognized data sources (databases) ...

Driver: mysql
(1)Data Source is DBI:mysql:mysql
(2)Data Source is DBI:mysql:test
```

If DBI and DBD are installed and working properly, you will see a list of data sources that can be used with AllCommerce.

Checking for the Correct Database

This step allows you to create a new database table or use an existing database.

If you have multiple databases installed, AllCommerce prompts you to select the appropriate data driver for use with AllCommerce.

```
#####
#
# Database Driver
#
#####

Which database driver will you be using? [mysql]
```

If you are using `MySQL` as your database, press the return key. If you are using one of the other stable supported databases, enter the name of the data driver and press return.

```
#####
#
# Database Table Setup
#
#####

Do you wish to create a new database table? (y/n)
```

Type [y] to Continue and create a new database table. See Create New Database Table.

Type [n] to Continue and use an existing database table. See section Use Existing Database Table).

Create New Database Table

This step will walk you through the process of creating a new database table for use with AllCommerce.

Default values are shown in brackets.

```
Which database driver will you be using? [mysql]
```

This step is asking for the name of the database driver you are using for AllCommerce. For example, if you are using MySQL as your database driver, you would enter `mysql`.

```
What is the name of the database? [osallcommerce]
```

This step is asking for the name of the database table you wish to create to use with AllCommerce. Suppose you are setting up a store to sell shoes; you may want to call the database table `shoes` or even `bigfeet`.

```
What is the database username? [root]
```

This step is asking for the database account user name. If you have not set up an account for the database user in MySQL, the default user will be `root`.

```
What is the database password? []
```

This step is asking for the database account user name's password. If you have not set up a password for the database user in MySQL, the default password for the database user should be left empty.

```
What is the database servername? []
```

This step is asking for the server name of the machine which is hosting your database. This machine can be either a separate server or the same server on which you have AllCommerce installed. If you are running AllCommerce and the database server on the same machine, this setting can either left blank or set to `localhost`.

```
What is the database port? [3306]
```

This step is asking for the communications port is used to communicate with the database server. In the case of MySQL, if you are using a separate database server, the default communications port should be 3306.

Use Existing Database Table

This step will walk you through the recreation of a new database table for use with AllCommerce.

```
Do you wish to recreate the database (i.e. drop and create)
  (y/n)?
```

Type [y] to Continue and select an existing database table. Type [n] to abort the installation.

```
Driver: mysql
(1) Data Source is DBI:mysql:mysql
(2) Data Source is DBI:mysql:osallcommerce
(3) Data Source is DBI:mysql:test

Which database ( e.g. 2 for (2) ) do you want to recreate?
```

This step is asking for the number of the database that you wish to drop and have recreated. Use the number that is listed next to the name you desire.

```
What is the username? [root]
```

This step is asking for the database account user name. If you have not set up an account for the database user in MySQL, the default user will be root.

```
What is the password? []
```

This step is asking for the database account user name's password. If you have not set up a password for the database user in MySQL, the default password for the database user should be left empty.

```
What is the servername? [localhost]
```

This step is asking for the server name of the machine which is hosting your database. This machine can be either a separate server or the same server on which you have AllCommerce installed. If you are running AllCommerce and the database server on the same machine, this setting can either be left blank or set to `localhost`.

```
What is the port? [3306]
```

This step is asking for the communications port which is used to communicate with the database server. In the case of MySQL, if you are using a separate database server, the default communications port should be 3306.

```
Are you sure you want to recreate the database osallcommerce
  (recreating the database will destroy any existing information
  in the database)? (y/n)
```

Type [y] to continue the installation. Type [n] to abort the installation.

Attention!

If you answer "y" to this step, all data in the database you are recreating will be dropped and overwritten and all information stored in this table will be lost.

Setting up the AllCommerce Data Structure

A data structure for AllCommerce will need to be imported into the database table that you have just created.

If the database table was created successfully, this step will create the AllCommerce data structure without any prompting.

```
#####
#
# AllCommerce Data Structure
#
#####

Creating tables ... done.
```

You will need to create a database table if this step was not done automatically. See Manually Creating a New Database Table.

Configuring a Web Server

A Web server will need to be configured for use with AllCommerce. At this time, `configure.pl` only supports the Apache Web server.

Apache Virtual Host

This section will walk you through the configuration of a Virtual Host for the Apache Web server.

Attention!

Due to the variations of Apache and SSL, you may need to edit your `httpd.conf` file in order for the system to work properly. The Apache configuration part of this configure script can be bypassed at the appropriate prompt, if necessary (this allows AllCommerce to be run with other Web servers).

```
#####
#
# Apache Virtual Host Setup
#
#####
```

```
Do you want to add virtual host entries into your Apache httpd.conf
(httpsd.conf) file? (y/n)
```

Type [n] to skip automated configuration of the Apache Web Server.

Type [y] to continue with the configuration of the Apache Web Server.

Please give the directory where your Apache file is located.

This step is asking for the directory in which your Apache Web server keeps the configuration files on your server. In a default Apache installation, this will be either `/etc/httpd/conf` or `/usr/local/httpd`.

```
Will you be running ssl? (y/n)
```

If you wish to run AllCommerce through a secure Web server, so user information is encrypted, you will want to enable ssl. This setting is used in various areas of AllCommerce which run in a secure mode.

```
ServerName?
```

This is the server name for the virtual host which you are configuring. Remember, this server name must either have a DNS record or an entry in your `/etc/hosts` file for Apache to work properly.

Server Name Example: `www.mydomain.com`

```
Port?
```

This is the port the Apache Web server uses to communicate. The default port for the Apache Web server is 80.

```
Server Admin?
```

This is an email address the Web Server can contact if a problem occurs.

```
Document Root?
```

This Document Root setting should be set to the directory which holds your AllCommerce file tree. In the case of a default AllCommerce installation, this will be `/home/httpd/os_allcommerce`.

```
Error Log?
```

The Error Log is the error log file name.

```
Transfer Log?
```

The Transfer Log is the transfer (access) log file name.

```
cgi-bin Directory?
```

The cgi-bin Directory should be set to the directory that contains your AllCommerce cgi-bin scripts. In the case of a default AllCommerce installation, this will be `/home/httpd/os_allcommerce/cgi-bin`.

`configure.pl` will now show you a preview of the entry it wishes to make in the `httpd.conf` or `httpsd.conf` file.

```
Write this output to Apache httpd (httpsd) conf file? [n]
```

Type [n] to skip saving the configuration for your Apache Web Server.
Type [y] to continue and save the configuration for your Apache Web server.

AllCommerce Configuration Files

AllCommerce has three configuration files that need to be configured in order to run the product. These configuration files contain almost all of the business rules for your site. The three configuration files which are set up with `configure.pl` are: `site.conf`, `machine.conf`, and `database.conf`. These files are opened automatically and will prompt you for the answers to various configuration parameters.

```
#####
#
# Configuration Files Setup
#
#####

Configuration file settings ...
```

Initial Data Load

An initial data load needs to be primed into AllCommerce in order for the platform to take on its personality as a commerce application. The initial data load is performed automatically if the database was created successfully and the import of the AllCommerce data structure was completed.

```
#####
#
# Initial Data Load
#
#####

Adding initial data load ... done.
```

If this step was not automated or displayed error messages, see Importing the Initial Data Load in a Manual Installation.

Shipping Methods and Rates Data Load

The shipping methods and rates data load allows you to use the download information from United Parcel Service (UPS), the U.S. Postal Service (USPS), and Federal Express (FedEx) to calculate shipping costs for a customer transaction.

Using information that you download from the shippers you want to use, this step translates these files into SQL insert statements that can be imported into your database.

```
#####
#
```

```
# Shipping Tables Data Load
#
#####

Running import_shipping_tables.pl ... done.

Adding shipping entries (This may take a few minutes) ...

No shipping entries to add.
```

If this step was not automated or displayed error messages, see shipping methods and rates data load during manual installation.

Successful Automatic Installation

After `configure.pl` has been successfully executed, you should be able to proceed to login to the AllCommerce admin screen and start creating your site.

```
https://sitename.sitedomain/cgi-bin/om/admin/login.cgi

username: admin
password: allcommerce
```

Using `configure.pl` Options

`configure.pl` also has several command line options that can be used while invoking the script, such as:

```
./configure.pl help
```

Attention!

If no keyword is provided after the script name, the entire installation process above is invoked.

The options supported in `configure.pl` are included in the table below:

configure.pl Option	What It Does
`help`	Provides a list of options
`database`	Checks Perl version, DBI, create new (or recreate) a database, load in the AllCommerce data structure
`idl`	Performs almost the same function as `database`, but this will also import the initial data load needed by AllCommerce
`conf`	Allows you to change only the AllCommerce configuration files
`httpd`	Allows you to append the Virtual Host settings (as long as they do not already exist) to the end of `httpd.conf`
`ship`	almost the same as `idl`, but allows you to import your shipping tables in order to properly execute this part of the script

3b. Installing Manually

Installing manually means installing step by step from a command line instead of running a script. You can install AllCommerce manually if you cannot run `configure.pl` or choose this method to install and configure AllCommerce.

This section walks you through the various steps of installing and configuring AllCommerce, creating a database, setting up a virtual host for your Web server, and discusses how to handle the various configurations and data loads that are needed for AllCommerce. The checks performed and steps are almost the same as an automated installation.

Attention!

Make sure that you are in the `os_allcommerce` directory when you use command line arguments.

Instead of `configure.pl`

If you had problems installing AllCommerce using `configure.pl`, you can start from the first manual step and begin the installation process again or you can move forward to the step where `configure.pl` ran into difficulties.

Checking for Dependencies

AllCommerce has several prerequisites that need to be installed in order for the system to work properly. Please make sure that the prerequisites are installed and working properly.

If you have not installed all the prerequisites that are needed for AllCommerce, we strongly advise that you install them before continuing on with the installation procedure.

Database Tables

AllCommerce uses a database table to store information that will be accessed by the ecommerce application. The information stored in the database table is a centralized point for all data within your site.

You can either create a new database or use an existing database table. We strongly recommend that you choose to create a new database table for use with AllCommerce.

Manually Creating a New Database Table

This step will walk you through the process of creating a new database table for use with AllCommerce.

To create a new database table go to a command prompt and type:

```
Mysqladmin create {new database name}
```

Manually Integrating with an Existing Database Table

To integrate with an existing table, follow the steps below.

From a command prompt type:

```
mysqldump -t mysql {existing database table} > data/{existing database table}.mysqld
```

This will save all data from the existing table you wish to use with AllCommerce. A copy of this data is stored in a file named {existing database table}.mysqld located in the data directory of AllCommerce.

```
mysqladmin drop {existing database table}
```

You will then see the following warning message:

```
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.
Do you really want to drop the '{existing database table}'
database [y/N]
```

Warning!

Please be sure you really are prepared to drop this data. Answering 'y' to this step means that all data stored in this database table will be lost.

Manually Setting up the AllCommerce Data Structure

AllCommerce requires that your database table be configured with a set of data structures specific to AllCommerce.

To import the AllCommerce data structure into the database table you created, type the following from a command line:

- For MySQL
  ```
  mysql {name of database table} < conf/tables.mysql.skel
  ```
- For PostgreSQL:
  ```
  PostgreSQL  {name of database table} < conf/tables.postgresql.skel
  ```

- For Oracle:
  You *must* use the script `configure.pl` for Oracle.

Manually Configuring the Web Server

AllCommerce requires that you have a Web server installed and working properly.

At this time AllCommerce is optimized to only run with the Apache Web server. However, there is a ported version of the AllCommerce available which runs under the Microsoft IIS server.

Apache Virtual Host

The Apache Web server will need to have several changes to the default configuration in order to work properly with AllCommerce. The Apache Web server configuration file which you need to edit is named `httpd.conf`. This file is located in the server root for Apache in a directory named `conf`.

For example, on a default installation of Apache, the server root will be either `/etc/httpd` or `/usr/local/httpd`.

Attention!

Zelerate recommends that you create an Apache virtual host for the site you wish to use with AllCommerce.

Sample Virtual Host Entry

Here's a sample virtual host entry for a typical AllCommerce installation:

```
<VirtualHost host.some_domain.com>
   ServerAdmin Webmaster@host.some_domain.com
   DocumentRoot /home/httpd/os_allcommerce/
   ServerName host.some_domain.com
   DirectoryIndex index.html index.shtml index.htm index.cgi
   ErrorLog logs/host.some_domain.com-error_log
   TransferLog logs/host.some_domain.com-access_log

   AddHandler cgi-script .cgi
   AddType text/html .shtml
   AddHandler server-parsed .shtml

   ScriptAlias /cgi-bin/ /home/httpd/os_allcommerce/cgi-bin/

   # allow access to the site
   <Directory "/home/httpd/os_allcommerce">
      AllowOverride all
```

```
        Options Includes FollowSymlinks ExecCGI
        Allow from all
    </Directory>

    # disallow access to everything else
    <DirectoryMatch "/home/httpd/os_allcommerce/(DOCS|bits|conf|crons
        |data|lib|skel|tools|transit|var)">
        Options None
        Order deny,allow
        deny from all
    </DirectoryMatch>

</VirtualHost>
```

Making Apache Work with AllCommerce

There are several important elements that need to be configured for Apache to work properly with AllCommerce.

DirectoryIndex

AllCommerce uses `index.cgi` as the directory index. You must make sure index.cgi is specified in the list for `DirectoryIndex`.

```
DirectoryIndex index.html index.shtml index.htm index.cgi
```

Directory Permissions

You will need to turn on some directory permissions in order for Apache to execute cgi scripts and follow symbolic links. These options are enabled using the `Directory` tag.

```
<Directory> </Directory>
```

ExecCGI and FollowSymlinks

The initial script, `index.cgi`, that welcomes users to your site and redirects them to the appropriate language and interface tree is a cgi script. Therefore, in order for the Web server to execute this script in the `os_allcommerce` directory you must give permissions to the Web server to execute `cgi` scripts.

AllCommerce uses symbolic links in the creation of its file names. You must allow the Web Server to follow symbolic links in the `os_allcommerce` directory.

The `ExecCGI` and `FollowSymlinks` options should be placed in the `Options` tag within the `Directory` brackets.

```
Options Includes FollowSymlinks ExecCGI
```

ServerName

This setting must match the combination of `siteserver` and `sitedomain` in the AllCommerce configuration file `machine.conf`. This will be the server and domain name you are configuring to run with AllCommerce.

Attention!

`ServerName` is the most common problem users have when configuring Apache to work with AllCommerce. If you do not have these parameters set correctly, you will experience a login loop problem when trying to use the AllCommerce administrator. This problem is caused because the Web server is unable to set a proper cookie.

Here is an example of a proper configuration for AllCommerce and Apache `ServerName`.

```
ServerName www.mysite.com
```

AllCommerce `machine.conf`:

```
      # The site domain name for the server you have OS
AllCommerce installed.
      # For example: '.mydomain.com'

      sitedomain      .mysite.com

      # The server name for your site.
      # For example: 'www'

      siteserver      www
```

You will need to restart your Web server after editing Apache `httpd.conf` for your changes to take effect. See Restarting the Server After Editing `httpd.conf`.

Warning!

Users of `openmerchant0.7.pre6` and earlier versions should note that the site configuration file `store.pm` has been changed to `site.pm`, which is now a machine-generated file. Some variables within have also had name changes. It is therefore (strongly) recommended that you do not attempt to reuse any existing `store.pm` files you may have and upgrade to this version of AllCommerce.

Manually Load AllCommerce Initial Data

You will have to load an initial data load into AllCommerce which contains default behaviors for this application.

To import the AllCommerce initial data load, from a command line: Change directory to the os_allcommerce directory and run:

```
./configure.pl idl
```

Manually Load Shipping Methods and Rates

AllCommerce uses real-time shipping methods in checkout. The shipping carriers supported in this release are: UPS, Fed Ex and USPS.

`The script import_shipping_tables.pl` allows you to use the download information from United Parcel Service (UPS), U.S. Postal Service (USPS), and Federal Express (FedEx) to calculate shipping costs for a customer transaction. By using information that you download from the shippers you want to use, `import_shipping_tables.pl` translates this files into SQL insert statements that can be imported into your database.

Weight Information

In order to charge shipping for an item, you must have weight information for each item being sold. This information should be placed in the database at the time of the creation of the object in the AllCommerce Administrator. Failure to provide weight information could result in incorrect shipping calculations.

Restrictions

`import_shipping_tables.pl` is designed for U.S. domestic service only. International shipping involves a greater set of problems (e.g., import taxes, currency, import restrictions).

Information Needed For `import_shipping_tables.pl`

Shipping tables are normally constructed based on the distance ("zone"), along with the type of service (e.g., "overnight") and weight ("rate"). In practical terms, two sets of tables are needed from each shipping company.

Attention!

Some of the information in this section is outdated; we will publish new information as soon as possible. See Known Documentation Bugs in *Release Notes for Zelerate AllCommerce 1.2.2*.

`import_shipping_tables.pl` is designed to take information from UPS, USPS, and FedEx Websites in order to construct the required shipping tables that will be put into the database.

For UPS:

- Go to www.ups.com/using/software/currentrates/rates_in_us.html.
- On the UPS Website, for zone information input the first three digits of the "Origin Postal Code" (i.e., where you will be shipping from) and press the download button.
- You will be prompted to download a file that is named by the three digit zip, suffixed with the extension `.csv` (e.g., `944.csv`). This is your zone information, specific to your shipping location.
- After getting the zone information, download the "Download All Rate Charts" via ZIP format.

For USPS:

- Go to postcalc.usps.gov/Zonecharts. Enter the first three digits of the origin postal code, and press the "Get Zone Chart" button. You will need to save this Web page.
- Use the "Save as" button from the browser to save the file "`default.asp`".
- For the rate chart, download the Express Mail and Priority Mail files (`em.txt` and `pm.txt`) from www.framed.usps.com/consumer/csv.htm (note that the extension is `.com`, rather than `.gov`).

For FedEx:

- Go to www.fedex.com/us/rates/zonelocator. Enter the 5-digit ZIP code and click on the "search by zip" button.
- You will then be prompted to download a PDF-formatted file. Download the file.
- Open the PDF format with your PDF viewer (e.g., Adobe Acroread (TM)).
- Click on the text button on the view and cut and paste the first table containing the continental US information (zipcodes and zones only - do not cut and paste header info) to a file called `fedexfile1.txt`.
- Cut and paste the other PDF table (containing Alaska and Hawaii ZIP codes and zones) to a file called `fedexfile2.txt` (remember to cut and paste only the ZIP codes and zones - no header info is needed).
- Download the rates by service at www.fedex.com/us/rates/services/download.html. Use the unix format to download `Rates_by_Service.tar.gz`.

Using `import_shipping_tables.pl`

If you are using all three shipping companies, then you should have the following files:

- UPS -> xxx.csv (where xxx is the three digit post zip) usrates.zip
- USPS -> default.asp em.txt pm.txt
- FedEx -> regionlocator.cgi Rates_by_Service.tar.gz

In the tools directory, create a directory called "`data`", if it is not already there. Place the files you will be using into the `data` directory. For `.tar.gz`, uncompress and untar the file and unzip the `.zip` files, as well.

In the tools directory, run `import_shipping_tables.pl`, which should convert the files above into SQL insert statements for import into your database. The SQL insert statements are grouped into files per the shipping method and either zone or rate (e.g., `UPS.rate.dbid` or `UPS.994.zone.dbid`).

Once these files are created, you will need to load them into your database. They will then be available as shipping options during customer checkout.

Successful Manual Installation

After manual installation has been successfully completed, you should be able to proceed to login to the AllCommerce admin screen and start creating your site.

```
https://sitename.sitedomain/cgi-bin/om/admin/login.cgi

username: admin
password: allcommerce
```

For more information on how to use the AllCommerce managers to build your site, see the *Zelerate AllCommerce User's Guide.*

Loading YOUR Data

AllCommerce provides a Content and an Inventory Manager to load data into the system.

Loading Images

AllCommerce stores images in the filesystem in the `os_allcommerce/images` directory. On most systems this directory can be mounted, which allows for a drag and drop interface. If you are running AllCommerce on a single server, images can be transferred through your systems secure copy function.

As items are created using the AllCommerce Content Manager, unique object identifiers are assigned. These unique identifiers should be used in naming the assets in the images directory.

Tax Tables

There are no tax tables used in this release of AllCommerce. The tax information is handled in the configuration file `site.conf`.

General System Security

AllCommerce is fully capable of taking advantage of the secure encryption schemes offered by secure Web servers.

You can offer secure online ordering and processing of data by utilizing an SSL-enabled web server such as IBM httpd or Stronghold.

Blowfish

AllCommerce utilizes the Blowfish algorithm for password, gift certificate, and other internal data encryption. This 64-bit, variable-length key designed by Bruce Schneier is gaining wide acceptance as a very strong encryption algorithm. It is unlicensed and freely available.

For more information on Blowfish visit: www.counterpane.com/blowfish.html

5. Troubleshooting Your Installation

After you complete your installation, you should reboot your system and start AllCommerce. See Common Installation Problems for tips.

Setting Up Templates

The AllCommerce system administrator uses administrative templates to define, for example, page names. These are the templates used for scripts that are mandatory for all sites generated by AllCommerce.

Mandatory Administrative Templates

The administrative and mandatory templates are in the `os_allcommerce` directory. These templates are divided into page templates that store page information and row templates which are used for repetitive elements.

```
bits/eng/admin/pages
bits/eng/admin/rows
```

The administrative templates are divided into page templates that store page information and row templates which are used for repetitive elements.

Scripts that ***must*** have the administrative templates are:

- welcome.cgi
- hint.cgi
- cart.cgi
- search.cgi
- registry.cgi
- wishlist.cgi

These scripts also require some cell files.

The table below lists the scripts and the page and cell templates they require, as well as links to some HTML samples included in Appendix D: Templates and Cell File Examples.

Script Name	Templates for Pages	Templates for Cell Files	What the Templates Do	HTML Sample
welcome.cgi	page_welcome_login.txt page_welcome_onwards.txt		Required for new or returning customer login. Displays login confirmation for returning customers and provides continuation for store link.	HTML sample HTML sample
hint.cgi	page_hint.txt		Displays a customer's password hint if they call hint.cgi from welcome.cgi.	HTML Sample
cart.cgi	page_shopping_cart_v1.txt	 cell_cart_300_header.txt cell_cart_300_list.txt	Displays a customer's current shopping cart contents, and provides a link to checkout.cgi, or back to shopping. Inserted in place of __newrow_cell_cart_300_header__ and __cell_cart_300_header__. Provides the titles for the columns on the shopping cart contents page. Replaces __newrow_cell_cart_300_list__ and __cell_cart_300_list__.	HTML Sample HTML Sample HTML Sample
search.cgi	page_search_123.txt		search.cgi displays its results on this template. You can use any template you wish, as long as you include a cell reference within it.	HTML Sample
		cell_215_80_int.txt	This is the cell that search.cgi will look for to replace any __column_215_n_ references it finds within page_search_123.txt.	HTML Sample

Script Name	Templates for Pages	Templates for Cell Files	What the Templates Do	HTML Sample
registry.cgi	page_registry _login.txt page_registry _config.txt page_registry _prod.txt page_registry _search_result .txt page_registry _cart_v1.txt	cell_registry_config .txt / cell_registry _complete.txt cell_registry_search .txt cell_registry_search _ head.txt cell_registry_search _list.txt cell_registry_button s.txt cell_registry_button s _pur.txt cell_registry_cart_li st.txt cell_registry _cart_list _pur.txt cell_registry_header .txt cell_registry_header _pur.txt		
wishlist.cgi	page_wishlist _notloggedin.txt			

Script Name	Templates for Pages	Templates for Cell Files	What the Templates Do	HTML Sample
registry.cgi	page_registry_login.txt page_registry_config.txt page_registry_prod.txt page_registry_search_result.txt page_registry_cart_v1.txt	cell_registry_config.txt / cell_registry_complete.txt cell_registry_search.txt cell_registry_search_head.txt cell_registry_search_list.txt cell_registry_buttons.txt cell_registry_buttons_pur.txt cell_registry_cart_list.txt cell_registry_cart_list_pur.txt cell_registry_header.txt cell_registry_header_pur.txt		
	page_wishlist_v1.txt	cell_wishlist_300_buttons.txt cell_wishlist_300_header.txt		

Script Name	Templates for Pages	Templates for Cell Files	What the Templates Do	HTML Sample
registry.cgi	page_registry_login.txt page_registry_config.txt page_registry_prod.txt page_registry_search_result.txt page_registry_cart_v1.txt	cell_registry_config.txt / cell_registry_complete.txt cell_registry_search.txt cell_registry_search_head.txt cell_registry_search_list.txt cell_registry_buttons.txt cell_registry_buttons_pur.txt cell_registry_cart_list.txt cell_registry_cart_list_pur.txt cell_registry_header.txt cell_registry_header_pur.txt		
		cell_wishlist_300_list.txt		

The `crontab` and Recurring Processes

There are several recurring processes that need to be automated. Zelerate recommends that you make these entries in the `crontab` for your web server to make the scripts below run at regular intervals.

Script	What It Does
`generate_site.pl`	`generate_site.pl` produces all the output pages for a site. This script places a fresh set of pages in the `html/eng/auto/` directory, and then calls a script named `update_symlinks` that creates references to these pages in the `html/eng/` directory. If you have manual pages in the `html/eng/man/` directory and wish to use these pages in your site, you must create pointers to `html/eng` s `update_symlinks` links the generated pages. This procedure allows manually created pages to override automatically generated pages. To automate `generate_site` and have it run every hour, place the following entry in the `crontab` for your Web server: `{ time to run } /home/httpd/os_allcommerce/crons/generate_site.pl`
`revert_carts`	`revert_carts.pl` needs to be automated to move reserved inventory out of the shopping cart and place the items back in stock. This script looks for users who have entered checkout, but have not checked out within 15 minutes. To automate `revert_carts` and have it run every 15 minutes, make the following entry in the `crontab` for your Web server: `{ time to run } /home/httpd/os_allcommerce/crons/revert_carts.pl`
`run_stats`	`run_stats.pl` produces statistic reports which are viewed using the AlLCommerce Statistics manager. To automate `run_stats` and have it run everyday at a specific time, make the following entry in the crontab for your Web server:

	{ time to run } /home/httpd/os_allcommerce/crons/run_stats.pl crons

Setting Up Multiple Stores

You can install multiple copies of the entire Zelerate AllCommerce system, one for each storefront you are hosting, to create multiple stores. You can also use a single installation of Zelerate AllCommerce to create multiple stores running on the same server. The advantage to using a single installation is that you use only one set of scripts, which simplifies the process of modifying or changing the code.

About the Scripts

The key to implementing multiple stores is to have the Zelerate AllCommerce scripts locate the correct site.pm file for the particular store for which a given script is executing. This allows each script to configure itself for that particular store so that it will connect to the correct database and read files, templates, etc. from the proper directories.

All Zelerate AllCommerce scripts exist under: cgi-bin/om/admin cgi-bin/om/ssl crons/ tools/

Since the goal is to run multiple stores from one single set of Zelerate Core Commerce scripts, you need to reference these scripts from directories built for other stores. It is imperative that you establish links to the scripts in the manner specified here in order for things to work properly.

Performing the Steps

1. Copy your Zelerate AllCommerce installation to a new directory for each new store you intend to implement. For example:

```
cp -R os_allcommerce new_store_name
```

2. Change directory (cd) to `new_store_name` and remove `cgi-bin`.
3. Recreate this directory as a symbolic link to the `cgi-bin` directory under `os_allcommerce{version}`.

```
Example:
cd new_store_name/ rm -rf cgi-bin ln -sf
../os_allcommerce{version}/cgi-bin/ cgi-bin
```

4. Do this for the `tools/` directory as well.
5. For the `crons/` directory, instead of removing and then linking the entire directory, you must go into the directory itself and remove each script file there.
6. Next, from still within the `crons/` directory, make symbolic links back to each script that exists in `os_allcommerce{version}/crons`.

The directories are linked differently to allow the scripts to find the site.pm file for the store executing the script. Each Perl script has access to the automatic variable $0 which, under ideal conditions, contains the path from which the script was called, along with the actual script name. The Zelerate AllCommerce scripts examine this variable in order to determine which directory they were called from. In this way they are able to find the site.pm file for the particular site for which they were called.

Executing the Scripts

All scripts under the cgi-bin directory are called by the webserver, which will execute these scripts based on the Script Alias path given for each virtual host. This path is contained in $0 when the script is executed. However, scripts in the crons/ directory are not generally called by the webserver, but rather as a cron job or by a user who wishes to execute them immediately after making a change to their site.

In the case where generate_site.pl is called from a sites crons directory, $0 would not contain the stores root directory; rather it would simply contain the name of the script itself. This would not allow for locating the correct site.pm file. However, when the script itself, not the entire directory, is a symbolic link, then the call to `pwd` will contain the path from the root directory through the symbolic link. This way scripts in crons/ can determine the installation directory for the particular store that is using them.

Tips

Remember that when adding to your httpd.conf file for a new store that the path to scripts needs to be something like:

```
ScriptAlias /cgi-bin/ /home/httpd/store_x/cgi-bin/
```

not the path to the `os_allcommerce{version}/cgi-bin` directory.

You might find it handy to save the first directory that you create as described above as a sort of "template" from which to copy for additional sites as you create them. Thus saving all the deleting and linking each time. If you do this you can then copy the template directory with:

```
cp -Rd store_template_directory new_store
```

The "d" argument in the example above will preserve the symbolic links; otherwise, cp will make real files in the new directory.

You could, for purposes of saving space, also make the `admin/` directory under `bits/eng/` a link as all the Zelerate AllCommerce scripts use the files in the admin subdirectories.

About Multiple Stores and Databases

Multiple stores can use single or multiple databases as needed. At this time, Zelerate AllCommerce supports database schemas, not partitions.

Directory Setup for Multiple Stores

For each multiple store, you need to have a separate set of AllCommerce tables; you can, however, use a single instance of the AllCommerce code base.

The following directories need to exist within the `os_allcommerce/` directory:

```
cgi-bin/
crons/
tools/
lib/
crons/
```

Setting Up Store 1

`store1/` uses store1 database tables

The directories below are links to the `os_allcommerce` directory:

`cgi-bin/`	→	`./os_allcommerce`
`crons/`	→	`./os_allcommerce`
`tools/`	→	`./os_allcommerce`
`lib/`	→	`./os_allcommerce`
`crons/`	→	all other files in this directory are linked individually; for example: →`./os_allcommerce/generate_site.pl`

`conf/`	a real directory
`images/`	a real directory
`bits/`	a real directory
`html/`	a real directory
`data/`	a real directory
`var/`	a real directory

Setting Up Store 2

`store2/` uses store2 database tables

The directories below are links to the `os_allcommerce` directory:

`cgi-bin/` → `./os_allcommerce`
`crons/` → `./os_allcommerce`
`tools/` → `./os_allcommerce`
`lib/` → `./os_allcommerce`
`crons/` → all other files in this directory are linked individually; for example:
→ `./os_allcommerce/generate_site.pl`

`conf/` a real directory
`images/` a real directory
`bits/` a real directory
`html/` a real directory
`data/` a real directory
`var/` a real directory

II. Managing Permissions and Users

The end user can have different permission levels for logging into AllCommerce. The system administrator must enable permission checking for permission levels to take effect.

The system administrator manages users and user permissions through the Permission Set and User options on the Content Manager.

Enabling Permission Checking for a User

Permission checking is enabled during configuration.

AllCommerce ignores permission checking if `override_permissions` was set to YES during configuration. Set `override_permissions` to NO to enable permission checking.

Creating a Permission Set

A permission set is a definition describing a group of permissions for an individual user. To create a permission set, select the Permission Set option on the Content Manager to bring up the Create a Permission Set page.

About Permissions

Accepted permissions are:

- rcmd: read, create, modify, delete (full access)
- x: exclude
- * indicates to everything

Default permissions are:

- r**: read only
- rcmd**
- \t: object class # -A
- \t attributes: title, brief, #

Entering the Permissions

A permission set is named by its Title with the Brief as its description. Enter the permissions as follows in the Permissions field, (Class ObjID, Attribute Name, and Permissions), and delimiting with tabs. Keep the format given below.

Class ObjID List	Attribute Name List	Permissions

*
OR
List of Class `objIDs`
	*
	OR
	List of Attribute Names
		`r`
		OR
		`rc`
		OR
		`rcm`
		OR
		`rcmd`
		OR
		`x`

Possible Permission Configuration Types

The table below summarizes permission assignments.

Class	Attribute	Permissions	Meaning
*	*	r/rc/rm/rcmd/x	Permissions to all attributes and all classes system wide
*	field1, … fieldx	r/rc/rm/rcmd/x	Permissions to attributes field1, … fieldx regardless of what class they belong to.
#-A, … , #-A	*	r/rc/rm/rcmd/x	Permissions to all attributes that belong to any of the classes #-A .. #-A
#-A, … , #-A	field1, … fieldx	r/rc/rm/rcmd/x	Permissions to attributes field1, … fieldx that belong to any of the classes #-A .. #-A
102-A	*	rc	Permitted to create and delete attributes.
102-A	*	rm	Permitted to attach/detach attributes to objects that you have write 'c' or 'm' permissions for.
102-A	field1, … fieldx	rm	Permitted to attach/detach field1, … fieldx attributes to objects that you have write 'c' or 'm' permissions for.

Creating a User

Choose the User option on the Content Manager page to bring up the Create a New User page.

Complete the User Name and Password fields for a new user. The Hint field allows you to type in text that is helpful to the user in remembering the password.

Set Generate Site Pages to Yes or No, depending on the permission level granted to the user.

Enter the permission sets you want to set for the user. You can add the OBJIDs of the permission sets that the user belongs to. Permission sets must be delimited by a colon (:).

If there are special rules for the user, add these additional permissions in the Unique Permissions field.

Attention!

Permission sets must exist before you can assign them to a user. See Creating a Permission Set if necessary.

III. Appendices

This part contains the following appendices:

Appendix A: AllCommerce Directory Structure

This is a brief summary of directory use in AllCommerce.

An "M" before each description flags items that are "Modified in normal use". Note that any item may be unchanged or modified on a given system.

```
>        <PRE>
>      .                    .   top level of distribution
>      ./DOCS               .   AllCommerce developer documentation
>      ./archive            .   save directory (anywhere in tree)
>      ./bits               .   HTML templates and data files
>        <language>         .   e.g., English (eng)
>          admin            .   administrative
>            cells          .   HTML templates for specific cells
>            include        .   HTML templates for specific includes
>            pages          .   HTML templates for specific pages
>            rows           .   HTML templates for specific rows
>            xml            .   XML templates
>          html             .   HTML templates
>            cells          .   HTML templates for specific cells
>            pages          .   HTML templates for specific pages
>            sel            .   menu selections
>          palm             .   PalmPilot-specific interfaces
>            cells          .   HTML templates for specific cells
>            pages          .   HTML templates for specific pages
>          ps               .   PostScript templates
>            cells          .   PostScript templates for specific cells
>            pages          .   PostScript templates for specific pages
>          system           .   data load and text (error, message) files
>          wml              .   WAP-enabled generic devices
>            cells          .   WML templates for specific cells
>            pages          .   WML templates for specific pages
>          xml              .   XML templates
>            cells          .   XML templates for specific cells
>            pages          .   XML templates for specific pages
>      ./cgi-bin            .   CGI scripts
>        om                 .   OpenMerchant (aka AllCommerce) area
>          admin            .   administrative scripts
>          ssl              .   scripts that require encrypted
connections
>      ./conf               .   configuration files
>      ./crons              .   non-interactive scripts
>      ./data               M   client or end-user dump data
>        stats              M   stats manager output files
>          cron             ?   ?
>          include          ?   ?
>      ./html               M   site HTML code
>        <language>         .   e.g., English (eng)
>          DOCS             .   AllCommerce end-user documentation
>          include          .   HTML include files
>            auto           M   automatically generated
>            man            M   manually generated
```

```
>          man             M   manually generated
>       ./images           M   site image (e.g., JPEG) files
>         <language>       .   e.g., English (eng)
>          buttons         M   button icons
>          general         M   ?
>          headers         M   ?
>          home            M   ?
>          os_allcommerce  M   ?
>            admin         .   ?
>              buttons     .   ?
>              general     .   ?
>              icons       .   ?
>              main        .   ?
>              menubars    .   ?
>              spine       .   ?
>          products        M   ?
>            250           M   full-size (250x250)
>            280           M   full-size (280x280)
>            400           M   full-size (400x400)
>            80            M   thumbnail (80x80)
>          spine           M   ?
>          splash          M   ?
>       ./lib              .   general-purpose libraries
>       ./palm             M   site Palm 7 files
>         auto             M   automatically generated
>         DOCS             .   AllCommerce end-user documentation
>         include          .   HTML include files
>          auto            M   automatically generated
>          man             M   manually generated
>         man              M   manually generated
>       ./ps               M   site PostScript code
>       ./tools            .   command-line (e.g., administrative)
scripts
>       ./transit          M   import/export staging area
>         export           .   export data
>          db              M   database-related
>          event           M   event-related
>          log             M   log-related
>          order           M   order-related
>          trigger         M   trigger-related
>         import           .   import data
>          db              M   database-related
>          event           M   event-related
>          log             M   log-related
>          order           M   order-related
 >         trigger         M   trigger-related
>       ./var              M   generated (e.g., lock, log) files
>         lock             M   lock files
>         log              M   log files
>         trigger          M   trigger files
>       ./wml              M   site WML (Wireless Markup Language)
>         auto             M   automatically generated
>         DOCS             .   AllCommerce end-user documentation
>         include          .   HTML include files
>          auto            M   automatically generated
```

```
>         man             M   manually generated
>        man               M   manually generated
>      ./xml               M   site XML (Extensible Markup Language)
>        auto              M   automatically generated
>        DOCS              .   AllCommerce end-user documentation
>        include           .   HTML include files
>          auto            M   automatically generated
>          man             M   manually generated
>        man               M   manually generated
```

Appendix B: AllCommerce Configuration Files

This appendix has details about AllCommerce configuration files in the AllCommerce 1.2 Release.

Configuration File	Description
`database.conf`	Database parameters for the `site.pm` file
`site.conf`	Site parameters for the `site.pm` file
`machine.conf`	Machine parameters for the `site.pm` file

Appendix C: Templates and Cell File Examples

welcome.cgi Templates And Cell Files

Here are minimal examples of the templates and cell files needed by welcome.cgi.

HTML Sample for page_welcome_login.txt

```
<HTML><!-- page_welcome_login.txt -->
 <HEAD>
  <TITLE>Please Sign In to __storename__ </TITLE>
 </HEAD>
 <BODY BGCOLOR="FFdd66" link="#330000" vlink="#333300"
   marginheight="0" marginwidth="0" leftmargin="0"
   topmargin="0">
  <FORM METHOD="POST" ACTION="/cgi-bin/om/__scriptname__">
  <INPUT TYPE="hidden" NAME="pg_targ" VALUE="__pg_targ__">
  <INPUT TYPE="hidden" NAME="sid" value="__sid__"><br>

   <TABLE><TR colspan=4 BGCOLOR="#AAAAAA" WIDTH=300>
    <TD>My site Login</TD></TR>

     __cell_welcome_new_cust_300__

     __cell_welcome_return_cust_300__

   </TABLE>
  </FORM>
 </BODY>
</HTML>
```

HTML Sample for page_welcome_onwards.txt

```
<HTML>
 <HEAD>
  <TITLE>__storename__ Login Success</TITLE>
  </HEAD>
   <!-- page_welcome_onwards.txt -->
  <BODY marginheight="0" marginwidth="0" leftmargin="0"
       topmargin="0"> <BR>
    <TABLE cellspacing="0" cellpadding="0" BORDER=0
         align="center">
     <TR>
      <TD>
       <B>You are now logged in to __storename__  Remember to
         close your browser when you leave your machine
          unattended!</B>
       <P>Account name:<B>__EMAIL__</B>
       <P>Please continue</TD>
     </TR>
```

```
      <TR>
      <TD><A HREF="__pg_targ__">Continue</A></TD>
      </TR>
     </TABLE>
   </BODY>
 </HTML>
```

HTML Sample for page_welcome_confirm.txt

```
<HTML>
 <HEAD>
  <TITLE>__storename__ | Account created</TITLE>
 </HEAD>
   <!-- page_welcome_confirm.txt-->
 <BODY>
  <TABLE cellspacing="0" cellpadding="0"
      BORDER=0 align="center">
    <TR>
     <TD><B>Account created</B>
      <B>Congratulations on setting up your
         new __storename__ account!</B><P>
         You have created an account
         with __storename__ using the following
         information:/TD></TR>
     <TR><TD>  Account name:<B>__EMAIL__</B></TD></TR>
     <TR><TD>Password: <B>__PASSWD__</B></TD>
      <TD>Please write this information down or print
          out this page and keep it in an empty ice cream
          container in your freezer.</TD>
     <TR><TD><A href="/html/__pg_targ__">Continue</A></TD>
    </TR>
  </TABLE>
 </BODY>
</HTML>
```

hint.cgi Templates for Cells

HTML Sample for page_hint.txt

```
<HTML>
 <HEAD>
  <TITLE>__storename__  | Your password hint</TITLE>
 </HEAD>
  <!-- page_hint.txt -->
 <BODY link="#330000" vlink="#333300" marginheight="0"
   marginwidth="0" leftmargin="0" topmargin="0">

  <FORM METHOD="POST" ACTION="/cgi-bin/om/hint.cgi">
   <INPUT TYPE="hidden NAME="pg_targ"
```

```
 VALUE="__pg_targ__"> <input type="hidden"
 name="sid" value="__sid__"> <BR>

<TABLE cellspacing="0"
  cellpadding="0" BORDER=0 align="center">
 <TR>
  <TD><B>Forgot My Password</B></TD>
  <TD>I can't remember my password.
     Please tell me what my hint is?</TD>
 </TR>
 <TR><TD>
  <TABLE BORDER=0 cellspacing="0" cellpadding="0">
    <TR> <TD>__HINT_err__<B>Your email address:</B>
        (user@domain.com)<BR></font></TD>
   </TR>
   <TR> <TD>__EMAIL_err__<BR>
   <INPUT TYPE="text" NAME="EMAIL_inp"
      VALUE="__EMAIL_inp__" SIZE="20"></td>
    <td align=middle><INPUT TYPE=submit value="submit"></TD>
   </TR>
   <TR><TD>
   <a href=/cgi-bin/om/welcome.cgi?pg_targ=__pg_targ__>
        I'm now ready to login.</a></TD>
   </TD> </TR>
  </TABLE>
 </TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

cart.cgi Templates and Cell Files

Here are minimal examples of the templates and cell files needed by cart.cgi.

HTML Sample for page_shopping_cart_v1.txt

```
<HTML>
 <HEAD>
  <TITLE>__storename__ | My shopping cart</TITLE>
 </HEAD>
   <!-- page_shopping_cart_v1.txt PM -->
  <BODY link="#330000" vlink="#333300" marginheight="0"
       marginwidth="0" leftmargin="0" topmargin="0">

  <FORM METHOD=post action=/cgi-bin/om/cart.cgi>
  <input type="hidden" NAME="pg_targ" VALUE="__pg_targ__">
  <input type="hidden" name="sid" value="__sid__">
  <input type="hidden" name="vcode" value="__vcode__">
  <input type="hidden" name="AXIS1_inp" value="__AXIS1_pre__">
  <input type="hidden" name="AXIS1_pre" value="__AXIS1_pre__">
```

```
          <TABLE BORDER=0 WIDTH=300 CELLSPACING=0 CELLPADDING=3>
           <TR>
            <TD collspan=3><FONT face="verdana,geneva,arial,helvetica"
                color="#999933">
             <B>My Cart</B></FONT><BR></TD>
            </TR>

             __newrow_cell_cart_300_header__
             __newrow_cell_cart_300_list__
             __newrow_cell_cart_300_buttons__

            <TR><TD colspan=3>
             <FONT FACE="arial,helvetica,san-serif" SIZE=-1
color="#999933">
              To change a quantity, enter the <b>new quantity</b> and
              press the update button.<BR>To remove an item, select the
              checkbox next to <b>remove</b>and press the update
              button.</font>
             </TD></TR>
            </TABLE>
           <TABLE BORDER=0 WIDTH=300 CELLSPACING=0 CELLPADDING=3>
              __cell_cart_300_header__
              __cell_cart_300_list__
              __cell_cart_300_subtotal__
              __cell_cart_300_buttons__
            </TABLE>
           </FORM>
          </BODY>
         </HTML>
```

HTML Sample for cell_cart_300_header.txt

```
        <!----- Titles of columns ----------------------------------->

        <TR>
        <TD><FONT face="verdana,geneva,arial,helvetica" color="#999933"
size=-1>
                <b>Product</b></font><BR>
        <IMG SRC="/images/general/dot.gif" WIDTH=140 HEIGHT=1></TD>
        <TD><FONT face="verdana,geneva,arial,helvetica" color="#999933"
size=-1>
                <b>Price</b></font><BR>
        <IMG SRC="/images/general/dot.gif" WIDTH=40 HEIGHT=1></TD>
        <TD><FONT face="verdana,geneva,arial,helvetica" color="#999933"
size=-1>
                <b>Qty.</b></font><BR>
        <IMG SRC="/images/general/dot.gif" WIDTH=40 HEIGHT=1></TD>
        <TD ALIGN=MIDDLE><FONT face="verdana,geneva,arial,helvetica"
color="#999933" size=-1>
        <b>Remove</b></font><BR>
        <IMG SRC="/images/general/dot.gif" WIDTH=60 HEIGHT=1></TD>
        </TR>
```

HTML Sample for cell_cart_300_list.txt

```
<TR>
<TD><A HREF="/html/__VSKU__.shtml"><FONT FACE="arial,helvetica,
       san-serif" SIZE=-1>__TITLE__
 </FONT></A><BR>
<FONT FACE="arial,helvetica,san-serif" SIZE=-2 color="#FF0000">
__QUANT_err__</FONT>
<FONT FACE="arial,helvetica,san-serif" SIZE=-2>
Sku number:__VSKU__</FONT></TD>
<TD>$__UNITPR__</TD>
<TD ALIGN=MIDDLE>
<INPUT TYPE="text" NAME="row.__ROWID__.QUANT_inp" SIZE="3"
 VALUE="__QUANT__">
<INPUT TYPE="hidden" NAME="row.__ROWID__.QUANT_pre" SIZE="3"
 VALUE="__QUANT__">
<INPUT TYPE="hidden" NAME="row.__ROWID__.SKU" SIZE="3"
 VALUE="__SKU__">
<INPUT TYPE="hidden" NAME="row.__ROWID__.VCODE" SIZE="3"
 VALUE="__VCODE__">
<input type="hidden" name="row.__ROWID__.VSKU" SIZE="3"
 VALUE="__VSKU__">
</TD>

<!----- Shopping List Checkbox -------------------------->

<TD ALIGN=MIDDLE><input type=checkbox name=row.__ROWID__.kill
   value="on"></TD>
</TR>
```

The first bit of data, the hyperlink, provides a link back to the product page. __QUANT_err__ is replaced with an error message if there is one, this would happen if a customer attempted to put more of one item in their cart than is allowed per the store_limit_all or store_limit_jit variables in store.pm. __VSKU__ is replaced with the product's sku with a variant code appended if the product has variants (see Product Variants for details). __VCODE__ is for the variant code itself.

search.cgi Templates

HTML Sample for page_search_123.txt

The only template required for search.cgi, page_search_123.txt is fairly simple. It just has to be there so search.cgi has some template on which to display its results. You can use any template you wish to design therefore, as long as you include a cell reference within it. So somewhere in your page_search_123.txt you will need:

__column_215_1__

This should be within a table, and increment the last digit (# 1) here, for each column you add.

HTML Example for cell_215_80_int.txt

Here is an example from the sample store "presents".

```
        <TR>
         <TD><TABLE WIDTH=215 CELLSPACING=0 CELLPADDING=0 BORDER=0>
        <TR>
         <TD WIDTH=80 VALIGN=TOP>
         <A HREF="/html/__SKU__.shtml"><IMG SRC="/images/products/80/
          __SKU___80.jpg" WIDTH=80 HEIGHT=80 BORDER=0
ALT="__TITLE__"></A>
         </TD>
         <TD WIDTH=10><IMG SRC="/images/general/dot.gif" WIDTH=10
            HEIGHT=1></TD>
         <TD WIDTH=125 VALIGN="top">
         <FONT FACE="arial,helvetica,san-serif">
          <A HREF="/html/__SKU__.shtml"><B>__TITLE__</B></A></FONT>
         <BR>
         <FONT FACE="arial,helvetica,san-serif" SIZE=-1>
                __BRIEF__</FONT></TD>
        </TR>

        <TR>
         <TD COLSPAN=3><IMG SRC="/images/general/dot.gif"
             WIDTH=215 HEIGHT=10></TD>
        </TR>
        </TABLE></TD>
        </TR>
```

This cell includes a reference to the products image, but this is not necessary.

Appendix D: Credit Card Verification System (CCVS)

The level of implementation of the CCVS is experimental. In future, AllCommerce will provide and API for this system. Look for updated information at http://www.zelerate.org/.

Installation

a) Install CCVS as user "nobody." Blowfish algorithm can be used for encryption/decryption here.
b) Uncomment functions and libraries that call `run_ccvs()`
c) Edit site.conf to make payprocess=ccvs
d) Handle error conditions (e.g., socket error).

Specific Information

e) Install CCVS on a machine different from the machine on which os_allcommerce is installed.
f) Run a daemon (e.g., `os_to_ccvs.pl`) on the same machine as CCVS. All the daemon does is receive sensitive information required for authorization from `os_allcommerce`, communicate with CCVS and return back the authorization status.
g) Edit `cluster.conf` to add the machine on which CCVS is installed and port on which the daemon is running.

If you install CCVS on the same machine as os_allcommerce, edit ccvs.pm to talk to CCVS on the same machine.

Appendix E: Troubleshooting

This appendix gives you tips on how to handle some of the most frequently encountered installation problems. See **www.zelerate.org** for documentation updates.

Common Installation Problems

Most installation problems have to do with improper configuration of `store.pm` and `httpd.conf`.

- Be sure that the host and domain values set in `store.pm` coincide with what is listed in `/etc/hosts` and your VirtualHost declaration in `httpd.conf`. If you used `www.presents.com` as an example for the sample store installation, then `storedomain` should be `.presents.com` and `storeserver` should be `www`.
- Make sure that your VirtualHost declaration has `www.presents.com` in the beginning of the declaration:
  ```
  <VirtualHost www.presents.com>
  ```
- Make sure that ServerName is also `www.presents.com` within your VirtualHost declaration.
- For `/etc/hosts`, if you are just testing on a local installation, you need to put "www.presents.com" on the same line as your `localhost.localdomain` name. For example:
  ```
  127.0.0.1 localhost.localdomain www.presents.com
  ```

Cannot Access Your Web Store

This common problem occurs often with people who have more than one `httpd.conf` file on their computer. This could happen if Apache was already installed unknown to the user who installs it again via a download from `www.apache.com`. If you are certain you have all the values set and everything configured correctly, but you are still getting odd *denial of access* errors when you point your browser to your newly installed store, try doing a `locate` for `httpd.conf`:

```
locate httpd.conf
```

If you get a list of more than one, possibly in `/etc/httpd/conf` and `/usr/local/apache/conf`, then you need to find out which one your web server is reading when it starts up.

One way to find out which `httpd.conf` file your web server is reading is to stop your web server, change the name of both `httpd.conf` files, and then restart Apache. This will show you which file it was trying to read when it fails to start and gives error messages about not finding `httpd.conf`.

Restarting the Server After Editing `httpd.conf`

This may seem trivial, but it seems to happen often and can be the cause of much frustration and wasted time. Simply using the `apachtl restart` osr `httpd stop, httpd start` commands may not really stop all processes. These steps will ensure that your changes to `httpd.conf` have taken effect:

Make sure that all processes have stopped by using the `kill` command:

```
killall httpd.conf
```

Next do a : `pstree` before restarting the server.

Error Logs

Often, web server problems can be ferreted out by examining the error logs that are generated by the server. As an example, Apache's error logs can usually be found in `/usr/local/apache/logs`. Within this directory are logs for the web server itself; these usually list errors in starting the server or in its configuration. You can also find logs specific to each host (virtual if more than one) that your web server handles. These are usually more informative for finding AllCommerce installation or configuration errors.

Where to Get Help

You can join the mailing lists at http://lists.zelerate.org/mailman/listinfo. The `devel` (development) and `docs` (documentation) lists are best for asking questions. You can post a problem or question and it will usually be answered by another user or by a member of the development or documentation teams.

Database Problems

Most postinstallation database problems have to do with improper configuration of the values in `database.conf`. Such errors can usually be discovered by examining the server logs for your site. The most common errors involve a bad name given for the `site_server`, but all of the `database.conf` values are equally important to allow AllCommerce scripts to successfully interact with your database.

A Reminder

Remember to check www.zelerate.org regularly for FAQs, documentation updates, and latest support information!