

# **Zelerate AllCommerce User's Guide**

*Release 1.2 Beta*

## Preface

The *Zelerate AllCommerce User's Guide* gives you detailed information about using Zelerate AllCommerce effectively to generate your ecommerce web sites.

## Audience

*The Zelerate AllCommerce User's Guide* provides complete information for the person who generates ecommerce web sites and enters information about products, product lines, brands, and other related items. This person may be someone who handles inventory, or is in merchandising or customer service.

This document assumes that you have a basic understanding of an operating system such as Windows, are familiar with graphical user interfaces to enter and manipulate data, and require no information about Internet commerce. In addition, using some of the advanced AllCommerce features requires technical understanding of Perl, HTML, and cgi scripts, object-oriented technology, and familiarity with web site design.

## How This Document is Organized

This document has the following major parts, which are further divided into sections covering major topics:

- [Part I: Understanding Zelerate AllCommerce](#) gives an overview of how the product works.
- [Part II: Using the Content Manager to Build Your Web Site](#) describes how to build the elements that make up your web site.
- [Part III: Generating Reports for User Activity](#) shows you how to run standard reports that show statistics about the shopper at your web site.
- [Part IV: Advanced Features](#) discusses adding enhancements to your web site such as a shopping cart.
- [Part V: Customizing the Installation](#) is useful if you are adding or modifying your basic AllCommerce setup and functionality.
- [Part VI: Appendices](#) contains several sections with lookup information, and a Developer's Corner that discusses Open Source development.

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.

## Printing this Document

This document is created for online viewing. If you print this document, you may find that the formatting changes from the way it looks on your browser. Using landscape format may help with tables. Zelerate recommends that you use either Mozilla or Netscape for printing.

## 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.	<i>Warning!</i>
Attention notes	Important information that could save you time or something you need to remember for the task at hand.	<i>Attention!</i>

## 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 System Administrator's Guide](#).

### External Information Sources

Other sources of information are:

[www.opensource.org](http://www.opensource.org)

### Zelerate Training

Zelerate offers comprehensive training to help you master Zelerate AllCommerce and use it effectively. See [www.zelerate.com](http://www.zelerate.com) to find out 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](mailto: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 that information is missing or incorrect, log a documentation bug at [bugs.zelerate.org](http://bugs.zelerate.org).

# Table of Contents

<b><u>I. Understanding Zelerate AllCommerce</u></b>	<b><u>I-8</u></b>
Overview	I-8
How Information is Stored in Zelerate AllCommerce	I-9
Special Purpose Tables .....	I-9
General Purpose Tables .....	I-9
Things table	I-9
Extraprop table	I-9
Linkage table	I-9
About Multiple Languages.....	I-10
Your First Contact with Zelerate AllCommerce	I-10
Basic Guidelines about Entering Information	I-11
Common Entry Fields .....	I-11
Points to Remember when Entering Information .....	I-12
<b><u>II. Using the Content Manager to Build Your Web Site</u></b>	<b><u>II-13</u></b>
Creating a Brand	II-13
About Brands .....	II-13
About an Object ID.....	II-13
Performing the Steps	II-14
Creating a Variant	II-15
About Variants and Variant Sets.....	II-15
Performing the Steps	II-15
Variant Naming Conventions .....	II-15
Creating a Variant Set	II-17
Performing the Steps	II-17
Creating a Product Line	II-17
Brand vs. Product Line .....	II-17
Performing the Steps	II-17
Creating a Product	II-19
Performing the Steps	II-19

Creating an Article	II-21
Creating an Internal Document	II-22
Creating a Pulldown	II-23
About Pulldowns.....	II-23
Performing the Steps	II-23
Variant Pulldowns.....	II-24
Performing the Steps	II-25
Creating a Warehouse	II-26
Working with Attributes	II-26
Creating an Attribute.....	II-26
Handling Multiple Languages	II-27
Associating an Attribute with All Products .....	II-27
Defining the Display Position for an Attribute .....	II-29
Associating an Attribute with a Specific Object.....	II-29
Adding Custom Fields	II-30
Defining Display Positions for Objects	II-30
Display Definition Format .....	II-30
Deferring Options Till Runtime	II-31
Displaying Multiple Rows	II-31
<b>III. <u>Managing Orders</u></b>	<b>II-33</b>
Editing Orders.....	II-33
Customers .....	II-34
Editing Objects	II-34
Searching for an Object to Edit	II-34
Exporting Order Information to XML Format	II-35
Order	II-35
Addresses .....	II-35
Contacts	II-35
ocart	II-35
Things	II-35
Exported File.....	II-36
Exported File Format .....	II-36

xml_page_order_format.txt	II-36
xml_page_contacts.txt	II-36
xml_page_order.txt	II-37
xml_page_shipto.txt	II-37
xml_page_billto.txt	II-37
xml_page_ocart.txt	II-37
<b><u>IV. Managing Inventory</u></b>	<b>III-38</b>
Adding Inventory	III-38
Adding Prices	III-39
<b><u>V. Controlling Merchandise Display</u></b>	<b>IV-41</b>
<b><u>VI. Generating Reports for User Activity</u></b>	<b>V-43</b>
About the Statistics Manager	V-43
Reporting Screens	V-43
Primary Report.....	V-43
Snapshot View	V-43
Detailed Stats .....	V-44
General Statistics .....	V-44
Report	V-45
Reporting Order Information .....	V-46
Previous Stats.....	V-46
Limitation.....	V-47
<b><u>VII. Using Advanced Features</u></b>	<b>VI-48</b>
The Shopping Cart	VI-48
Adding Shopping Carts to Your Web Site.....	VI-48
Quick Start .....	VI-48
How Shopping Carts are Created.....	VI-48
Templates and Form Elements for cart.cgi .....	VI-50
Adding Checkout to Your Web Site	VI-51
How Checkout Works.....	VI-51
<b><u>VIII. Customizing the Installed Site</u></b>	<b>VII-52</b>
Working with Templates	VII-52

Site Templates	VII-52
Intermediate and Article Pages .....	VII-52
Intermediate Catalog Page	VII-54
Product Pages .....	VII-56
A Brief Example of Template and Cell File Use .....	VII-56
<b>IX.   <u>Appendices</u></b>	<b>VIII-59</b>
Appendix A: AllCommerce cgi Scripts	VIII-60
Appendix B: Other Scripts	VIII-61
Appendix C: AllCommerce Functions	VIII-62
Appendix D: Implementing Cookieless Shopping	VIII-63
Functions	VIII-63
Requirements .....	VIII-63
General Guidelines for Testing .....	VIII-63
Limitation .....	VIII-63
Appendix E: Developers' Corner	VIII-65
The Zelerate Development Community .....	VIII-65
About Open Source Development .....	VIII-65
How to Get Involved .....	VIII-66

# I. Understanding Zelerate AllCommerce

To be able to utilize the flexibility Zelerate AllCommerce gives you, you need to understand how the system works and how to perform the basic tasks necessary to generate your web site.

## Overview

Zelerate AllCommerce consists of the following main functional areas. Each of these areas is delineated by a tab option on the main page that appears when you first log in to AllCommerce as an administrator. You can select the appropriate tab to perform the tasks within each functional area.

All these functional areas utilize a search script to look for information in the database.

Functional Areas	What You Can Do Here
<a href="#">Content Manager</a>	This is where you create the various assets that you use in your web site, such as products and related information such as product lines, variants, and variant sets.
<a href="#">Link Manager</a>	Here you establish relationships between all products on a page on your web site.
<a href="#">Inventory Manager</a>	Here you can add or modify inventory in your warehouse. Inventory is automatically decremented as purchases are made from your web site; you can also modify inventory manually.
<a href="#">Order Manager</a>	This screen gives you access to orders that have been placed through AllCommerce on your web site. You can review existing orders and modify them.
<a href="#">Statistics Manager</a>	This is the AllCommerce report-writing tool.
<a href="#">Warehouse Manager</a>	Here you can maintain warehouse information. The current release of AllCommerce supports one warehouse.



## How Information is Stored in Zelerate AllCommerce

Everything you create within AllCommerce has a unique *object ID*. The `objID` is the primary reference key for all objects. A few other fields associated with objects like products and brands are: `objtype`, `objclass`, `title`, and `brief`.

Virtually all of the administration information is stored in the tables within the database. There are specific tables for some parameters, but much of the setup information is stored in general purpose tables such `things` and `extraprop`.

AllCommerce also has some special purpose tables. Depending on feedback from customers, the number of special purpose tables is likely to decrease as AllCommerce evolves.

### Special Purpose Tables

Special purpose tables encode system states for “prices,” “inventory,” “zonemap,” and “zonerate,” the first two of which are self-explanatory. The two zone files encode mappings for calculating interzone shipping charges. “zonemap” encodes the zones for each shipping method while “zonerate” encodes the shipper's pricing for each method.

### General Purpose Tables

Apart from the special purpose tables, the balance of the state data is stored in general-purpose tables which encode virtually everything else in the system.

#### Things table

The `things` table is used to encode every product, Web page, category, shipping method, and service used within AllCommerce. Note that `sku_skel` and `sku_series` are not stored in the `things` table, but rather in the `extraprop` table.

#### Extraprop table

The `extraprop` table contains name/value pairs which supplement the attributes encoded within the `things` table, such as localization-specific fields.

#### Linkage table

The “linkage” table is the glue that sticks everything together. It defines relationships between “things” and specifically is responsible for encoding the entire navigational Web. An item may appear multiple times within the hierarchy of links by merely adding another link.

## About Multiple Languages

The current release of AllCommerce requires you to manually perform symlinks to utilize multiple language functionality. This functionality enables you to add an object in one language, then change languages and set the language-specific information to be appropriate for the different language.

These symlinks are:

```
bits/fra
ln -fs ../<lang>/admin admin
ln -fs ../<lang>/admin html
ln -fs ../<lang>/system system
```

Anything that is language-sensitive will appear in blue when you are in another language. Information that is not language-sensitive will appear normally in their entry boxes. If you remove language sensitivity, the blue disappears and the change is applied to all languages. In all cases, <lang> is the language that you selected when you logged into the Administrator.

## Your First Contact with Zelerate AllCommerce

At first use, you must log in to the administrator and define your site before you can start creating products and other items for the site. Zelerate recommends that you first define your site as a brand, product, or a product line and select Home Page as the display page style. See the following sections for instructions on how to perform these tasks:

- [Basic Guidelines about Entering Information](#)
- [Creating a Brand](#)
- [Creating a Product](#)
- [Creating a Product Line](#)

## Basic Guidelines about Entering Information

This section discusses some basic points you need to remember while entering information in AllCommerce. These points are relevant for all data entry pages.

### Common Entry Fields

These fields are common to *most* data entry pages; in addition, some pages have additional fields.

Field/Check Box	Where	What Happens Here
<b>Generate Page?</b>	Content Manager  Link Manager  Inventory Manager	Selecting this check box before you Enter Changes on any page regenerates the site display for the object ID that you are modifying.  You must have permission to generate pages. Permissions are assigned by the AllCommerce system administrator at your company.
<b>Title *</b>	Content Manager	Enter the name of the object (e.g., page or product) that you are creating. It can be from 3 to 80 characters in length. Reference in an HTML template with "__title__".
<b>Brief *</b>	Content Manager	A short (5 to 250 character) description of the object you create. Reference in an HTML template with "__brief__".
<b>Description</b>	Content Manager	Description is a larger text area than Brief for describing the object. It does not have a character limit. Description would be better named as Body or Contents since this is a large text entry area that could be used for the main body of the product. Reference the object description in a page template with "__descript__".
<b>Keywords</b>	Content Manager	Enter one or more words that you can easily associate with the new object. This will allow you to perform a keyword search for this object when you need to reference it again for other administration functions.
<b>Ranking</b>	Content Manager	Here you can give priority to products that are displayed on a page that shows a number of products, such as a search results page. The greater the value for the product, the closer to the top of the list it will appear. Equal values for multiple objects are resolved randomly.
<b>Status</b>	All	Status is where you define whether the object is currently active or inactive.
<b>Display on Site</b>	All	This field offers two choices—Display or Non-Display. Non-Display items are not generated and are not included in the search function. Display items are, of course, displayed on your web site.

<b>Page Style</b>	Content Manager	This field offers you several choices through a list and enables you to define the display, which is based on the type of page template you want for your web site. The Display on Site field must be set to Display for Page Style to be effective. For more information on templates, see <a href="#">Working with Templates</a> .
-------------------	-----------------	--

## Points to Remember when Entering Information

- Data fields that can be referenced from a page template are marked with an asterisk. While this document describes use in HTML templates, any markup language for which you are generating content can map these values. Most of these text fields are designed as handy data fields for whatever use you wish to make of them. This is particularly useful for objects that will exist as displayable entities.
- There is no way to delete an object within AllCommerce. To remove an object from your web site, set it to `Inactive`.
- If you enter more than one word during a search, those multiple words must appear in the `title`, `brief`, or `keyword` fields in the object definition. For example, if you enter "the car" in the search field, you must have some text that looks like "the car" in the above-mentioned fields. In other words, "the car" would match "the car" and "the cart", but it will **not** match "car the" or "then car". The same words must appear in the same order as entered.
- For an internal document, be **sure** to set Display on Site to "Non-Display." Otherwise, the document will be displayed on your site!
- When doing searches, note that the title is not searched for variant products. If you type in "Pens - red", and you have a master product "Pens" and a variant product for "Pens - red", it will not match. However, if you do type in "Pens" as your search string, AllCommerce will show the master product, from where you can navigate to the variant products. If you type "Pens", it will show "Pens" and "Pens - red". If you type "red", it will show "Pens - red". However, if you type "red pens" or "Pens - red", it will not match. Try to use one word searches wherever possible.

Other values not included in the search function are alternate title.

- Non-Display items are not generated by [generate\\_site.pl](#), and are not searched for in [search.cgi](#).
- You cannot define an alternate title for variant products.
- If you enter invalid data in enterable fields, they are reset to the default values (or left blank); AllCommerce displays relevant error messages under the field with the incorrect data.

## II. Using the Content Manager to Build Your Web Site

The Content Manager is your gateway to various tools you can use to build your web site. This section discusses the tools and tells you how to manage products and related information such as product lines, variants, and brands and use them in your web site.

To access the Content Manager, select the Content Manager tab on the main AllCommerce page. The Content Manager offers several choices that allow you to access different pages to create content and objects.

You can create things in an order different from the one described here. You will probably create brands first, then variants if you need them, and finally product lines or actual products. These functionalities are described in this usage sequence, since you will need to create a brand before you can create a product that is associated with it, and a Variant Set before a product has variants.

### Creating a Brand

You need to create brands before you can begin creating other objects, such as products, because you need to associate a product with an existing brand when you create it. Select the Brands option in the Content Manager to bring up the Create a New Brand page.

#### About Brands

Brands are simply ways of grouping product objects with a common attribute value (the brand). Internally, each brand will have a number of attribute fields associated with it. Most of these attributes are stored in the `things` table.

Creating a brand is similar to creating a product or web page, although you may notice that there is less information that you need to enter. You can also decide whether or not to have a brand be a displayable object, with or without any number of its associated products. When creating a brand, AllCommerce assigns it an "objID" (object ID). This value is generated by incrementing the highest existing objID value that exists within the system.

#### About an Object ID

If you are creating a brand for the first time, and it is also the first object you have ever created in AllCommerce, the brand would have an objID of 1-AA.

The AA portion of the id is called the series. It can also increment when the numeric portion of the objID reaches a maximum value. Typically, this would be the largest integer value that is allowed in your database before it "rolls over". The next series value would be "AB".

The `objtype` attribute field for brands is 47004-A. This is defined in the initial database load and is a reference to the objID for the brand class definition. A brand's objclass is 136-A, which is a reference to the content type class.

The `sku` field for a brand would be the three-letter code you enter when you create the brand. The `msku` field is followed by the single-character `site_code` value from `site.conf`, followed by `0000`. Something like: "aaax0000", where `aaa` is the three-letter code and `x` is the `site_code`.

The actual brand class exists as an `os_allcommerce` object. This brand class (`os_allcommerce` object) is entered into the database during the installation's initial data load. All brands you create are instantiations of a base class. In fact, anything you create in `os_allcommerce` is an instantiation of a base class.

## Performing the Steps

Enter relevant information for the Generate Page? Brief, Description, Keywords, Status, Ranking, Page Style, and Display on Page fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. Enter information in these additional fields:

SKU Code:

You can choose the first three letters that will make up the new brand's SKU. For example, if you are entering a new brand called Flosten Manufacturing, you could enter "f1o" as the first three letters. Reference in an HTML template with "\_\_sku\_skel\_\_".

If "f1o" is already taken, pick another code (e.g., "f1m"), trying for something that will be memorable.

When you are finished, select Enter Changes to create the new object.

## Creating a Variant

Variants are variable attributes that belong to objects you create; in other words, a variant object expresses the quality of an object, such as its color or size. As an example, consider a pen that is the same in all ways (e.g., manufacturer, style, price), yet it comes in several ink colors, and a variety of tip sizes. It would be tedious to have to create a number of different pen objects, changing only one or two characteristics each time. This is where variants come into the picture. You can create only one pen object, and define a number of variations for the same object with variants and variant sets.

### About Variants and Variant Sets

If you know you have a pen that comes in a variety of colors and tip sizes, you would first create all the variants for the pen line. For example, if you know you will have red, blue, and green pens, and these do not exist in the system, you first create three variant objects: one each for red, blue, and green. Next you need to create a variant set that includes these three variants; the variant set can, for example, be called “color.” Later, when you create the pen object itself, you can tell AllCommerce that the pen object has a number of possible variations; you can do this by listing the Variant Set(s) in the Variant Set field on the Create a Product page.

Creating a variant or a variant set is optional. Not all products will need variants and/or variant sets.

### Performing the Steps

Select the Variant option on the Content Manager to bring up the Create a Variant page. Enter information in the Name field:

Name :

List the variant attribute for an object, such as “red”

### Variant Naming Conventions

All of the AllCommerce templates are based on HTML table objects that expand in width to fit the longest item. Specifically, a table column in which variant names are located will grow to the maximum length of the longest variant name. Long variant names take up a lot of room, and leave less room for the rest of the display. Keep your variant names as brief as possible.

## Creating a Master Product & Variant Products

1. Create Variant objects.
2. Create Variant Set(s). A variant set is a way of grouping Variant objects together. If you had Variant 1-AA Red, 2-AA Blue, 3-AA Green, you could add a Variant Set called Color, which has "1-AA:2-AA:3-AA" entered in the Variant Set entry field on the Create a Product page.
3. Create a Product, and put the relevant Variant Set objects in the Variant Sets entry field of the Product. You can have up to four Variant Sets defined here, all separated by ":".
4. Once you finish creating the object, you will see some links appear on the left hand side (on the spine). You can select Variant Products and this will get you to the place where you instantiate the variant products.

This functionality makes it easier to handle multiple languages; it is easier to only translate the Variant once, and allow it to be included in many Variant Sets, reducing storage and translation time.

### ***Attention!***

Variant Product Titles are now stored in the database. They are regenerated automatically when a Master Product Title changes or when a Variant title changes.



## Creating a Variant Set

A variant set is a group of variants for a particular object. You define it in AllCommerce as a list of differing attributes for an object.

### Performing the Steps

Select Variant Set on the Content Manager to bring up the Create a Variant page.

Enter relevant information for the Name and Brief fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. The Name of the Variant Set defines its grouping; for example, you may have a variant set called Color which has the variants red, blue, and green with the object IDs for red, blue, and green that you entered on the Create a Variant page.

Enter the objIDs of all the variants that are associated with this variant set, separating them with a colon (:).

You are now ready to create a product with the variant sets you just created. When you create the actual pen object, you will enter the objID for this variant set in the Variant Set field on the Create a Product page. See [Creating a Product](#).

## Creating a Product Line

A product line is used to group together any objects that you want displayed on one page. After you define a product line, you use the Link Manager to link all individual objects on the relevant page. See [Controlling Your Merchandise Display](#) for details.

### Brand vs. Product Line

If you wish to show all items that are of a certain *brand*, you do not have to create a product line. Choose a page style for brand and then add links for a brand object under the Link Manager.

However, there are still times when you may want to group a number of objects together on a page, regardless of brand.

For example, you may want to group Computer Monitors on one page. You would create a product line for Computer Monitors, then link all computer monitor objects that you want to display on this page, regardless of brand. However, if you want to display all HP products, you could have a HP brand and use that brand page to link all HP products, including the HP monitor.

### Performing the Steps

Select the Product Line option on the Content Manager page to bring up the Create a Product Line page.

Enter relevant information for the Generate Page?, Title, Brief, Description, Keywords, Status, Page Style, Ranking, and Display on Site fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*.

When you are finished, select Enter Changes to create the new object.

## Creating a Product

All products need to be associated with a brand, so you need to create your product brands before beginning to create any products.

Most of the basic data for a product is stored in the `things` table. However, some data is stored in the `extraprop`, `linkage`, and `inventory` tables as well. You will use the Create a Product page also to create the actual web pages for the product.

### Performing the Steps

Select the Product option on the Content Manager to bring up the Create a Product page. Enter relevant information for the Generate Page?, Title, Brief, Description, Keywords, Status, Page Style, Ranking, and Display on Site fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. Enter information in these additional fields:

Brand: \*

This is where you select the character part of the `msku` for your object that is derived from existing brands. Starting with the `sku` from the brand's definition entry (as in the brand-creation example above), all products are given a unique `sku`. This is incremented separately for each product, starting from the base brand's `sku`. From the above example, where the first brand's `msku` was `aaax0000`, the first product created under this brand would have an SKU of `aaax0001`.

Author:

This is the person creating this object and should have their name referenced in an HTML template with "`__author__`".

Variant Set:

If a product you create has variant sets, you associate those relevant sets with the product in the Variant Set field. You can define up to four variant sets in this field by entering four `objIDs` and separating each by a colon (:).

You can associate any previously defined variant sets with the object by entering the variant `objID` here. Later, you can instantiate the actual variants of the product via the variants link.

If you configure this field, then the product that you are defining is referred to as a *Master Product*. All instances of the product with the variations based on the Variant Set (or sets) are called Variant Products.

Stock Behavior:

Describes how to treat the item or page when its inventory count falls to zero. AllCommerce supports just-in-time, real time, recalled, and discontinued stock levels.

When you are finished, select Enter Changes to create the new object.

Select the link called Variant Products to get to the page where you can create the needed variant permutations of this product; see Creating a Variant. Until this is done, you have really only created something like a new object class. That is, you have defined the possible states for this object, but no actual objects exist yet.

If you need to create more variations at a later time, you can enter the master product's `objID` in the search box on the main page. You will get a display page that has links to Variant Products and to the Link Manager. Select Variant Products to return to that page.

You can also bring up the Create a Variant page by searching for the master product. When AllCommerce displays the search results, select the Variant Products link to bring up the Create a Variant page. You can also create a variant by selecting the `objID` link of the master product and going to the appropriate page.

Select the Enter Changes button. Now, on the main Content Manager page, you will see numerous links. Select the "Variants" link and you will be routed to this display.

## Creating an Article

Select the Article option on the Content Manager page to bring up the Create an Article page.

Creating an article is similar to creating a product, except that the content is mostly text. The control screen appears the same, except for the absence of the area where you would have associated a brand to the object. You do not choose a brand. AllCommerce assigns this value, based on the object type value for article pages.

Enter relevant information for the Generate Page?, Title, Brief, Description, Keywords, Ranking, Status, and Display on Site fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. Enter information in the additional field:

Author: \*

Records the article's author. Reference in an HTML template with "\_\_author\_\_".

When you are finished, select Enter Changes to create the new object.

## Creating an Internal Document

Creating an Internal Document is similar to Creating an Article. If you wish to create internal documents that are not to be published to the web, this is where you do it. Select Internal Document on the Content Manager page to bring up the Create an Internal Document page.

Enter relevant information for the Generate Page?, Title, Brief, Description, Keywords, Ranking, Status, Page Style, and Display on Site fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. Enter information in the additional field:

Author: \*

Records the internal document's author. Reference in an HTML template with "`__author__`".

When you are finished, select Enter Changes to create the new object.

## Creating a Pulldown

You can create, change, or add to the options shown in pulldown menus within AllCommerce. Select the Pulldown option on the Content Manager page to bring up the Create a Pulldown page.

### About Pulldowns

Let us say you have just designed a new home page template for your web site. When you go to create a new home page in one of the managers, like Create a Product Line, you will want to be able to select your new template from the list in the pulldown menu for Page Style.

In order for AllCommerce to know of this template's existence and allow you to choose it, you must first add it to the list of available templates displayed in the Page Style pulldown menu.

To do this, you must first call up the existing Page Style menu in the Create a Pulldown section of the Content Manager by entering Page Style (the pulldown's name) in the Keyword Search entry box on the main Content Manager screen. Using the pulldown name to search shows matches, if any. Select the objID beside the Page Style entry. All current values for the object will be displayed in the large text box near the bottom of the page. If your new template was named `home_page1.txt`, you would enter it as

```
home_page1|Home Page
```

in the text box with the other entries already there. Note that you should not enter the `.txt` extension and that whatever name you wish to be displayed on the pulldown that corresponds to this template is listed immediately after the `|.` Use a `|` to separate the template's actual name from its textual reference.

You can also create new pulldowns that you may want to add to your site. You would have to modify/create templates to work with these pulldowns.

#### ***Attention!***

The following pulldowns are system pulldowns that are used internally by AllCommerce. Do not modify the internal values for these pulldowns. Internal values are the values to the left of the `|` on each row.

- `sel_status`
- `sel_behavior`
- `sel_visibility`
- `sel_orderstatus`
- `sel_warehouse`

### Performing the Steps

Select Pulldown on the Content Manager to bring up the Create a Pulldown page. Enter relevant information for the Generate Page?, Name, and Brief fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. Enter information in the additional field:

Pulldown

This is where you enter the actual list for the pulldown. For the example of warehouses, you would see:

```
1-AA| Warehouse1
2-AA| Warehouse2
```

Where 1-AA and 2-AA are objIDs for the 2 warehouses that are defined in the system. Warehouse1 and Warehouse2 are the words that you will see in any warehouse pulldowns throughout AllCommerce. AllCommerce 1.2 supports one warehouse; see [Creating a Warehouse](#). Each entry must be on its own line, with the 'key' being stored before the "|" and the text to be displayed in the pulldown stored after the "|."

### ***Attention!***

Keep the text in the pulldown brief since this is displayed.

Select Filename:

This is the name of the select file that will be automatically generated for you in the following directory:

```
.../os_allcommerce-1.0.1/bits/<lang>/html/select
```

where <lang> is the language that you selected when you logged into the Administrator.

This file name **MUST** start with "select\_", and **MUST** be unique. If either of these rules are broken, an error message is generated.

When you are finished, select Enter Changes to create the new object. If you modified an existing pulldown, such as `select_pagestyle` for example, the additional choice added or modified will appear in the pulldown in the Content Manager.

After you enter your changes, the filename specified in the Select Filename field is automatically generated under the directory `.../os_allcommerce-1.0.1/bits/<lang>/html/select`, where <lang> is the language that you selected when you logged into the Administrator.

## **Variant Pulldowns**

There is one Variant Pulldown for each Variant Set that is defined in the master product. You will see a pulldown for each variant that you created under the Variant Set field in the master product definition. Each pulldown will have the title from the variant, and will have a pulldown consisting of the text that you created in the variant definition under the Items block. Variant product titles are stored in the database in the `things` table; they are regenerated automatically when a Master Product Title or a Variant title changes.



then select the Status and the Display on Site mode (display or non-display) for the product. Enter the shipping weight and the ranking.

## Performing the Steps

Enter relevant information for the Ranking, Status, and Display on Site fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. For each instance that you wish to create, select one value from ALL of the variant pulldowns and enter its shipping weight as well.

Information in the fields below is automatically generated:

### Title

This field is automatically generated. It is generated based on the title of the master product, followed by a "-". The text for each variant pulldown that you selected will then be displayed, separated by a comma. The display order is the same as the order in which the variants are defined in the Variant Set of the master product.

### Vcode

The `vcode` is stored in the `things` table under the `vcode` field. This is a non-display field. This is also a colon-delimited field which stores the `objID` for each variant selected from the variant pulldowns. These values are stored in the order that the variants are defined in the Variant Set of the master product.

When you are finished, select Enter Changes to create the new object.

## Creating a Warehouse

Select Warehouse from the Content Manager page to bring up the Create a Warehouse page. This is where you define all of the warehouses that you have for your site. Enter relevant information for the Title, Brief, Keywords, and Status fields; see [Common Entry Fields](#) in *Basic Guidelines for Entering Information*. When you are finished, select Enter Changes to create the new object.

After you enter your changes, the `sel_warehouse.txt` file is automatically generated under the directory `.../os_allcommerce-1.0.1/bits/<lang>/html/sel`, where `<lang>` is the language that you selected when you logged into the Administrator.

From here on, the warehouses in the `sel_warehouse` file are displayed in the warehouse pulldowns throughout the site.

The warehouses are also used in Inventory, where each warehouse is listed. This allows you to manually change the inventory for a product on a per-warehouse basis.

This release of AllCommerce supports one warehouse.

## Working with Attributes

An attribute can be loosely defined as additional information about a product, such as whether a product is on sale. For example, you may want to attach the attribute `on_sale` to all your products during a specific time period; this task consists of three steps:

5. Defining the attribute
6. Associating the attribute with all products, that is, all objects of a certain class
7. Defining the display position for the attribute on the Object Definition page, which is where you define what the display on your web site will look like

You will probably use attributes mostly to create Custom Fields on your web site where you display additional information about products.

### Creating an Attribute

Select the Attribute option on the Content Manager to bring up the Create an Attribute page.

The fieldname for the attribute that you enter here defines the attribute, which is stored in the `extraprop` table. For example, you create the attribute `on_sale` with the following values on the pulldown:

Fieldname	<code>on_sale</code>
Description	On Sale
HTML String	<code>select:sel_on_sale</code>
Language Sensitive	No

The Description field shows up on displays when you attach an attribute to an object. The description should, therefore, be short and meaningful. The value in this field can be redefined in other languages.

**Warning!**

Do not redefine the field definition for other languages. Keep the field name in the language you created it in when working with other languages.

The HTML string is described below. Do not make modifications to the code.

```
HTML_TYPE:HTMLOPTION=value:HTMLOPTION=value:...
Where:
  HTML_TYPE:
  text
      HTMLOPTION choices:
      size=#
      maxlength=#
  textarea
      HTMLOPTION choices:
      rows=#
      cols=#
      wrap=visible
  select
      HTMLOPTION choices:
      sel_filename=name of the sel file to use to generate the pulldown. This
      must be the name sel_fieldname where fieldname is the name that you typed
      into Field Name when defining this point. The sel_file must be defined
      before Display on Siteing. You can create the sel_file by adding a
      pulldown.
      Multiple=multiple sets the select file to allow multiple selections at once.
  checkbox
      HTMLOPTION choices:
      name=name of the checkbox
      value=default value, checked or not
```

**Handling Multiple Languages**

Language Sensitive is a flag indicating if the new attribute should be considered language-sensitive. If you choose No, the attribute will retain the same value for all languages. If you choose Yes, you will be able to change the value to this attribute in all languages.

**Associating an Attribute with All Products**

Once you have created the attribute, you associate it with the desired products. To do this, enter the following values on the pulldown:

Name	On Sale
Brief	Boolean if this is on sale or not
Encoding	0   No 1   Yes

Select Filename	sel_on_sale
-----------------	-------------

## Defining the Display Position for an Attribute

Use the Object Definition pulldown to define the position where your attribute will be displayed on your web site. For instructions, see [Defining Display Positions for Objects](#).

To define the display position for the attribute, edit the Object Definition for Product as follows:

Field Name	Entry
Name	Product
Brief	This is an internal field for AllCommerce
GUI	gen Generate_Pages title name brief Brief descript Description keywords on_sale On Sale Today status Status visibility Display on Site pagestyle Pagestyle weight_man Weight behavior Stock Behavior axisnames Variant Set ship_weight Shipping Weight sku_series Brand

When you bring up products, you will see a new pulldown between Description and Keywords, with the tag On Sale Today.

## Associating an Attribute with a Specific Object

In some cases, you may not want to associate an attribute with all products. Let us suppose that you only want to keep track whether a single product is on sale. You do this by associating the attribute with a single product; this task consists of the following steps, described here with the `on_sale` example.

1. Find the object with which you want to associate the `on_sale` attribute. The object must already be defined. If it is not, you must define it before you can associate any attributes with it.
2. Select the Attach/Detach link for the object.
3. Choose the attribute On Sale from the Attach Attributes list in the section box.
4. Select Enter.  
The attribute will move over to the Detach Attributes list; this indicates that the attribute is now attached and you can detach it if you like.
5. Select the Back to Object link.
6. You will see the On Sale attribute appear at the very end of the list of attributes associated with the product.
7. Set the value of this attribute.
8. Select Enter Changes to commit the value to the database.

You can attach/detach several attributes at once by selecting multiple items from the list. Selected attributes will move from Attach to Detach or vice versa depending on where you made the selection.

***Attention!***

You may find that the attributes you selected to attach or detach are not moving from one box to another. This is an indication of a problem with permissions.

Make sure that the following items are owned by the browser user and group:

- The `/os_allcommerce/conf` directory
- The subdirectory in the `/os_allcommerce/conf` directory
- The `html_def.pm` file in the `/os_allcommerce/conf` directory

See [Managing Permissions](#) in the *Zelorate AllCommerce System Administrator's Guide* for details on granting permissions to attach/detach attributes and view attributes once they are associated with products.

## Adding Custom Fields

Custom fields are actually attributes that are added to the default display fields. See [Creating an Attribute](#).

## Defining Display Positions for Objects

You can define the appearance of your web site display on the Object Definition page. The Object Definition page appears when you select the Object Definition button for the object; you can also select the Object Definition option on the Content Manager to bring up this page.

The Object Definition functionality allows you to edit object information and control the display of objects on your web site; it does not remove any information from the database; it allows you to add information only.

***Warning!***

Do not remove any information unless you are absolutely sure that you want that action to be permanent.

For example, let us suppose that you do not want Author information to be stored for Articles. You can remove Author from the Article object definition; AllCommerce will permanently remove existing Author information from being displayed and will stop storing Author information for all your articles from that point onward. However, the existing Author information for Articles is still stored in the database.

## Display Definition Format

The format of the object display definition code is `<tag>_gui`, where `<tag>` is the object type of the object to Display on Site, if defined; otherwise or it is the object class of the object to Display on Site. The series part of the object ID (-A part) must be in lowercase.

You can create different types of object display definitions with the `<tag>_gui` code fragment; `<tag>` can be anything you want, as long as it is not a name that is in use.

The display definition is in the format below:

```
fieldname:options/Tag to Show on Page OR <...>
```

where:

`fieldname` is the fieldname where the data entered is to be stored. It is also the fieldname that is used to access `$html_def` to determine the way in which the data is to be displayed on the browser. This fieldname must be a valid fieldname in the tables.

`:options` is either blank for no options, or `display`.

HTMLOPTIONS choices:

Value=initial value to display if not to be read from the database. This has to be hardcoded.

`<...>` is:

<code>&lt;row_start&gt;</code>	Start a new row, continue adding attributes as a column, until <code>&lt;row_end&gt;</code> .
<code>&lt;row_end&gt;</code>	Finish the row definition started above.
<code>&lt;next_row&gt;</code>	Force starting a new row before reaching <code>&lt;row_end&gt;</code> . This is needed for multiple row generation where each row has a unique row number associated with it, and the data that needs to be included with this one unique row number needs to span on to another row.
<code>&lt;other sel_file to include&gt;</code>	

### Deferring Options Till Runtime

Any portion of the defined display options can be deferred until runtime by surrounding it with `<portion_>`. Surrounding anything with `<portion_>` enables you to insert something dynamically. Unless the option is populated, the `permuteclean` call in the code will remove it at runtime.

### Displaying Multiple Rows

If you need to display multiple rows, like prices or variant products, you must write your own display routine using the display code fragments described earlier. For example, suppose you want

to display multiple rows of shipping information and want to create new or modify existing ones from the same page, you need to write a specific generation routine. This routine should use `<row_start>`, `<new_row>`, and `<row_end>` information. After this is done, the mode can be changed in the routine to `update_rows` to allow updates.



## Managing Orders

The Order Manager allows you to maintain information about existing orders and enter information about new orders.

If you do not know the specific order ID for an order, use the detailed Search feature to look for it. The Search feature allows you to narrow your search by specifying the approximate dates between which the order was placed, the order status, or the warehouse. Accepted order status choices appear in the list of values for that field.

Selecting the Print Report check box will give you a screen print of the search results.

Select the order ID for the order you are interested in and the Order Editing page comes up; here you can edit the orders' status and other data.

### Editing Orders

On the Order Editing page, you will find a number of parameters for the order. You can change or update these as needed.

At the top of the page is the order id for the order you are currently editing. Immediately below this order id is the warehouse this order is to be pulled from. Next is a pulldown menu where you can select from a broad range of order-status options. Below this is a pulldown menu for selecting a shipping method for this order. The shipping method value will initially select the method chosen by the customer when they placed the order on your site. AllCommerce gives you the flexibility to change this shipping method should you need to, but the changeability of this feature is determined by the current order status. If the order status is in-route to a customer, for example, you can no longer change the shipping method for this order. All of the changeable parameters for an order are dependent on the order's status as to whether or not they can be modified.

Below the shipping method pulldown is a header line for Product, Title, Quantity, Price, and Total Purchased (the dollar amount for each line item). Each item ordered is represented as a line entry under these headings with associated data displayed under each column header. The quantity field is changeable under most values for Order Status except when the order is marked as Awaiting Fulfillment, In-route to Customer, Awaiting Funds Capture, or in either of the two voided states. Setting the quantity to zero also zeroes the line total.

The Refund field allows for full or partial refund values. Once a value is entered into this box, it will persist so that you know how much of a refund was given. The Sub total field will reflect the adjusted order amount. To undo a refund, you may enter a negative amount in the box.

The next line consists of three buttons: "Update Info", "Reset", and "New Search". Select "Update Info" when you are done editing the order to make the changes take effect. "Reset" clears any fields back to their defaults, assuming that you have made changes, but have not Selected on "Update Info" yet. "New Search" returns you to the main Order Manager entry page (where you started), so that you may search for other orders.

Below the "Update Info" button, the customer's billing address is displayed. To the right of this are the Sub total, Tax total, Shipping total, and Order total for this order.

The last bit of information is the order's shipping address. These shipping address fields are changeable except when the order status is Awaiting Fulfillment, In-route to Customer, Awaiting Funds Capture, or in either of the two voided states.

## Customers

Customer data, such as addresses, are stored in the database. Each order is related to the customer who placed it.

## Editing Objects

You edit information about an object in the same way you modify information to control how an object is displayed. See [Defining Display Positions for Objects](#).

## Searching for an Object to Edit

You can specify a variety of search criteria from the search screen. They are listed below. Note that this search is put together in an "AND" fashion. Therefore, the Status will always be searched, as well as any other entries that you may make.

Search for Objects:

Enter an `objID`. AllCommerce will display all objects that have an `objID` that resembles the one you entered, and have the status that was entered in the "Status:" pulldown. For example, if you enter 2-AA, you will see 2-AA, 12-AA, 22-AA, and so on.

Keyword Search:

Enter a keyword of interest. All objects that have that keyword in their `Title`, `Brief`, or `Keywords` fields will be matched.

Status:

Active

Will show all products in the `things` table that are currently active.

Inactive

Will show all products in the `things` table that are currently inactive.

Once you make a query, all of the products that matched your query will be displayed.

## Exporting Order Information to XML Format

The `os_allcommerce/tools/xml_order_export.pl` can be used to export the order information from the site database to a text file which is in XML format.

XML Order Export takes as parameters a list of Order IDs (the `objID` key from the order table), separated by spaces. For each order ID input, the order information will be exported. These XML files can then be used to import data into external data sources, such as warehousing systems.

For example, `perl xml_order_export.pl 31-AA 44-AA`

This will export the order with the `objID` 31-AA and the order with the `objID` 44-AA, and all associated data from relational tables.

The following tables are queried to gather the information regarding the order.

### Order

Get information about the order, the user, addresses, price, and shipping weight of the order. Note that since this is a relational database, the values for the addresses, the user are just a key into another table. The actual data must then be retrieved from the other tables. Also, the `ocart` table must be queried to determine what all of the items are that are purchased. The following are the other tables that are queried to retrieve the information.

### Addresses

Find the details about the shipping address and the billing address for an order.

### Contacts

Gives the user name of the individual who purchased this order.

### ocart

Describes each item in the order cart that was purchased with this order. It is a list of all of the items purchased.

### Things

Finds the title of each item in the ocart.

## Exported File

The exported file is stored in the following directory `os_allcommerce/transit/export/order`

It will be named OBJID.xml, where OBJID is the order objID that was passed into the program at the command line.

## Exported File Format

Templates can be generated to layout the format of the exported files, but if none are created, then a dump of the data in XML format is outputted. All of the data is exported from the above tables, with the and values being the name of the field that the value was retrieved from, and the group of fields are surrounded by tags, where table is the name of the table that the fields were selected from.

You can create templates in the directory `os_allcommerce/bits/<lang>/xml_pages`, where `<lang>` is the language that you selected when you logged into the Administrator.

Templates MUST have the following names:

### xml\_page\_order\_format.txt

This template outlines the format of the final .xml file. This file can have hard coded information, along with the following tags which will be replaced with the data gathered and represented from the other tables:

- `__ORDER__` replaced with data from the order table in the default format if no template is specified, or in `xml_page_order.txt` format if this template is defined.
- `__OCART__` replaced with data from the ocart table in the default format if no template is specified, or in `xml_page_ocart.txt` format if this template is defined.
- `__SHIPTO__` replaced with data from the addresses table in the default format if no template is specified, or in `xml_page_shipto.txt` format if this template is defined.
- `__BILLTO__` replaced with data from the addresses table in the default format if no template is specified, or in `xml_page_billto.txt` format if this template is defined.
- `__CONTACTS__` replaced with data from the contacts table in the default format if no template is specified, or in `xml_page_contacts.txt` format if this template is defined.

### xml\_page\_contacts.txt

This template outlines the format of the data that is retrieved from the contacts table. Tags that can exist are `__tag__` where tag is the name of any field that is in the contacts table that you wish to be included in the exported file.

### xml\_page\_order.txt

This template outlines the format of the data that is retrieved from the order table. Tags that can exist are \_\_tag\_\_ where tag is the name of any field that is in the order table that you wish to be included in the exported file.

### xml\_page\_shipto.txt

This template outlines the format of the data that is retrieved from the addresses table for the ship to address. Tags that can exist are \_\_tag\_\_ where tag is the name of any field that is in the addresses table that you wish to be included in the exported file.

### xml\_page\_billto.txt

This template outlines the format of the data that is retrieved from the addresses table for the bill to address. Tags that can exist are \_\_tag\_\_ where tag is the name of any field that is in the addresses table that you wish to be included in the exported file.

### xml\_page\_ocart.txt

This template outlines the format of the data that is retrieved from the ocart table for each item purchased in the order. Tags that can exist are \_\_tag\_\_ where tag is the name of any field that is in the ocart table that you wish to be included in the exported file, as well as \_\_title\_\_ which is retrieved from the things table for each item that is in the order.

Example:

```
xml_page_ocart.txt
__objID__ __title__
```

This would only show the objID and the title for EACH product purchased.

You can define any of the above templates as you desire. It will use all templates that are defined, and use default values for the other data. If you do not want to see any data from one of the tables, you can define a template that is completely blank. Then you will not see any data from that template.

### III. Managing Inventory

Inventory Manager is where you adjust (i.e., increment and decrement) your inventory. While inventory is automatically decremented as purchases are made from your web site, there are also situations where you may need to do this manually. Manual inventory may need to be done for such problems as loss.

Select Inventory Manager from Main Administrator screen. AllCommerce takes you to the Inventory Search screen, where you can search for the objects you need.

#### Adding Inventory

Once you have found the object you are interested via the search function, Select the Inventory link for that object to add/modify inventory. You will be presented with a display which lists all of the warehouses that you have defined in your system with enterable fields to add/modify the inventory amounts.

***Attention!***

If you just see the titles for the fields, and no enterable boxes, as well as no warehouse name, you must create a warehouse before continuing. See [Creating a Warehouse](#).

After creating the Warehouse, you will have enterable fields to change the inventory for products.

Warehouse :

The warehouse for which the inventory can be modified.

In Stock :

The current amount of inventory that is in stock for this product in this warehouse.

***Attention!***

The maximum value for In Stock is 2,147,483,647. The minimum value is -2,147,483,648, as this is a 32-bit, signed number field. If the values are above or below these limits, they are automatically converted to the maximum value, if the number is too high, or the minimum value, if the number is too low.

Status :

Status is how you decide whether the object is currently active or inactive.

Adjust :

This is where you enter what you want to adjust. You enter a positive number to increment the In Stock field, or a negative number to decrement the field.

When you are finished, Select Enter.

After you enter the data, the `In Stock` value will update to represent the changes that you made in inventory, as well as any changes that were automatically made to the inventory for this product while you were in the display.

## Adding Prices

If you do not know the specific object ID for the product for which you want to add or modify pricing, use the detailed Search feature to look for it. The displayed search results show you the object ID for the product, as well as three choices--Links, Inventory, and Pricing.

Select Pricing to create new or modify prices for the product, and to regenerate your page with the new prices. In the checkout process, the lowest active price will be the one displayed. Therefore, you can put in temporary sale prices here.

The Rule Generated? field is currently set to N since this feature is not activated yet. In future releases, there will be a way to automatically generate rule-generated prices. This feature will allow you to generate prices for products based on a specific rule (e.g., 10% off all items for today).

Generate Page

The default value for this is stored by `inventory.cgi` in the `$generate_default` variable. By default, this is set to "not checked". If you wish to change the default to "checked", modify `inventory.cgi`, changing `$generate_default` to 1.

### ***Attention!***

You must have the appropriate permission to generate a page.

Unit Count

Unit Count was created for future support. Currently it is set to 1, but in the future it will be an enterable field. You will then be able to price objects at a discount if the customer buys a specified quantity.

Unit Price

Unit Price is the price of the product. This must be a positive number, with at most two digits after the decimal place. You do not enter the currency symbol (e.g., \$); enter just the price.

## Currency

Choose the currency in which you are operating your web store. Currencies supported in AllCommerce 1.2 are:

- FRF France, Franc
- USD United States of America, US Dollar
- MXP Mexico, Peso
- EUR European Economic and Monetary (EMU), Euro

## Rule Generated

Rule Generated is a function that was created for future compatibility. The rules that are generated through this display are all "N" (not rule generated). Once the rule-generated prices are supported, the existing rule-generated prices for the product will show up in the list of prices, with a Rule Generated flag of "Y".

When you are finished, Select Enter.

### ***Attention!***

If you select Inventory or Pricing for a master product, you will see an intermediate page with links to all variant products for the master. You must then select one of those links, and then follow the steps to add the inventory and prices. See [Adding Inventory](#) and [Adding Prices](#).

If you do not have any variant products defined for the master product, you will receive a message indicating that you must first define the products before you can price them or put them into inventory.



## IV. Controlling Merchandise Display

The Link Manager allows you to establish relationships between all the products you show on your web site, controlling item or page linkages for existing objects. Once you have your templates and cell files created, the Link Manager allows you to change the organization of your web site without any HTML recoding. The Link Manager also allows you to manage cell styles for linked products through the Depth field.

Enter the object ID and search for the product that you want to link.

If you do not know the specific object ID, use the detailed Search feature to look for the object you want. The Search feature allows you to narrow your search by entering different criteria, such as the type of object, its brand, or by using specific keywords.

When you enter a specific and valid object ID, AllCommerce displays the Add New Link and Modify Existing Link sections. These sections contain additional fields where you can enter details about the object, including its display position on your web page. Entering information in all these fields is not mandatory.

When you are finished, you can Enter Changes to record the relationship in the database. You can select the Generate Page checkbox if you have permission to do so.

The Modify Existing Link section also gives you the option to Remove a link. You can perform this task if you have permission to do so.

Generate Page?

The default value of this is stored in `linkage.cgi` in the `$generate_default` variable. By default, this value is set to "not checked". If you wish to change the default to "checked", modify `linkage.cgi`, and change `$generate_default` to 1.

Object:

Enter the `objID` of the object you wish to link.

Alternate Title:

If you wish to substitute a title that is different from the object's actual title, you can enter the alternate here. This will only affect the title shown on the link page, not the object's actual page.

Weight:

This allows you to order the placement of multiple objects that will be linked to this page. This feature is handy if you are linking objects that would naturally go in a particular (e.g., chronological or cardinal) order. The higher the weight value, the closer to the top of the generated page the link will appear. Links with identical weights will be resolved randomly.

Style:

Here you select the cell template for this linked object. You can mix cell styles on the same page if you need to, but remember that cells chosen, although different in their coding, must match the width value of the cell reference in the link pages template. There is a brief introduction to template and cell use in [Working with Templates](#).

A quick way to tell what size of cells you can link to the page is to look at the text behind **Page style is:** tag under the title of the object ID to link. The text will include the name of the Page Style, as well as information in brackets. Choose a cell style that matches at least one of the numbers in the brackets of the page style definition.

Depth:

This text area allows you to enter the maximum number of objects linked to another object. If you wish to allow an iterative cell replacement you can use the master product, or the brand's **objID** instead of entering data for each object in a separate row. For example, you may want to allow cell replacement for an entire product line--all variants of a master product object or all objects belonging to a certain brand.

Select the Generate Page check box if you have permission to do so. Selecting this check box automatically generates the HTML pages for your site.

After you have made your changes, select Enter Changes to make these changes take effect.

## V. Generating Reports for User Activity

The Statistics Manager (Statsman) provides primary and detailed reports based on the records stored in the database tables. These records are gathered while users navigate through stores pages. The reports can be generated dynamically or periodically.

### About the Statistics Manager

The Statistics Manager (Statsman) tracks the activities of a user based on the sid (session id) that is assigned when a user enters the store's site. Therefore, Statsman treats the same sid as one visitor and all the actions which have the same sid will be calculated under this visitor. For this reason, if multiple visitors are using the same machine with the same browser (this may happen at a public internet access terminal or kiosk), when the new visitor begins his/her purchasing without closing all the browsers that have been used by the previous visitor, both visitors will have the same sid, and thus they will be treated as one visitor.

Statsman assumes that a user "admin" is responsible for only administration affairs and should not do any purchasing at the store. Therefore, all the actions done at the store by the user "admin" will be excluded in the statistics reports generated by Statsman. If users defined as "administrators" want to make purchases at the store, they should log out before going to the store's site.

#### *Attention!*

The administrator should close and reopen all the browsers before logging in as "admin."

### Reporting Screens

Statsman provides three kinds of screens:

- Primary Report
- Detailed Stats
- Previous Stats

#### Primary Report

The primary report is the home page of Statistics Manager. It will give the primary statistics report to the user. This is a dynamic report that is triggered by selecting on the link to Stats Manager. By default, the report displays the statistics for the time period the past 24 hours from the moment the user selects the Stats Manager.

#### Snapshot View

Total Number of Page Views is the total number of pages that have been viewed by all the visitors at the store during the time period. The pages that are counted here include store home page, intermediate page, product page, product page with related items, product page for variant, article intermediate page, article page, and internal document page. Note that all the pages that are generated by CGI script such as cart.cgi, welcome.cgi, and checkout.cgi, are not included here.

**Total Number of Visitors:** this is the total number of visitors that have visited the store during the set time period. As mentioned in premises, the number of visitors are counted by sid. One sid is treated as one visitor.

**Number of Orders:** this is the total number of orders that have been placed by all the visitors at the store during the set time period. Things left in the shopping cart and uncompleted orders in checkout are not included. Only completed orders are counted. Note that voided orders are taken off the total number of orders.

**Order Total:** this is the total amount of completed orders that have been placed by all the visitors at the store during the time period. The result is the sum of the net order amount excluding the taxes and shipping charges.

**Top 25 Items:** this lists the top 25 items that have been purchased by all the visitors at the store during the set time period. The items are sorted by quantity, with the biggest number first. The list shows the sku number, quantity, price, and total for each item.

## Detailed Stats

Detailed Stats gives the detailed statistics report about the store for the past 24 hours. This is a dynamic report that is generated at the same time as the primary report. Detailed Stats is saved under the directory `/[site]/data/stats/include/`, and is named with the format `yyyymmdd.html`. When Selecting on Detailed Stats at the home page of Statistics Manager, the page with the current date will be displayed. If multiple detailed reports are generated within the same day, the latest one will replace the previous one. Therefore, there is only one version of detailed report for each date.

For General Statistics and Report, most of the fields are given both an absolute number (Time) and relative number (Percent).

## General Statistics

**Total Number of Page Views** is the total number of pages that have been viewed by all the visitors at the store during the set time period. The pages that are counted here include store home page, intermediate page, product page, product page with related items, product page for variant, article intermediate page, article page, and internal document page. Note that all the pages that generated by CGI script such as `cart.cgi`, `welcome.cgi`, and `checkout.cgi`, are not included here.

**Total Number of Home Page Views:** This shows the total number of visits of the home page during the set time period. Every time `home.shtml` is visited, this number will be added by one. The percentage value is the percentage of home page viewed versus the total pages viewed.

**Total Number of Visitors** is the total number of visitors during the set time period. Every visitor who has been assigned a sid (session id) is counted. Note that every time a returning visitor with a different sid visits the store, he/she will be added again to this number. This number equals to the sum of the number of Logged in and Not Logged in. It also equals to the sum of the Number of Visitors Left and Number of Visitors Purchased.

**Logged in** is the total number of visitors logged in at the store during the time period. The percentage value of this field is the percentage of Logged in Visitors versus Total Number of Visitors.

**Not Logged in** is the total number of visitors without logging in during the set time period. The percentage value of this field is the percentage of Not Logged in Visitors versus Total Number of Visitors.

**Number of Visitors Left** is the total number of visitors who have left without finishing any orders during the set time period. It includes the visitors with an empty cart, visitors with stuff in their cart, and visitors who left the web site before finishing their checkout. Each of these three kinds of visitors will be discussed below. The percentage value of this field is the percentage of Number of Visitors Left versus Total Number of Visitors.

**With Empty Cart:** this category of visitors left the store with an empty shopping cart. They either did not select any purchase button, or removed the items from their shopping cart before they left. The percentage value of this field is the percentage of With Empty Cart versus Number of Visitors Left.

**With Stuff in Cart:** this category of visitors who put some stuff into their shopping carts, but left before they logged in, during the set time period. The percentage value of this field is the percentage of With Stuff in Cart versus Number of Visitors Left.

**Just before Completing Purchase:** this category of visitors had put stuff into their shopping cart and already logged in but left during one of the five checkout steps. The percentage value of this field is the percentage of Just before Completing Purchase versus Number of Visitors Left.

**Number of Visitors Purchased:** number of visitors who left the store with at least one finished order during the time period. The percentage value of this field is the percentage of Number of Visitors Purchased versus Total Number of Visitors.

**Number of New Customers Joined:** total number of visitors who had created their new accounts at the store during the time period. Same visitors who opened more than one accounts are treated as different visitors. The percentage value of this field is the percentage of the Number of New Customers Joined versus Total Number of Visitors.

**Page Views Per Visitor:** total number of pages that has been viewed by each visitors at the store during the time period. This number is calculated from Total Number of Page Views/Total Number of Visitors. If there are no visitors during the set time period, this field shows N/A (Not Applicable).

**Conversion Rate:** average number of pages that has been viewed of each order that has been finished by the visitors during the set time period. This number is calculated from Total Number of Page Views/Number of Orders. If there are no orders during the set time period, this field shows N/A (Not Applicable).

## Report

**Top 25 Pages Visited:** top 25 pages viewed by visitors at the store during the set time period. The pages are sorted by the number of times they have been visited with the biggest number shown first. The percentage value of this field is the percentage of Total Number of Pages Viewed.

**Least 25 Pages Visited:** least 25 pages viewed by visitors at the store during the set time period. The pages are sorted by the number of times they have been visited with the smallest number shown first. The percentage value of this field is the percentage of Total Number of Pages Viewed. Note that if a page is not viewed, it is not counted here.

**Top 25 Browsers Used:** top 25 browsers used by visitors when they visited the store during the set time period. The browsers are sorted by number of times used with the biggest number shown first. The percentage value of this field is the percentage each specific browser versus total number of browsers used.

**Top 25 Refer Sites:** top 25 sites that visitors came from when they first visited the store during the set time period. The sites are sorted by the number of times they have been referred with the biggest number shown first. The percentage value of this field is the percentage each specific refer site versus total number of sites referred.

## Reporting Order Information

**Number of Orders:** this is the total number of orders that have been placed by all the visitors at the store during the set time period. Things left in the shopping cart and incomplete orders at the checkout are not included. Only completed orders are counted. Note that voided orders are taken off the total number of orders.

**Order Total:** this is the total amount of completed orders that have been placed by all the visitors at the store during the set time period. The result is the sum of the net order amount excluding the taxes and shipping charges.

**Cost of Shipping:** this is the total amount of shipping costs for all completed orders during the time period.

**Amount of Tax:** this is the total amount of taxes for all completed orders during the time period.

**Top 25 Items:** this lists the top 25 items that have been purchased by all the visitors at the store during the time period. The items are sorted by quantity, with the biggest number first. It shows the sku number, quantity, price, and total for each item.

## Previous Stats

Reports showing statistical history can be created by setting up a cron job through the system command. Statistics Manager (Statsman) will generate detailed statistical reports for the store for the past 24 hours. The pages created by the cron job are saved under the directory

`/[site]/data/stats/cron/`, and are named with the format `yyyymmdd.html`. If multiple reports are generated in the same day, the latest one will replace the previous one. The date the reports have

been generated is shown inside the Previous Stats section. Choosing the date and Selecting the display button will display the detailed statistics report for the specific date.

The format of the Previous Stats report is exactly the same as [Detailed Stats](#).

For detailed information about setting up a cron job, see [The crontab and Recurring Processes](#) in the *Zelerate AllCommerce System Administrator's Guide*.

## **Limitation**

When the least 25 pages viewed are counted, only the pages with at least one visit are included. In other word, the pages with zero visits are not counted.

**Home Page Viewed** counts the number of times home.shtml has been visited. The home page home.shtml is counted separately from the page that is linked to home.shtml.

## VI. Using Advanced Features

This section describes how to enhance your web site by using some of the more advanced features that AllCommerce provides, such as the Shopping Cart and Checkout functions. Performing these tasks requires [working with templates](#).

### The Shopping Cart

The shopping cart allows customers to select items for purchase from your site.

#### Adding Shopping Carts to Your Web Site

Implementing the `cart.cgi` script enables you to add a shopping cart to your web site. This section describes the different types of carts that are created, how they are created, and what they are used for.

#### Quick Start

If you just want to get `cart.cgi` up and running without reading any of the script's details, you should read the section on templates for `cart.cgi`. This should be sufficient to allow you to test a shopping cart on your site. Remember, you need to implement `welcome.cgi` as well, or you may get some strange results.

#### How Shopping Carts are Created

When a customer is online putting items into their shopping cart, the cart that AllCommerce keeps track of for that person is known as a `tcart` (t for temporary). The `levents` table is where this cart begins life. This table is used to generate unique numbers for various purposes within AllCommerce and a reference for a `tcart` is one of them. Within the `levents` table is a field (`dtable`) which lists which table this event was created for, the `tcart` table is one example. The `series` field is appended after the event id with a "-" to form an id for the cart, "1-a" for example, which is entered as the value in the `tcart` table for the `cartid` field. After a cart id is generated, a `LineId` is generated for the `lineid` field within the `tcart` table. `cart.cgi` puts these values into the `tcart` table as one record, along with data about the product itself. Data such as the product's sku, its unit price, and quantity go in the `tcart`.

This data is accessed during generation (or should we say, permutation) of `page_cart_cart.txt`. This data is actually used to replace the corresponding reference tags within `cell_cart_list.txt`. The `__rowid__v` tags within that cell template get the `LineId` from the `tcart` table, and `cart.cgi` permutes a new `cell_cart_list.txt` for each `tcart` whose `sid` field contains a value that matches the customer's `sid`. All cart creation is done within the function `cartaddcheck()`, located in `lib/state.pm`.

The first time a customer actually selects a purchase button from a page displaying a product for sale, `cart.cgi` is executed and it checks to see whether or not this person already has a `tcart` for this product. If they do, and they are incrementing or decrementing the quantity for this product, the



`quant(ity)` field is updated accordingly after a couple of checks are made to assure that there is sufficient stock, and that the new quantity won't exceed any limits set in `store.pm` (`store_limit_all` and/or `store-limit-jit`). If, on the other hand, there is no `tcart` for this product for this person, then `cart.cgi` first creates one, followed by a line-id.

Now `cart.cgi` checks availability of the stock. This logic is actually performed in the subroutine `"cartaddcheck()"` in `state.pm`. The first check is made to the "inventory" table. The `lstock`, `wh`, and `status` fields are extracted for the record whose `sku` field matches the product's `sku` and the `vcode` field matches the `$var` variable (which would be the product's variant code if it had one; otherwise, it's undefined). `lstock` holds the current stock quantity for the warehouse (`wh`) being checked. Currently, there is only logic to check for a `wh` value of 1. The code has been designed, however, to allow future upgrading to for multiple warehouses.

The next thing that gets checked is the "behavior" field from the "things" table. While there are five possible values for "behavior", there are only two actual behavioral decisions that need to be made. These are done with respect to the values set in `store.pm` for `"store_limit_jit"` and `"store-limit_all"`. If the `behavior` value for the product from the `things` table is `"jt"`, then this is a "just in time" stock item and the quantity for this product within a shoppers cart may not exceed the value of `"store_limit-jit"`. All other values for `behavior` cause `cartaddcheck` to compare a shopper's quantity of this item with that of `"store_limit_all"` and the value for `lstock` from the inventory table. If the shopper is not exceeding the store limit and `lstock` exceeds the currently requested quantity, then the item(s) is/are placed in their `tcart`. Otherwise, an error message advising them of this limit or lack of stock is generated and displayed on `page_cart_cart.txt` when it is permuted and sent to the shopper's browser.

See [Adding Checkout to Your Web Site](#) for details about checkout functionality.

## Templates and Form Elements for `cart.cgi`

As in `welcome.cgi`, `cart.cgi` is also dependent on having a template available for dynamic output to display a shopping cart's contents. This section will also describe the form elements you will need to have on product pages, or any page for which you wish to incorporate cart functionality. The one template required by `cart.cgi` is called:

```
page_cart_cart.txt
```

It must have this name unless you wish to change the code within the script, and it must be present in `os_allcommerce{version}/bits/{lang}/html/pages` if you are not using the sample store. This template's purpose is to display a customer's current shopping cart contents, and provide a link to `checkout.cgi`, or back to shopping.

Here is a simple example for `page_cart_cart.txt`:

```
<HTML>
  <HEAD>
    <TITLE>__storename__ | My shopping cart</TITLE>
  </HEAD>
  <!-- page_cart_cart.txt PM -->
  <BODY LINK="#300" VLINK="#333"
    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__">
    <TABLE BORDER=0 WIDTH= CELLSPACING=0 CELLPADDING=3>
      <TR>
        <TD COLSPAN=3>
          <FONT FACE="verdana,geneva,arial,helvetica"
            COLOR="#999933">
            <B>My Cart</B>
          </FONT><BR>
        </TD>
      </TR>

      __newrow_cell_cart_header__
      __newrow_cell_cart_list__
      __newrow_cell_cart_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=
        CELLSPACING=0 CELLPADDING=3>
        __cell_cart_header__
        __cell_cart_list__
        __cell_cart_buttons__
    </TABLE>
</FORM>
</BODY>
</HTML>

```

This template also requires several cell files that are dynamically inserted into the generated page; they show individual products and their relevant information, such as price and quantity. These cells are replicated for each item that is currently in a customer's cart.

Before discussing the cell files, let's look at the template itself. The only elements of absolute necessity within this template, other than the cell file references, are the values for the form tag. The first value, "pg\_targ", is replaced by `cart.cgi` with a hyperlink to the previous page from which `cart.cgi` was called, if there is no value for it when `cart.cgi` is executed. It is left as is, if there is already a value.

This could happen if `cart.cgi` was called from `page_cart_cart.txt` because a customer wanted to change the quantity of a certain item in their cart. In this case, `cart.cgi` says "Oh, I was called by a page I generated, so I don't need to figure out a value for `pg_targ`". You can actually hear it say this if you turn up your speakers real loud.

For any other condition, "pg\_targ" is replaced with a link to "home.shtml". The value "sid" is given a customer's session id. "vcode" is replaced with the single-character code that identifies a variant of a particular product. The concatenation of "vcode" and a product's sku make up the value for a "vcode", which is an element used in one of the cells that will be described in the next section. The submit button for this is in the `cell_cart_buttons.txt` cell file.

## Adding Checkout to Your Web Site

The Checkout feature allows the customer to complete shopping, select a payment method, make gift selections such as wrapping and cards, and store and edit personal addresses. AllCommerce provides multiple language support for templates and objects, such as Name, Brief, and Description. Checkout also allows multicurrency formats and validates information accordingly. Checkout is implemented through `checkout.cgi`.

### How Checkout Works

When the customer decides to check out and selects Checkout, a call is made to `checkout.cgi` via a link on `page_shopping_cart_v1.txt`. Then `checkout.cgi` converts all the customer's `tcarts` into a `pcart` (p for permanent) that can be referenced from the Order Manager to query for orders made online.

## VII. Customizing the Installation

This section discusses how to modify defaults to customize AllCommerce for your use. Mostly, this section describes how to use templates.

### Working with Templates

AllCommerce uses a template-based publishing model. This model enables you to create a certain type of web page once, and reuse the template to build any number of pages. These templates are called *site templates*. You will use site templates, also called store templates, to create the online store on your web site.

AllCommerce also uses templates for some dynamically generated pages that a customer will interact with on your site, and these are required to be present in a specific location with specific names. These templates are called administrative templates. For detailed discussion about these mandatory administrative templates needed by AllCommerce for dynamic page generation, see [Setting Up Templates](#) in the *Zelerate AllCommerce System Administrator's Guide*.

All html templates are stored in the file system and template names are stored in the database using the AllCommerce Content Manager. You will find templates in the `os_allcommerce` directory. AllCommerce provides a list of field names and functions that can be easily called out of the templates using a very simple AllCommerce expansion language.

### Site Templates

Site templates are the templates that you use to create your online store; they are the ones that your shopper views. Pages for the shopping cart and checkout are also based on site templates.

In general, you will probably find it necessary to have at least the following three types of templates for an online store:

- Intermediate or non-product page
- Product display page
- Article page

### Intermediate and Article Pages

An *intermediate page* really is used to describe just about anything that's not a product page. Examples would be homepage, or a page that displays links to, and a catalog page with short descriptions of products that fall under a particular category.

See *Using the Content Manager* for details on how to use the Content Manager to create your site. Briefly, any data that you enter into one of the object creation sections of the Content Manager can be referenced, or mapped, to a specific location in an html template of your choosing. You simply need to know the reference method AllCommerce uses in order to design your page templates.

In the Content Manager, on the Create an Intermediate Page screen, the values: Title, Author, Brief and Description can all be mapped to locations on the template you create for your page. These particular fields are referenced by "\_\_" followed by the field name in uppercase, followed by "\_\_" . For example, if you wanted Title to appear somewhere on your page, in the template you would reference it with "\_\_TITLE\_\_". The same method would be used for any of the other data fields you wish to reference. Here's a very simple example of a page template:

```
<HTML>
<HEAD><TITLE>__storename__ | __TITLE__</TITLE></HEAD>
<BODY>
<H2>__BRIEF__</H2><BR>
<PRE>__DESCRIPT__</PRE>
</BODY>
</HTML>
```

This could be used for an Article Page, where you could add \_\_AUTHOR\_\_ or, with the addition of some images and links, perhaps a simple home page. Note that "storename" is not in uppercase, this causes generate\_site.pl to make a function call to extract this value from the store configuration values assigned in store.pm. These function calls don't always look only for data in the store configuration, however, but we'll discuss such function calls later. The other values are generally referenced from the 'things' table in the database for your store. They are inserted when the page is created from whichever manager you are using, e.g. Create a Product, Create an Intermediate Page or Create an Article Page, and then extracted by generate\_site.pl as it goes through the database and builds each page from the assigned template.

You might want to have a link on this page to allow customers to search your site for a particular item. To do this, simply add a call to search.cgi somewhere on the page:

```
<HTML>
<HEAD><TITLE>__storename__ | __TITLE__</TITLE></HEAD>
<BODY>
<H2>__BRIEF__</H2><BR>
<PRE>__DESCRIPT__</PRE>

<FORM ACTION="/cgi-bin/om/search.cgi>
<INPUT TYPE="text" name="search" size=20 maxwidth=30>
<INPUT TYPE="submit" value="search">
</BODY>
</HTML>
```

search.cgi looks for the value of the query element named search and uses that as its argument. You will need to supply a template for search.cgi to use for it to display its search results to the customer.

## Intermediate Catalog Page

Another type of Intermediate Page you would probably find useful is the catalog-type page that displays a number of products based on their similarities. For example, you have a number of books by Joe available on your site. Each of these books has a unique Product Page that displays detailed info about the book including its cost, a button to put it into a shopping cart and a picture of the book's cover. You would want to create an Intermediate Page that serves as a catalog-type page which displays all books by Joe available on your site. On this Intermediate Page you will need links to each book's Product Page and maybe a brief description of each book near each link.

Although we haven't yet described how to make and use Product Page templates, let's assume at this point you already have done this and created a bunch of pages for each of Joe's great books. Now we want an intermediate catalog page that lists all of these pages and provides a brief description. Here's an example of a first attempt:

```
<HTML>
<HEAD><TITLE>__storename__|__TITLE__</TITLE></HEAD>
  <BODY>
    <center><H2>__DESCRIPT__</H2></center>
    <TABLE WIDTH = 250>
      <TR><TD>A really great book by Joe <A HREF="/html/joe1.html">
        Joe's first book</A>
      </TD></TR>
```

lots more of these references...

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

But this is not effective since each reference needs to be entered prior to page generation. What is needed is a way to say on a template "put a reference here to a product page, and include these details about it, and replicate this section for however many pages the database says should be on this page". A long sentence to be sure, but a convenient idea. And we can do this in AllCommerce by creating "cells", small html code sections that are inserted into the main intermediate page template for each item linked to the page. So instead of the previous example, we do this:

```
<HTML>
<HEAD><TITLE>__storename__|__TITLE__</TITLE></HEAD>
  <BODY>
    <center><H2>__DESCRIPT__</H2></center>
    <TABLE WIDTH= 250 >
      __column_250_1__
```

```

</TABLE>
</BODY>
</HTML>

```

AllCommerce will look for a file called cell\_250.txt in the site/cells directory and replace \_\_column\_250\_1\_\_ with its contents when generate\_site.pl is executed. You need to create cell\_250.txt and put it in the cells directory. A cell file for the above example should look like:

```
<TR><TD>__DESCRIPT__<A HREF="/html/__SKU__.html">__TITLE__</A></TD></TR>
```

This simple example allows AllCommerce to include the description, title and a hyperlink to the product page for Joe's first book as in the first example. But, since no references are hard coded, any Product Page that was linked to the Intermediate Page can be referenced by this cell. When generate\_site.pl is executed, it checks for each Product Page that is linked to the Intermediate Page. For each of these pages, generate\_site fills in the referenced fields in the cell with data for the Product Page. Although there is only one reference to \_\_column\_250\_1\_\_ in the Intermediate Page's template, generate\_site will put a new cell into the generated page for each Product Page linked to it. So, if you linked five products to this Intermediate (or catalog) page, five cells will be inserted into the final html version of the Intermediate Page, each with the referenced data for the particular products. The number 250 and in \_\_column\_250\_1\_\_ is primarily there to allow you to remember what size table this cell was created for. Notice that the table in this example has a width defined as 250 pixels. The 1 says that this intermediate page has 1 column of cells on it. AllCommerce interprets these values for loading cells into each column. A second column should be labeled \_\_column\_250\_2\_\_ and so on. AllCommerce would expect to find a cell file to be named like: cell\_250.txt, or cell\_somename\_250.txt. You have the flexibility of choosing other cell types when you link products to this page, but they all must have 250 as part of their name if the column is declared in the template to be 250. Within the Merchandise Manager there is a pulldown menu where you choose the cell style to be used when doing linking of Product to Intermediate Pages. The only restriction to your choice of cell style is that the numeric portion of the cell's name (the 250 in these examples) must match the numeric value in the column tag within the template page, e.g. \_\_column\_250\_n\_\_ as above. See the documentation on the Merchandise Manager for a description of how to use it to link pages together.

As a final example, if you wanted another type of cell with a product's picture, you could do it like this:

```

<TR><TD><IMG SRC="/images/products/80/__SKU__80.jpg"
WIDTH =80 HEIGHT =80><BR>__DESCRIPT__<A
HREF="/html/__SKU__.html">__TITLE__</A></TD></TR>

```

This would include a jpg image that is named by the products sku, plus \_80 to indicate that it is an 80X80 pixel image. You do not have to name your images with a reference to their size if you do not wish to, but they do need to have the SKU of the product they represent as part of their name in order to be referenced within a template like this.

## Product Pages

For a Product Page, you probably want a larger image of the product, its price, its description again and most importantly, a button to allow a customer to put it into their shopping cart when they decide to buy it. We've covered how to include all of these elements into a page template with the exception of the product's price and a way of putting it into their shopping cart. To get the product's price onto your template, use `__UNITPR__` anywhere you wish the price to appear. To allow a customer to put the item in their shopping cart, you need to provide a call to `cart.cgi`, and pass it the product's sku. Somewhere within your page's template you will need a form element like this:

```
<FORM method = "post" action = "cgi-bin/om/cart.cgi">
```

And the following hidden input values:

```
<input type = "hidden" name = "sid" value = "">
<input type = "hidden" name = "VSKU" value = "__VSKU__">
<input type = "hidden" name = "add" value = "1">
<input type = "hidden" name = "UID" value = "__UID__">
```

and don't forget a submit button.

The first hidden value for "sid" is to accommodate future functionality for dynamic web site page generation. This allows customer tracking for people who aren't accepting cookies on their web browser. The "VSKU" value tells `cart.cgi` what product this is. The "add" value is set to 1 in this example; it tells `cart.cgi` to add 1 of these items to the customer's shopping cart. "UID" passes the user's id if they have one, this is also for when dynamic page generation is incorporated into AllCommerce.

## A Brief Example of Template and Cell File Use

In this example, we will make a page that displays all objects or products that are under a similar grouping. These might be, for example, the sort of gummy fruit candies that don't taste even remotely like real fruit, but are quite enjoyable nonetheless.

Gummy fruit is a subcategory of candies in general. We will design a simple page that has a heading, a basic description of the types of candies on that page, and, finally, links to display pages of the individual products. We could make a template like this:

```
<HTML>
  <HEAD>
    <TITLE>__title__</TITLE>
  </HEAD>
  <BODY BGCOLOR="#FFFFFF">
    <CENTER>
      <FONT COLOR="#FF0000" SIZE="1">__descript__</FONT>
    </CENTER>

    <TABLE WIDTH=300 BORDER=0>
```



```

<TR>
  <TD>
    <TABLE BORDER=0 WIDTH=150>
      <!--FIRST COLUMN-->

      __column_150_1__

    </TABLE>
  </TD>
  <TD>
    <TABLE BORDER=0 WIDTH=150>
      <!--2ND COLUMN-->

      __column_150_2__

    </TABLE>
  </TD>
</TR>
</TABLE>
<
</HTML>

```

This template would construct a page with two columns, each having two rows. The line "`__column_150_N__`" causes AllCommerce (`generate_site.pl`, technically) to look for a file in the `.../bits/<lang>/html/cells` directory, and replace this line with the contents of that file. In all cases, `<lang>` is the language that you selected when you logged into the Administrator.

This is the same idea as lies behind inserting fields from text areas when creating a page. The number 150 is a way for you to name cells, based on their width in pixels, but it is up to you to enforce this through the proper HTML encoding. AllCommerce only makes sure that the number in the cell reference matches the number that is part of the cell name (150 in this example).

Here is an example of how you might do `cell_150.txt`:

```

<TR>
  <TD>
    <A HREF="__objID__.shtml">__descript__</A>
  </TD>
</TR>
<TR>
  <TD>
    <IMG SRC="/IMAGES/__objID__.gif">
  </TD>
</TR>

```

What this will do is cause `generate_site.pl` to insert text from the description field of a linked object into the anchor text, replacing `__objID__` with the real `objID` of the object you link to the page. Also, if you have a small GIF image of the product, you should name it with the object's `objID`, so that a reference to it will also be generated. Then the lines `__column_150_N__` will be replaced with the modified cell.

You could make a reference to any data you wish about a particular object you want described by a cell. If one of your products is "gummy paramecium" and you already created this object, complete

with price per unit, then you could put in a reference to its price with (\_\_unitpr\_\_). The last number in the column reference (1, 2, ... , N, as in these examples) is used by AllCommerce to split links into N columns during page generation.

Assuming that you have already created objects (pages) for the individual candy items, and that you have a main display page template and cell file ready for use, go to the Content Manager and follow directions for creating the product display page. See [Creating a Product](#).

## VIII. Appendices

This section consists of the following appendices:

- Appendix A: [AllCommerce cgi Scripts](#)
- Appendix B: [Other Scripts](#)
- Appendix C: [AllCommerce Functions](#)
- Appendix D: [Implementing Cookieless Shopping](#)
- Appendix E: [Developers' Corner](#)

## Appendix A: AllCommerce cgi Scripts

This appendix gives you a list of all the AllCommerce cgi scripts and describes their functions. These cgi scripts are located in `cgi-bin/` directory.

CGI Script Name	What It Does
admin/	
admin_object.cgi	Content, Inventory and Pricing Manager
inventory.cgi	Inventory Manager
login.cgi	Administrative login Manager
statsman.cgi	Stats and reporting Manager
linkage.cgi	Link Manager
order.cgi	Order Manager
om/	
search.cgi	Keyword and other criteria search
welcome.cgi	Creating accounts / login
hint.cgi	hint retrieval
cart.cgi	shopping cart
sid.cgi	session id manager
search.cgi	Site search
track.cgi	Track users browsing pattern
wrapper.cgi	Cookieless shopping
ssl/	
checkout.cgi	Checkout

## Appendix B: Other Scripts

This appendix lists the Perl (.pl) and other scripts included in AllCommerce.

Script Name	What It Does
crons/	
generate_site.pl	Produces all the output pages for a site. This script produces a fresh set of pages in the <code>html/&lt;lang&gt;/auto/</code> directory, and then calls a script named <code>update_symlinks</code> which creates references to these pages in the <code>html/&lt;lang&gt;/</code> directory. In all cases, <code>&lt;lang&gt;</code> is the language that you selected when you logged into the Administrator.
generate_temps.pl	Creates temporary cached pulldowns
update_symlinks.pl	Updates symbolic links
run_stats.pl	Creates statistic reports
xml_order_export.pl	Creates an xml file with data about the user's order
revert_carts.pl	Puts items back in stock that were previously reserved in the user's cart
update_conf.pl	Creates the <code>site.pm</code> file with information from the different <code>.conf</code> files
tools/	
import_shipping_tables.pl	Imports fed ex, ups, usps shipping data
pqabuilder.pl	Creates a pqa application using site data for the palm
link_home.pl	Utility that creates a link from an object ID to <code>home.shtml</code>

## Appendix C: AllCommerce Functions

This appendix lists the functions included in AllCommerce.

Function Name	Use
<code>__jump__</code>	Defining templates
<code>__columnwidth__</code>	Defining templates
<code>__cartlink__</code>	Link the shopping cart to ???
<code>cartaddcheck (state.pm)</code>	Checks the sid, orderid, and requested quantity. If the sid is invalid, returns error. If there is no order id, creates a new one. If the = ordered object is not set to JIT, check the inventory level for this = object.
<code>localize_template (localization.pm)</code>	Takes a template name w/o extensions, an ISO 639-3 language code, a flag = indicating the subtree of bits/{lang}/ to look in (nominally "admin" for = the administrative interfaces, "site" for the HTML interface, "wap" for = wireless PDAs and such, or "XML" for the XML) processes returns the = template as a single string.
<code>pcart_to_tcart</code>	Move the objects in the shopping cart (pcart) into the temporary cart = (tcart) when customers place an order.
<code>site.pm</code> <code>site_limit_jit</code>	Stores business rules. The quantity limit for "just in time" objects.
<code>site_limit_all</code>	The quantity limit for all the objects.

## Appendix D: Implementing Cookieless Shopping

Cookieless shopping is implemented through `wrapper.cgi`. The `wrapper.cgi` script is invoked by `inbound.cgi` when the web browser does not accept a cookie, so that shopping can continue.

The wrapper gets the page contents from the Apache server and rewrites the links to pass through the wrapper and display on the contents on the browser.

The wrapper gets the page contents from the Apache server and rewrites the links to pass through the wrapper and display on the contents on the browser.

### Functions

The three major functions are `FixLinks`, `InsertEntry` and `Load_Module`

- `FixLinks` - Rewrite links to pass through wrapper
- `InsertEntry` - Write information about the pages visited in the `Selects` table.
- `Load_Module` - Load `LWP::UserAgent` and exit with an appropriate error message if not found

### Requirements

Wrapper uses `LWP::UserAgent`. To get it working, the following packages have to be installed.

- `URI`
- `HTML::Parser`
- `MIME::Base64`
- `Digest::MD5`
- `libwww-perl`

These packages can be obtained from <http://www.linpro.no/lwp> or CPAN.

### General Guidelines for Testing

- Configure web browser to not accept cookies or to prompt for cookies which you don't accept. On recent Netscape browsers that is the `Edit->Preferences` option. Then select `Category Advanced`. You will see the `Cookies` options.
- You should either see your site with all your links passing through wrapper or an error message on the browser with an appropriate error message.
- Go through the entire checkout process, check to see all links pass through wrapper.
- Check entries in `sessions` and `Select` tables.

### Limitation

Wrapper reads one line at a time and rewrites links to pass through it, hence it arises a lot of limitations. Care is required while creating or modifying templates. Shown below are classical examples, which are not handled by wrapper.

1. HREF= "12-AA.shtml" --- Extra space

HREF = "12-AA.shtml" --- Extra space

HREF="12AA.shtml" --- Hardcoded in digit-char format

HREF="12-AA.html" --- Hardcoded to.shtml

HREF=12-AA.shtml --- NO "

FORM METHOD=POST

ACTION="/cgi-bin/om/search.cgi"

--- Need to be on the same line

FORM METHOD=POST ACTION="/cgi-bin/om/search.cgi" --- Extra space between POST and ACTION

HTTPS requests are handled as HTTP.



## Appendix E: Developers' Corner

This appendix is for developers interested in participating in Open Source projects or modifying and extending the code.

### The Zelerate Development Community

A Zelerate AllCommerce project is a collaborative software development effort aimed at creating a robust source code implementation of applications designed to easily customize online pricing, inventory, and ordering information. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the application and its related documentation. These volunteers are known as the Zelerate AllCommerce Group. In addition, hundreds of users have contributed ideas, code, and documentation to the project.

### About Open Source Development

Zelerate maintains a web site, [www.zelerate.org](http://www.zelerate.org), as a developers' resource for Zelerate AllCommerce. AllCommerce is Zelerate's open source e-commerce Internet application written in Perl and designed for scalability, performance, and ease of administration. The site serves as Zelerate's interface with the Open Source community, featuring development plans, source code freely available under the [GNU General License \(GPL\)](#), and documentation for all audiences. Other items on this site are:

#### *Mailing Lists*

We have mailing lists that connect you to other developers, both internal and external. The mailing lists are tailored to both [beginning AllCommerce users](#) and our more [senior developers](#). You can also subscribe to the [announcements mailing lists](#) which we use to broadcast information about new releases, bug fixes and upcoming events.

The developer mailing list is not a user support forum. It is for people actively working on development of the server code and documentation, and for planning future directions. If you have user/configuration questions, send them to the [newbies mailing list](#).

#### *Bug Tracking*

Our bug tracking software allows developers to keep abreast of code changes and to monitor feature refinement.

#### *Related Information*

In the Features section, developers can find breaking news and articles focusing on our product and related open source issues. We include a column by Arne Flones and a Notes From the Field area where updates from Linux User Groups and other community events are posted. There is also a Calendar of upcoming conferences Zelerate is attending so that members of the Zelerate AllCommerce Group may meet face-to-face on occasion.

### ***Maintainers***

In almost all AllCommerce directories there is a MAINTAINERS file, which is used to specify who maintains the code stored in that directory.

### ***Contributors***

We welcome your contributions to the AllCommerce project. You can contribute to the Zelerate.org site. Download and install the AllCommerce and Phoneystore software, and then contact us through the developers mailing lists at <http://lists.zelerate.org/mailman/listinfo>.

## **How to Get Involved**

Contribute to the site and become a member of the Zelerate AllCommerce Group!

Zelerate AllCommerce is an organic entity; those who benefit from it by using it often contribute back to it by providing feature enhancements, bug fixes, and support for others in public newsgroups. The amount of effort expended by any particular individual is usually fairly light, but the resulting product is made very strong. This kind of community can only happen with open source software -- when someone pays for software, they usually aren't willing to fix its bugs.

We want to see Zelerate AllCommerce used very widely so use our source and send us comments, patches and bug reports. Email us. We like to hear from you.