Foundation Store
## Table of Contents

1. The Foundation Store ................................................................................................................. 1
   1.1. Customizing Foundation ............................................................................................................ 1

2. Tree design .................................................................................................................................. 3

3. The Catalog Directory .................................................................................................................. 5

4. Page Templates ............................................................................................................................ 7
   4.1. Template File Locations ............................................................................................................. 7
   4.2. Themes ..................................................................................................................................... 10
   4.3. Template Definition Files ........................................................................................................ 11
   4.4. Edit Page Function .................................................................................................................. 15
   4.5. Region Templates ................................................................................................................... 16
   4.6. Template Page List .................................................................................................................. 20
   4.7. Special Page List ..................................................................................................................... 22
   4.8. Components ............................................................................................................................ 22

5. The Database Tables ..................................................................................................................... 33
   5.1. 2ndDayAir.csv ......................................................................................................................... 34
   5.2. 450.csv .................................................................................................................................. 34
   5.3. Ground.csv ............................................................................................................................. 34
   5.4. NextDayAir.csv ...................................................................................................................... 34
   5.5. access.asc ............................................................................................................................. 34
   5.6. affiliate .................................................................................................................................... 38
   5.7. area .......................................................................................................................................... 40
   5.8. banner ..................................................................................................................................... 45
   5.9. cat ............................................................................................................................................. 46
   5.10. country ................................................................................................................................. 50
   5.11. downloadable ........................................................................................................................ 51
   5.12. files ....................................................................................................................................... 52
   5.13. inventory.txt .......................................................................................................................... 52
   5.14. locale.txt ............................................................................................................................... 52
   5.15. merchandising.txt .................................................................................................................. 52
   5.16. mv_metadata .......................................................................................................................... 53
   5.17. options ................................................................................................................................... 53
   5.18. order_returns.txt .................................................................................................................... 60
   5.19. orderline.txt .......................................................................................................................... 60
   5.20. pricing .................................................................................................................................... 61
   5.21. products .................................................................................................................................. 62
   5.22. route.txt .................................................................................................................................. 67
   5.23. saletax.asc .............................................................................................................................. 68
   5.24. shipping.asc ............................................................................................................................ 68
   5.25. state.txt .................................................................................................................................. 68
   5.26. transactions.txt ....................................................................................................................... 68
   5.27. userdb.txt .............................................................................................................................. 69
   5.28. variable.txt ............................................................................................................................ 71
Table of Contents

6. HTML Hypertext links ........................................................................................................................................73

7. Images ............................................................................................................................................................75

8. Browser Cookies ............................................................................................................................................77

9. Dependencies in administration ...................................................................................................................79
1. The Foundation Store

The Foundation store is distributed with Interchange to give you a starting point with which to build your e–business. While the Foundation store is designed to be relatively easy to start with, it is still a full–featured demonstration of a number of Interchange capabilities. Once you understand the Foundation store and how it works you are well on your way to understanding the Interchange software.

The following is a list of some popular features:

**Category Searches**

Regardless of the number of products in a catalog, categorizing them makes them easier to find. Pick a field in the database, typically named `category`, and classify the products for search using Interchange.

**Images**

You can display a thumbnail image for the items that have images. To do this, add an image field in the database. (See the 'image' field of the products database.)

**Related Items**

You can embed searches of similar products on an individual product display page with the `[query ...]` or `[loop ...]` tags. Or, if customer data is developed, search a past order database and display products that would be of interest to that customer.

**Reviews/Testimonials**

You can key the placement of a review or testimonial on the existence of a file being in a certain directory. This is reasonable to do when a user is viewing a single product.

1.1. Customizing Foundation

The foundation store has been customized to add just about every ecommerce and content management feature under the sun. If you are asking the question "can I do ...." the answer is almost assuredly "Yes".

Note that customizing the foundation store will take some experience in web design and database operations. Perl experience is also helpful. Most of all it will take persistence and empowering yourself to make changes and watch what happens.

It is usually helpful to make a test store, perhaps just an unmodified copy of foundation, to test out your additions before you make them on your own store. That way you can try anything you want without the chance of causing problems on your own store.
2. Tree design

By determining how users will enter and exit the catalog, complex and intelligent conditional schemes are possible, especially if the Cookies capability is exercised. However, it is recommended that simplicity be used. Consumers will not make purchases if they can't navigate their way around the catalog.

It is important to remember that users will lose their session (and items in their shopping cart) if their browser does not accept cookies and they leave the site. Interchange addresses this problem by using the `area` and `page` tags. If you are using frames, source all frame panes containing Interchange links from an initial page served by Interchange. If you don't do this, the user may have multiple session IDs depending on which frame generated the link.

Note that Interchange can work properly even if the browser doesn't store cookies. In this situation Interchange inserts a session ID into each URL; if the ID is preserved as the user navigates from page to page the session will remain intact.
3. The Catalog Directory

Interchange pages are contained in the catalog directory. Each individual catalog has its own base directory. The catalog directory has the following structure by default:

**catalog.cfg**

File containing configuration directives for a particular catalog. Configuration settings established in the catalog.cfg directory will not effect any other catalogs running under the version of Interchange you are using. Subcatalogs can have differing information in a file named for that subcatalog.

**config**

Directory that will be read when directives are set with the filename notation. For example, the file config/static.pages will be read when the following directive is encountered in the catalog.cfg file.

```
StaticPage <static.pages
```

This directory also contains template information used with the makecat program.

**error.log**

File which contains catalog-specific errors. It is also where any syntax errors in embedded Perl code are shown.

**etc**

Directory normally used for tracking files, order profiles, and other configuration and log information.

**pages**

Directory that contains the pages of the catalog. This can be considered to be the "document root" of the catalog. Pages contained therein are called with the path information after the script name. For example:

```
/cgi-bin/simple/products/gold will call the page in the file pages/products/gold.html.
```

**products**

Directory that contains database source files, including the special Interchange databases shipping.asc, pricing.asc (and other shipping database files).

**session**

Directory that contains session files.

**tmp**

The temporary or scratch directory used for various storage reasons, like retired ID numbers, search paging
files, sort tests, import temporary files, etc. This is the default set by ScratchDir. It can be redefined to be located on another partition.
4. Page Templates

This section describes the files located in the Foundation demo.

4.1. Template File Locations

This diagram shows the directory and file structure used for the default Foundation 'templates' directory. The base will be a directory with the name of your catalog, here called CATROOT.

```
CATROOT/
   |----templates/
   |       |----cart
   |       |----components/
   |       |       |----affiliate_receptor
   |       |       |----best_horizontal
   |       |       |----best_vertical
   |       |       |----cart
   |       |       |----cart_display
   |       |       |----cart_tiny
   |       |       |----category_vertical
   |       |       |----cross_horizontal
   |       |       |----cross_vertical
   |       |       |----modular_buy
   |       |       |----modular_update
   |       |       |----none
   |       |       |----promo
   |       |       |----promo_horizontal
   |       |       |----promo_vertical
   |       |       |----random
   |       |       |----random_horizontal
   |       |       |----random_vertical
   |       |       |----saved_carts_list_small
   |       |       |----search_box_small
   |       |       |----upsell
   |       |       |----upsell_horizontal
   |       |       |----upsell_vertical
   |       |----default --> foundation
   |       |----foundation/
   |       |       |----cart
   |       |       |       |----fullwidth
   |       |       |       |----leftonly
   |       |       |       |----leftright
   |       |       |----regions/
   |       |       |       |----LEFTONLY_BOTTOM
   |       |       |       |----LEFTONLY_TOP
   |       |       |       |----LEFTRIGHT_BOTTOM
   |       |       |       |----LEFTRIGHT_TOP
   |       |       |       |----NOLEFT_BOTTOM
   |       |       |       |----NOLEFT_TOP
   |       |       |----simple
   |       |       |----theme.cfg
   |       |----fullwidth
   |       |----leftonly
   |       |----leftright
   |       |----regions/
   |       |----LEFTONLY_BOTTOM
   |       |----LEFTONLY_TOP
   |       |----LEFTRIGHT_BOTTOM
   |       |----LEFTRIGHT_TOP
   |       |----NOLEFT_BOTTOM
   |       |----NOLEFT_TOP
```
Foundation Store

|-----LEFTRIGHT_BOTTOM
|-----LEFTRIGHT_TOP
|-----NOLEFT_BOTTOM
|-----NOLEFT_TOP
|-----sampledata/
|-----computers/
  |-----images/
  |-----items/
    |-----generic.gif
    |-----gift_certificate_large.gif
    |-----yourimage.gif
  |-----thumb/
    |-----generic_thumb.gif
    |-----gift_certificate.gif
    |-----thumb.gif
|-----products/
  |-----inventory.txt
  |-----merchandising.txt
  |-----mv_metadata.asc
  |-----options.txt
  |-----pricing.txt
  |-----products.txt
  |-----userdb.txt
|-----reports/
  |-----download/
    |-----00352as.pdf
    |-----11993ab.pdf
    |-----22083da.pdf
    |-----49503cg.pdf
    |-----59330rt.pdf
    |-----59402fw.pdf
    |-----66548ch.pdf
    |-----73358ee.pdf
    |-----83491vp.pdf
    |-----90773sh.pdf
|-----products/
  |-----mv_metadata.asc
  |-----products.txt
  |-----userdb.txt
|-----tools/
|-----etc/
  |-----after.cfg
  |-----before.cfg
|-----images/
  |-----items/
    |-----os28004.gif
    |-----os28005.gif
    |-----os28006.gif
    |-----os28007.gif
    |-----os28008.gif
    |-----os28009.gif
    |-----os28011.gif
    |-----os28044.gif
    |-----os28057a.gif
    |-----os28057b.gif
    |-----os28057c.gif
    |-----os28062.gif
    |-----os28064.gif
    |-----os28065.gif
    |-----os28066.gif
    |-----os28068.gif
    |-----os28068a.gif
4.2. Themes

This section explains how themes are defined in Interchange via the STYLE variable and the theme configuration file, theme.cfg.

4.2.1. STYLE

The STYLE variable in CATROOT/products/variable.txt indicates the template style to be used as the theme for the catalog; the appropriate templates for that theme are found in CATROOT/templates/__STYLE__/. (To change the value of the STYLE variable, either edit variable.txt directly or use the table editor feature of the admin interface.)

The default theme for Interchange is the Foundation demo; hence, the STYLE variable is assigned the value 'Foundation' in variable.txt. The theme is defined in catalog.cfg as follows (line numbers added):

```
# Here we set up the catalog theme.
1 ParseVariables Yes
2 ifndef STYLE
3 Variable STYLE default
4 endif
5 include templates/__STYLE__/theme.cfg
```

Variables that make up the look and feel of the STYLE (theme) are defined in the file CATROOT/templates/foundation/theme.cfg, which is read by Interchange in line 5 above.

4.2.2. theme.cfg

The file CATROOT/templates/foundation/theme.cfg serves three purposes:

1. It defines the THEME and THEME_IMG_DIR variables,
2. It defines a cascading style sheet for the theme, and
3. It defines the location of region templates according to the traffic settings for the Interchange daemon.

The THEME variable is used to set the location of the region templates in the traffic settings section of the theme.cfg file. It is also used in the cart template definition file (CATROOT/templates/cart) to set the path of
an image. The THEME_IMG_DIR variable is used to set image paths in the template region files and the
template component files.

The look and feel of the Foundation theme are defined primarily in the cascading style sheet specified in the
theme.cfg file. This

The Interchange TRAFFIC setting, defined system-wide in interchange.cfg, is described in the
??document??. The settings in theme.cfg pertain to the location of region templates for the high and low
traffic settings. For example, if you need to define a separate set of high traffic templates, you would change
the ConfigDir variable in theme.cfg to point to the directory containing those templates.

4.3. Template Definition Files

The template definition files store the name and description of the template as well as components and options
for that template.

```
templates/cart
templates/fullwidth
templates/leftonly
templates/leftright

templates/foundation/cart
templates/foundation/fullwidth
templates/foundation/leftonly
templates/foundation/leftright
templates/foundation/simple
```

4.3.1. Template Walkthrough — leftonly

This section is best read while viewing the file CATROOT/templates/leftonly and the 'Edit Page' page in the
Content Editor of the Interchange Administration Tool.

Looking at the example template definition file, all lines located between the [comment] and [/comment] tags
(lines 1 and 53) control what is available in the Edit Page screen of the Administration Tool.

**Lines 2−5:** Template specification

```
2  ui_template: Yes
3  ui_template_name: leftonly
4  ui_template_layout: LEFTONLY_TOP, UI_CONTENT, LEFTONLY_BOTTOM
5  ui_template_description: Page with top/left areas.
```

Line 2 indicates that this file is a template for the user interface. Line 3 names the template, while Line 4
indicates the regions that comprise the template and that will eventually make up the new page that is created
from the template. Line 5 provides a description used to identify the template when it appears in a Select
Template pull-down menu on the Edit Page of the Administration Tool. This description can be changed or
modified to better describe a new template or a template that is created from the stock templates provided with
Interchange.

**Lines 7−8:** Break

```
7  break:
8    widget: break
```
This code creates a separation line in the Edit Page between sets of options. In the default Interchange installation the line is grey, but the color can be changed. Note — Changing this color applies the change to any catalog served by Interchange.

**Lines 10–11: Page Title**

```itl
10 page_title:
11        description: Page title
```

This code tells Interchange to display a text field on the Edit Page for entering the page title ('Title of New Page' in this example). The value entered is assigned to the scratch variable page_title and is set as a default value at the bottom of the template definition file using the following ITL:

```itl
54 [set page_title][set]
```

which, in turn, sets the scratch variable on the new page using the tag

```itl
[set page_title]Title of New Page[set]
```

The scratch variable page_title is parsed by the following code in the region template specified in the template definition file and called in the new page:

```itl
<title>[scratch page_title]</title>
```

**Lines 13–15: Page Banner**

```itl
13 page_banner:
14        description: Page banner
15        help: Defaults to page title
```

Assigns a textual title for the page to the scratch variable page_banner, which is assigned by the following ITL:

```itl
55 [set page_banner][set]
```

The scratch variable page_banner is set on the new page using the tag

```itl
[set page_banner]Banner of New Page[set]
```

The scratch variable can be parsed in the region template by this code:

```itl
[either]
    [scratch page_banner]
[or]
    [scratch page_title]
[/either]
```

This results in the page banner being displayed if defined. Otherwise, the page title is used.

**Lines 17–20: Members Only**

```itl
17 members_only:
18        options: 1=Yes,0=No*
19        widget: radio
```

4.3. Template Definition Files
This creates a radio−button form element on the Edit Page with the user can specify whether the page can be accessed if a visitor is logged in (Yes) or not (No). The default is indicated by an asterisk.

The scratch variable members_only is assigned by the ITL code

```
56 [set members_only][set]
```

and set on the new page using the tag

```
[set members_only]0[/set]
```

if the page can be accessed without logging in or

```
[set members_only]1[/set]
```

if it can not.

The members_only function is handled by the following code within each region template file:

```
[if scratch members_only]
  [set members_only][/set]
[if !session logged_in]
  [set mv_successpage]@@MV_PAGE@@[/set]
  [bounce page=login]
[/if]
[/if]
```

This code says that if "members only" is set to yes, and the visitor is logged in, to display the page. Otherwise, redirect the visitor to the login page.

**Lines 22–23: Break**

```
22 break1:
23    widget: break
```

Another separation line.

**Lines 25–28: Horizontal Before Component**

```
25 component_before:
26    options: =none, best_horizontal=Best Sellers, \
        cross_horizontal=Cross sell, \
        promo_horizontal=Promotion, \
        random_horizontal=Random items, \
        upsell_horizontal=Upsell
27    widget: select
28    description: Component before content
```

This allows the inclusion of a defined component (included in the CATROOT/templates/components directory) to be displayed before, or above, the page's content. It provides a pull−down menu on the Edit Page displaying the available components. The components, identified here on line 26, can be assigned a name via the value=name convention.
The scratch variable component\_before is assigned in the template definition file by the ITL code

```
57 [set component\_before][set]
```

It is called with the following code within the LEFTRIGHT\_TOP, LEFTONLY\_TOP, and NOLEFT\_TOP region templates:

```
[if scratch component\_before]
   [include file="templates/components/[scratch component\_before]"]
[/if]
```

**Lines 30–33: Horizontal After Component**

```
30 component\_after:
31   options: =none, best\_horizontal=B\_ests Sellers, \ 
    cross\_horizontal=Cross sell, \ 
    promo\_horizontal=Promotion, \ 
    random\_horizontal=Random items, \ 
    upsell\_horizontal=Upsell
32   widget: select
33   description: Component after content
```

Similar to component\_before, this allows the inclusion of a defined component after, or below, the page's content.

The scratch variable component\_before is assigned in the template definition file by the ITL code

```
58 [set component\_after][set]
```

It is called with the following code within the LEFTRIGHT\_BOTTOM and LEFTONLY\_BOTTOM region templates:

```
[if scratch component\_after]
   [include file="templates/components/[scratch component\_after]"]
[/if]
```

**Lines 35–38: Horizontal Item Width**

```
35 component\_hsize:
36   options: 1,2,3*
37   widget: select
38   description: Component items horizontal
```

This setting allows you to choose how many items the horizontal components display. For example, the horizontal best sellers component ("best\_horizontal") uses this setting to randomly select the best sellers. Notice the default is 3 if nothing is defined. It is called by the following code in the promo\_horizontal and random\_horizontal components in the Foundation demo.

```
random="[either][scratch component\_hsize][or]2[/either]"
```

**Lines 40–45: Before/After Banner**

```
40 hbanner:
41   options: =--custom--=, Also see..., Best Sellers, \ 
```

Foundation Store
New items, \nSome of our fine products, \nSpecials, You might also like

widget: move_combo
description: Before/after Banner
help: Banner for Before/after component

Allows a title for the horizontal components to be defined to be displayed in a header above the component’s items. It is called with the [scratch hbanner] tag and used in the Foundation demo in the random_horizontal component.

**Lines 47–51: Special Tag**

```
47 hpromo_type:
48    options: specials=Specials, new=New items
49    widget: select
50    description: Special tag
51    help: Only for a horizontal Promotion
```

This setting is only viable when a promotion is used for a horizontal component. It tells the promotional component which row(s) to evaluate in the merchandising table for display within the component. This setting, used in the promo_horizontal component, typically correlates to the featured column of the merchandising table as follows:

```
[query arrayref=main
  sql="SELECT sku,timed_promotion,start_date,finish_date
    FROM merchandising
    WHERE featured = '[scratch hpromo_type]'
  "]
```

### 4.4. Edit Page Function

Creating a page with the following specifications using the Edit Page function results in the HTML and ITL code displayed below.

**Specifications:**

- **Template:** Page with top/left areas.
- **Page title:** test
- **Page banner:** test
- **Members only:** No
- **Component before content:** Best Sellers
- **Component after content:** Random items
- **Component items horizontal:** 3
- **Before/after Banner:** New items
- **Special tag:** Specials
- **Content:** <P>My first HTML/ITL page!

**Resulting code:**

```
[comment]
ui_template: Yes
ui_template_name: leftonly
```
An important point demonstrated here is the inclusion of the region templates LEFTONLY_TOP and LEFTONLY_BOTTOM through the @_VARIABLE_NAME_@ notation. These are included because of line 4 of the leftonly template definition file:

```
4  ui_template_layout: LEFTONLY_TOP, UI_CONTENT, LEFTONLY_BOTTOM
```

However, understand that you are free to change the region templates used in the file by editing the file itself or, better yet, using an existing region as a starting point for a region of your own design.

The next section explains the structure of region templates.

4.5. Region Templates

Interchange region templates (or "regions") are portions of HTML and ITL that are included in pages within a catalog. Using regions, along with the cascading style sheet defined in theme.cfg, allows you to control the look and feel of specific parts of each catalog page.

The default Foundation region set, found in CATROOT/templates/foundation/regions, includes the following:

- LEFTONLY_TOP
- LEFTONLY_BOTTOM
- LEFTRIGHT_TOP
- LEFTRIGHT_BOTTOM
- NOLEFT_TOP
- NOLEFT_BOTTOM

The Foundation demo uses the Variable feature extensively to simplify hand page editing. Basically, a Variable is a define that permits the substitution of text for a simple __VARIABLE__ string in a page. For example, in the test page above, the variables LEFTONLY_TOP and LEFTONLY_BOTTOM correspond to region templates that provide a logobar, menubar, leftside menu, and copyright footer. Content, consisting of HTML and ITL, is placed within the BEGIN and END CONTENT comments. The following illustration shows how this looks on the page:

```
+-----------------+-----------------+
| LOGOBAR         | MENUBAR          |
+-----------------+-----------------+
```
In this diagram, LEFTONLY_TOP contributes the LEFTSIDE, LOGOBAR, and MENUBAR sections, while LEFTONLY_BOTTOM contributes the COPYRIGHT section.

The following subsections provide an inventory of where each of the region templates, included with the Foundation demo, are used in the pages and template definition files that make up the catalog.

4.5.1. LEFTONLY_TOP

The LEFTONLY_TOP template region is used in the following template pages:

```html
pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/canceled.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/login.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/returns.html
pages/saved_carts.html
pages/ship_addresses.html
pages/ship_addresses_added.html
pages/ship_addresses_removed.html
pages/stock-alert-added.html
pages/stock-alert.html
```

The LEFTONLY_TOP template region is used in the following templates:

```html
templates/foundation/cart
templates/foundation/leftonly
templates/foundation/simple
```
4.5.1.1. Region Template Walkthrough — LEFTONLY_TOP

```html
1  <!-- BEGIN LEFTONLY_TOP -->
2  [if scratch members_only]
3      [set members_only][/set]
4  [if !session logged_in]
5      [set mv_successpage]@@MV_PAGE@@[/set]
6  [bounce page=login]
7  [/if]
8  [/if]
9
10  <html>
11  <head>
12    <title>[scratch page_title]</title>
13    __THEME_CSS__
14  </head>
15
16  <body marginheight="0" marginwidth="0">
17  
18  <!-- top left and right logo -->
19  <table width="100%" border="0" cellspacing="0" cellpadding="0">
20    <tr>
21      <td align="left" valign="middle" class="maincontent">
22        &nbsp;<img src="__THEME_IMG_DIR____LOGO__" />
23    </td>
24    <td align="right" valign="middle" class="maincontent">
25      <img width="174" height="60" src="__THEME_IMG_DIR__logo2.gif"&nbsp;>
26    </td>
27  </tr>
28  </table>
29
30  <!-- menu bar along the top -->
31  <table width="100%" border="0" cellspacing="0" cellpadding="0">
32    <tr>
33      <td width="100%" class="menubar">
34        <a href="[area index]">
35          <img name="Home" border="0" src="__THEME_IMG_DIR__home.gif"></a>
36        <img src="__THEME_IMG_DIR__sep.gif">
37        <a href="[area login]">
38          [if session logged_in]
39            <img alt="Log Out" border="0" src="__THEME_IMG_DIR__logout.gif"></a>
40          [else]
41            <img alt="Log In" border="0" src="__THEME_IMG_DIR__login.gif"></a>
42          [/else]
43        [/if]
44        <img src="__THEME_IMG_DIR__sep.gif">
45        <a href="[area ord/basket]">
46          <img alt="Your Cart" border="0" src="__THEME_IMG_DIR__cart.gif"></a>
47        <img src="__THEME_IMG_DIR__sep.gif">
48        <a href="[area ord/checkout]">
49          <img alt="Check Out" border="0" src="__THEME_IMG_DIR__checkout.gif"></a>
50        </td>
51      <td>
52        <a href="[area customerservice]">
53          <img alt="Customer Service" border="0" src="__THEME_IMG_DIR__service.gif"></a>
54        </td>
55      <td>
56        <a href="[area aboutus]">
57          <img alt="About Us" border="0" src="__THEME_IMG_DIR__about.gif"></a>
58      </td>
59    </tr>
60  </table>
```

Foundation Store
4.5.2. LEFTONLY_BOTTOM

The LEFTONLY_BOTTOM template region is used in the following template pages:

pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/canceled.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/login.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/returns.html
pages/saved_carts.html
pages/ship_addresses.html
pages/ship_addresses_added.html
pages/ship_addresses_removed.html
pages/stock-alert-added.html
pages/stock-alert.html

The LEFTONLY_BOTTOM template region is used in the following templates:

templates/foundation/cart
templates/foundation/leftonly
templates/foundation/simple

4.5.3. LEFTRIGHT_TOP

The LEFTRIGHT_TOP template region is used in the following template pages:
The LEFTRIGHT_TOP template region is used in the following templates:

 templates/foundation/leftright

4.5.4. LEFTRIGHT_BOTTOM

The LEFTRIGHT_BOTTOM template region is used in the following template pages:

 pages/browse.html
 pages/index.html
 pages/results.html
 pages/results_big.html
 pages/swap_results.html

The LEFTRIGHT_BOTTOM template region is used in the following templates:

 templates/foundation/leftright

4.5.5. NOLEFT_BOTTOM

The NOLEFT_BOTTOM template region is used in the following template pages:

 pages/ord/checkout.html
 pages/splash.html

The NOLEFT_BOTTOM template region is used in the following templates:

 templates/foundation/fullwidth

4.5.6. NOLEFT_TOP

The NOLEFT_TOP template region is used in the following template pages:

 pages/ord/checkout.html
 pages/splash.html

The NOLEFT_TOP template region is used in the following templates:

 templates/foundation/fullwidth

4.6. Template Page List

/home/ic/catalogs/ft/pages/:

 aboutus.html
 account.html
 browse.html
canceled.html
change_password.html
contact.html
customerservice.html
deliver.html
flypage.html
help.html
index.html
login.html
logout.html
lost_password.html
modular_modify.html
new_account.html
privacy_policy.html
process_return.html
quantity.html
results_big.html
results_either.html
results.html
returns.html
saved_carts.html
ship_addresses_added.html
ship_addresses.html
ship_addresses_removed.html
splash.html
stock_alert-added.html
stock_alert.html
swap_results.html

/home/ic/catalogs/ft/pages/admin/report_def:

Order%20Status.html
Products%20to%20edit.html

/home/ic/catalogs/ft/pages/admin/reports:

Order%20Status.html
Products%20to%20edit.html

/home/ic/catalogs/ft/pages/affiliate:

index.html
login.html

/home/ic/catalogs/ft/pages/ord:

basket.html
checkout.html

/home/ic/catalogs/ft/pages/query:

check_orders.html
get_password.html
order_detail.html
order_return.html

4.5. Region Templates
4.7. Special Page List

/home/ic/catalogs/ft/special_pages/:

badsearch.html
canceled.html
cc_not_valid.html
confirmation.html
failed.html
interact.html
missing.html
needfield.html
nomatch.html
noprompt.html
notfound.html
order_security.html
reconfig.html
sec_faq.html
security.html
violation.html

4.8. Components

• Added new [control] and [control−set] tags to set series of Scratch− like option areas. Used for components in UI content editing.

Interchange components are portions of HTML and ITL that are included in pages within a catalog depending on options set in the Administration Tool. The default component set includes the following:

affiliate_receptor
best_horizontal
best_vertical
cart
cart_display
cart_tiny
category_vertical
cross_horizontal
cross_vertical
modular_buy
modular_update
none
promo
promo_horizontal
promo_vertical
random
random_horizontal
random_vertical
saved_carts_list_small
search_box_small
upsell
upsell_horizontal
upsell_vertical

/home/ic/catalogs/ft/templates/components:
4.8.1. affiliate_receptor

Not used in Foundation demo

4.8.2. best_horizontal

The best_horizontal component is used in the following templates:

```text
templates/foundation/cart
templates/foundation/leftonly
templates/foundation/leftright
```

Not used in Foundation demo pages

4.8.3. best_vertical

The best_vertical component is used in the following template:

```text
templates/foundation/leftright
```

Not used in Foundation demo pages

4.8.4. cart

The cart component is used in the following page:

```text
pages/ord/basket.html
```

4.8.5. cart_display

The cart_display component creates a small shopping cart that is displayed on the search results page (pages/results.html). It is displayed after an item in a list of results from a search is added to the shopping cart. cart_display is called in results.html by the following include statement:

```text
[include file="templates/components/cart_display"]
```

The cart_display component is used in the following pages:

```text
pages/results.html
```

4.8.5.1. Component Walkthrough -- cart_display

The remainder of this section is best read in conjunction with the file CATROOT/templates/components/cart_display in a text editor.

Lines 1–6: Component Specification

```text
1  [comment]
2  ui_component: cart_display
3  ui_component_group: info
4  ui_component_label: Smaller cart for display in content area
5
```
These lines control what is shown in the Edit page screen of the admin interface.

Line 8 is an HTML comment noting the start of the code for the component. (Note that this can serve as a useful debugging tool to help you locate the component in the resulting HTML generated by Interchange when you view the source of a page loaded in the browser.)

Line 9 checks to see if there are items in the shopping basket. If there are, the remaining code up to the closing [/if] tag on line 64 is executed. If not, Interchanges continues executing the remaining code in results.html (the file that calls the cart_display component).

Line 10 centers the table started in line 11. Lines 12–29 create a header row in the shopping cart consisting of the header titles Action, SKU, Description, Quantity, Price, and Extension.

Line 30 defines the remainder of the table as a section while the [item-list] tag on line 31 tells Interchange to execute the code up to the closing tag ([/item-list] on line 59 for each item the customer has ordered so far.

4.8. Components
Line 33 begins the next row in the table. The [item–alternate] tag provided as the value of the class attribute tells Interchange to alternate between displaying the rows according to the "maincontent" and "contentbar1" styles (gray and white, respectively).

Lines 34–36 create a link to the shopping cart (basket.html) where the customer can remove or change the quantity of the item ordered.

Line 37 displays the SKU of the item. Lines 38 and 39 provide a link to the product display page (flypage.html) for the item. The [item–description] tag providing the content of the [page] tag enables the item's name to be displayed as the link to the product display page.

Line 41 checks whether the item is a gift certificate. If it is it displays "Amount of gift:" and the [item–quantity] (number of gift certificates, in this case) under the headings "Quantity" and "Price", respectively. Otherwise, lines 48 through 50 display the quantity and price of the item ordered. Lines 45 or 53 (depending on whether the item is a gift certificate) display the item subtotal (quantity multiplied by price) for the item under the heading "Extension".

Lines 59 through 64 close out the tags for the component, and line 66 indicates the end of the component code.

4.8.6. cart_tiny

The cart_tiny component is used in the following pages:

pages/account.html
4.8.7. category_horizontal

Not used in Foundation demo pages or templates.

4.8.8. category_vertical

The category_vertical component provides a listing of all products in the catalog, organized by prod_group (e.g., Hand Tools, Ladders). category_vertical is usually displayed in the LEFTSIDE section of the page, under the search_box_small component.

The category_vertical component is used in the following pages:

pages/browse.html
pages/canceled.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/index.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/saved_carts.html
pages/ship_addresses.html
pages/query/order_return.html
4.8.8.1. Component Walkthrough -- category_vertical

The remainder of this section is best read while viewing the file CATROOT/templates/components/cart_display in a text editor.

**Lines 1–6: Component Specification**

    [comment]
    ui_component: category_vertical
    ui_component_group: Navigation
    ui_component_label: Vertical category list

Page_class:

    label: Page class
    widget: select
    lookup: which_page
    db: area
    help: Defines which sets of items should be displayed
    advanced: 1

Set_selector:

    label: Page area selector
    widget: select
    db: area
    lookup: sel
    help: Defines which sets of items should be displayed
    advanced: 1
[/comment]

These lines control what is shown in the Edit page screen of the Administration Tool.

    <tr><td align="center" class="categorybar">
    <br>
    <table>
    <!−− BEGIN COMPONENT [control component category_vertical] −−>

Lines 23–25 set up the row and table within that row that will hold the vertical category list. Line 27 identifies the start of the code for the list.

    [loop]
    prefix=box
    search="
    fi=area
    st=db
    tf=sort
    ac=0
    ac=0
    co=yes
    36
    co=yes
    37
    sf=sel
    39
    op=eq
    40
    se=[control set_selector left]
    41
    sf=which_page
    43

4.8. Components
Lines 28–45 build a list of product categories obtained through a search of the area table.

```
47    <tr>
48      <td valign="top" class="categorybar">
49        <b>[box-exec bar_link]area[/box-exec]</b>
50    </td>
51    </tr>
52    <tr>
53      <td valign="top" class="categorybar">
54      [set found_cat][/set]
55      [loop prefix=cat
56          search="
57              fi=cat
58              st=db
59              tf=sort
60              tf=name
61              rf=code,name
62              sf=sel
63              se=[box-code]
64          "
65       &nbsp;&nbsp;[cat-exec bar_link]cat[/cat-exec]<BR>
66      [/loop]
67    </td>
68    </tr>
69    [/loop]
70    </td>
71    </tr>
72    [/loop]
73    </table>
74    <br>
75    <br>
76    </tr></tr>
77    <br>
```

Lines 47–78 generate a list of links based on the products and product categories identified in the search.

**4.8.9. cross_horizontal**

The cross_horizontal component is used in the following pages:

- pages/browse.html
- pages/index.html
- pages/results.html
- pages/results_big.html

The cross_horizontal component is used in the following templates:

- templates/foundation/cart
- templates/foundation/lemonthly
- templates/foundation/lefright
4.8.10. cross_vertical

Not used in Foundation demo pages.

The cross_horizontal component is used in the following templates:

```
templates/foundation/lefright
```

4.8.11. modular_buy

The modular_buy component is used in the following pages:

```
pages/flypage.html
```

The modular_buy component is used in the following templates:

```
templates/components/modular_update
```

4.8.12. modular_update

The modular_update component is used in the following pages:

```
pages/modular_modify.html
```

4.8.13. promo

The promo component is used in the following pages:

```
pages/contact.html
pages/results_big.html
```

4.8.14. promo_horizontal

The promo_horizontal component is used in the following pages:

```
pages/aboutus.html
pages/canceled.html
```

The promo_horizontal component is used in the following templates:

```
templates/foundation/cart
templates/foundation/lefright
```

4.8.15. promo_vertical

Not used in Foundation demo pages.

The promo_horizontal component is used in the following templates:

```
templates/foundation/lefright
```
4.8.16. random

The random component is used in the following pages:

```
pages/browse.html
pages/index.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/results.html
pages/swap_results.html
```

4.8.17. random_horizontal

Not used in Foundation demo pages.

The random_horizontal component is used in the following templates:

```
templates/foundation/cart
templates/foundation/lefonly
templates/foundation/lefright
```

4.8.18. random_vertical

Not used in Foundation demo pages.

The random_vertical component is used in the following templates:

```
templates/foundation/lefright
```

4.8.19. saved_carts_list_small

The saved_carts_list_small component is used in the following pages:

```
pages/ord/basket.html
```

4.8.20. search_box_small

The search_box_small component is used in the following pages:

```
pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/browse.html
pages/canceld.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/index.html
pages/login.html
pages/logout.html
pages/modular_modify.html
```
The `search_box_small` component is used in the following templates:

- templates/regions/LEFTONLY_TOP
- templates/regions/LEFTRIGHT_TOP

### 4.8.21. upsell

Not used in Foundation demo pages.

### 4.8.22. upsell_horizontal

The `upsell_horizontal` component is used in the following pages:

- pages/flypage.html

The `upsell_horizontal` component is used in the following templates:

- templates/foundation/cart
- templates/foundation/leletonly
- templates/foundation/leftright

### 4.8.23. upsell_vertical

Not used in Foundation demo pages.

The `upsell_vertical` component is used in the following templates:

- templates/foundation/leftright
5. The Database Tables

Interchange catalogs are centralized around the database. You can alter any of the standard databases, add new databases, or remove unneeded databases

The foundation catalog includes the following tables, organized here by content:

- **Your site content data**
  - area.txt
  - cat.txt
  - downloadable.txt
  - merchandising.txt
  - options.txt
  - pricing.txt
  - products.txt

- **Customer data**
  - access.asc
  - gift_certs.txt
  - userdb.txt

- **Transaction–related data**
  - inventory.txt
  - orderline.txt
  - order_returns.txt
  - transactions.txt

- **Third–party relationship data**
  - affiliate.txt
  - banner.txt

- **Site administrative data**
  - component.txt
  - files.txt
  - ichelp.txt
  - icmenu.txt
  - locale.txt
  - mv_metadata.asc
  - route.txt
  - shipping.asc
  - variable.txt

- **Shipping and tax**
  - 2ndDayAir.csv
  - 450.csv
  - country.txt
  - Ground.csv
  - NextDayAir.csv
  - salestax.asc
  - state.txt

You may also see symbolic links pointing to index tables, for example products.category.txt linking to products.txt. These are automatically generated indexes, in this case into the products table to speed category searches. See Dictionary Indexing With INDEX in the database documentation for details about auto–indexing of text databases.
The following dictionary lists and describes each table used in the Foundation demo.

5.1. 2ndDayAir.csv

Shipping table from UPS (http://www.ups.com/using/services/rave/rate/). This and all shipping tables should be updated periodically.

5.2. 450.csv

Shipping table from UPS for 450xx Zip Code origin. You will probably need to get your own from the UPS site (http://www.ups.com/using/services/rave/rate/) and clip the headers.

5.3. Ground.csv

Shipping table from UPS (http://www.ups.com/using/services/rave/rate/).

5.4. NextDayAir.csv

Shipping table from UPS (http://www.ups.com/using/services/rave/rate/).

5.5. access.asc

Administrative access table. This table is used by the Administration Tool. For more description on these fields, see the Interchange Administration Tool guide.

Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Login name or group name (group names begin with ':')</td>
</tr>
<tr>
<td>password</td>
<td>Hashed password</td>
</tr>
<tr>
<td>name</td>
<td>Administrator's name</td>
</tr>
<tr>
<td>last_login</td>
<td>Last login time</td>
</tr>
<tr>
<td>super</td>
<td>Set to 1 if superuser</td>
</tr>
<tr>
<td>yes_tables</td>
<td>Tables the user may edit</td>
</tr>
<tr>
<td>no_tables</td>
<td>Tables the user may not edit</td>
</tr>
<tr>
<td>upload</td>
<td>No Description</td>
</tr>
<tr>
<td>acl</td>
<td>No Description</td>
</tr>
<tr>
<td>export</td>
<td>No Description</td>
</tr>
<tr>
<td>edit</td>
<td>No Description</td>
</tr>
<tr>
<td>pages</td>
<td>No Description</td>
</tr>
<tr>
<td>files</td>
<td>No Description</td>
</tr>
<tr>
<td>config</td>
<td>No Description</td>
</tr>
<tr>
<td>reconfig</td>
<td>No Description</td>
</tr>
<tr>
<td>groups</td>
<td>Administrator's group memberships</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>meta</td>
<td>No Description</td>
</tr>
<tr>
<td>no_functions</td>
<td>Explicitly disallowed functions</td>
</tr>
<tr>
<td>yes_functions</td>
<td>Allowed functions with permission flags</td>
</tr>
<tr>
<td>table_control</td>
<td>No Description</td>
</tr>
<tr>
<td>personal_css</td>
<td>Administrator's personal CSS (for admin screen presentation)</td>
</tr>
</tbody>
</table>

### 5.5.1. username

**Example Data**

```
:admins
:users
BigUser
goody
ic
```

The login name for an administrator or an administration group. Group names are prefixed with a colon (`:`).

### 5.5.2. password

**Example Data**

```
Ksjs65bMNjPQ
```

Hashed password.

### 5.5.3. name

**Example Data**

```
Interchange Site Administrator
Interchange Site Associates
Business Users
2nd Shift
Mr. Jones
Inbound Sales
```

Descriptive name for the administrator or administration group.

### 5.5.4. last_login

**Example Data**

```
989424489
```

Last login time (in unix time() format).

5.5. access.asc
5.5.5. super

Boolean value. If true (1), the administrator has Interchange Site Administrator privilege.

5.5.6. yes_tables

Example Data

    affiliate=vcx component=v gift_certs=v inventory=vx ...
    NONE

Tables this administrator or administration group can access. This is a space–delimited list of
'table_name=permission_flags' entries.

5.5.7. no_tables

Example Data

    access mv_metadata variable

Tables this administrator or administration group can not use. This is a space–delimited list of tables names.

5.5.8. upload

No Description

5.5.9. acl

No Description

5.5.10. export

No Description

5.5.11. edit

No Description

5.5.12. pages

No Description

5.5.13. files

No Description
5.5.14. config

No Description

Example Data

Allowed Values

5.5.15. reconfig

No Description

5.5.16. groups

Example Data

   ausers
   busers

Allowed Values

Groups the site user belongs to. You can set permissions for groups.

5.5.17. meta

No Description

5.5.18. no_functions

Example Data

   orderstats trafficstats

Space–delimited list of functions explicitly not allowed for the site user.

5.5.19. yes_functions

Example Data

   item=lvecd itemtype=lvc order=lvca orderstats trafficstats ...
   NONE

Functions the site user can perform. This is a space–delimited list of functions, with permission flags if appropriate.

Usage examples

   • dist/lib/UI/pages/admin/access_permissions.html
5.5.20. **table_control**

No Description

Usage examples

- dist/lib/UI/Primitive.pm
- dist/lib/UI/pages/admin/special/keyViolation.html
- dist/lib/UI/usertag/if_mm

5.5.21. **personal_css**

Used in the Administration Tool screens to make personal changes to the page presentation. This is done by creating your own personal CSS (cascading style sheet).

Usage examples

- dist/lib/UI/pages/admin/preferences.html

5.6. **affiliate**

cat_root/products/affiliate.txt

This table contains data related to your affiliate programs. See also the **affiliate_receptor** component.

**Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>affiliate</td>
<td>Affiliate ID</td>
</tr>
<tr>
<td>name</td>
<td>Name of affiliate organization</td>
</tr>
<tr>
<td>campaigns</td>
<td>Campaigns this affiliate participates in</td>
</tr>
<tr>
<td>coupon_amount</td>
<td>Discount for customers from affiliate participating in coupon campaign</td>
</tr>
<tr>
<td>join_date</td>
<td>When the affiliate signed with you</td>
</tr>
<tr>
<td>url</td>
<td>Your default URL to use for customers coming from the affiliate site (not the affiliate's home page)</td>
</tr>
<tr>
<td>timeout</td>
<td>Timeout in seconds after which purchases are no longer credited to the affiliate</td>
</tr>
<tr>
<td>active</td>
<td>Boolean, set to 1 for active affiliates</td>
</tr>
<tr>
<td>password</td>
<td>Affiliate login password</td>
</tr>
<tr>
<td>image</td>
<td>Affiliate's logo</td>
</tr>
</tbody>
</table>

5.6.1. **affiliate**

Example Data

- consolidated
- hardhat
This field contains the unique Affiliate ID.

5.6.2. name

Example Data

Consolidated Diversified
Hardhat Construction

This is the descriptive name of the affiliate.

5.6.3. campaigns

Example Data

coupon

This field lists the campaigns that the affiliate participates and enables campaign features and tracks traffic from advertising campaigns. The foundation catalog implements a coupon campaign in the affiliate_receptor component. If you want to add campaigns, you will also need to develop the appropriate logic within the affiliate_receptor component and pages that use it.

5.6.4. coupon_amount

Example Data

5

This is the discount offered customers from the affiliate participating in the coupon campaign.

Note — This is implemented in the affiliate_receptor component as a flat discount amount. If you wanted a percentage discount instead, you would modify the [discount] tag in catalog_root/templates/components/affiliate_receptor (see the [discount] tag for more detail).

5.6.5. join_date

Example Data

20000827
20000910

This is the date when the affiliate signed with you.

5.6.6. url

Example Data

http://demo.akopia.com/~hardhat
http://www.minivend.com/consolidated/

5.6. affiliate
The value in this field is used to direct visitors coming from the Affiliate to your home page or a page you have designed specifically for visitors from that Affiliate's site. Note that this should not be the URL of the Affiliate's home site.

**5.6.7. timeout**

Example Data

```
0
3600
```

The value in this field is used to specify the amount of time a customer has to place an order to still give the Affiliate credit for it. If the customer goes over this amount of time, the Affiliate doesn't get credit for the customer visit. The timeout delay is measured in seconds, with the value of 0 (zero) disabling it. It is recommended that you use a value in the thousands to make sure the customer has enough time to shop.

**5.6.8. active**

This is a boolean value indicating whether the affiliate is active.

**5.6.9. password**

Example Data

```
akopia
```

Password for affiliate login (see catalog_root/pages/affiliate/login.html). Note that the password is stored in plaintext by default.

**5.6.10. image**

Example Data

```
http://demo.akopia.com/~hardhat/images/logo.gif
http://www.minivend.com/consolidated/conslogo.gif
```

Affiliate's logo image.

**5.7. area**

```
cat_root/products/area.txt
```

This table is used to implement dynamic navigation bars. For example, it is used in the category_horizontal and category_vertical components. Note the similarity to the cat table, since both area and cat tables supply data for building links to results pages.

When building entries in a navigation bar, it is the bar_link subroutine in the /dist/catalog_before.cfg configuration file that actually reads and processes the values from the table.

See also the following catalog and administrative templates:
Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Unique key</td>
</tr>
<tr>
<td>sel</td>
<td>Space-delimited list of navigation bars to contain the entry</td>
</tr>
<tr>
<td>name</td>
<td>Display label</td>
</tr>
<tr>
<td>which_page</td>
<td>Page class in which the navigation bar may appear</td>
</tr>
<tr>
<td>sort</td>
<td>Sorting prefix for entry (preempts standard alphanumeric sort)</td>
</tr>
<tr>
<td>display_type</td>
<td>How to label links in the navbar (name, icon, url or image)</td>
</tr>
<tr>
<td>image</td>
<td>Image URL (if appropriate)</td>
</tr>
<tr>
<td>image_prop</td>
<td>HTML attributes for output &lt;img&gt; tag (if appropriate)</td>
</tr>
<tr>
<td>banner_image</td>
<td>Image name for use in target page</td>
</tr>
<tr>
<td>banner_text</td>
<td>Text for use in target page</td>
</tr>
<tr>
<td>link_type</td>
<td>Type of links in the navbar (external, internal, simple, complex)</td>
</tr>
<tr>
<td>url</td>
<td>Target for internal or external link_type</td>
</tr>
<tr>
<td>tab</td>
<td>Database table file to use with 'simple' link_type</td>
</tr>
<tr>
<td>page</td>
<td>Results page to use with 'simple' link_type</td>
</tr>
<tr>
<td>search</td>
<td>Search spec used with 'complex' link_type</td>
</tr>
<tr>
<td>selector</td>
<td>The selector used to scan the products table for products in the category</td>
</tr>
<tr>
<td>link_template</td>
<td>Overrides template used for building navbar links.</td>
</tr>
</tbody>
</table>

5.7.1. code

Example Data

```
1
2
3
```

Unique key.

5.7.2. sel

Example Data

```
left
```

Space-delimited list of navigation bars that should contain the entry. Note that comma or null should also work as a delimiter.

5.7. area
5.7.3. name

Example Data

Hand Tools
Hardware
Ladders
Measuring Tools
Painting Supplies
Safety Equipment
Specials
Tool Storage

Label to display.

5.7.4. which_page

Example Data

all

Page class in which the navigation bar may appear.

5.7.5. sort

Example Data

00
03
04
05
06

Lexographic (alphanumeric) sorting prefix. Note use of '03' rather than '3', which would sort after '13'. This controls the order of the categories in your navigation bar.

If this is not set, your navbar entries will sort in alphabetical order.

5.7.6. display_type

Example Data

name
icon
url
image

What to use for the labels in the navigation bar (for example, name, icon, url or image). The navigation bars in the foundation catalog are set up with 'name' display_type.

<table>
<thead>
<tr>
<th>display type</th>
<th>Link shown as</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Displays name only</td>
</tr>
</tbody>
</table>
5.7.7. image

Image URL for image or icon display_type.

5.7.8. image_prop

For image or icon display_type, this contains the HTML attributes for the HTML that will appear in the navbar, for example:

```html
<img src="image" alt="name" image_prop>name
```

5.7.9. banner_image

Example Data

```
promo_image.gif
```

This field is not related to banner ads. It is useful if you want to pass to your results page an image that is specific for the navbar entry (perhaps to display a banner above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' link_type), then this will add 'va=banner_image=banner_image' to the resulting search specification. This puts the contents of banner_image into the Values hash of your search results page. You can access it with [value banner_image] (see the value tag). You will have to modify the standard results page (or set up and specify your own) in order to display the image.

```
<IMG src="[value banner_image]" alt="[value banner_text]">
```

The foundation catalog does not implement banner_image in the preconfigured navigation bars.

5.7.10. banner_text

Example Data

```
This Is A Title For Hand Tools
```

This field is not related to banner ads. It is useful if you want to pass to your results page some text that is specific for the navbar entry (perhaps to display a title above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' link_type), then this will add 'va=banner_text=banner_text' to the resulting search specification. This puts the contents of banner_text into the Values hash of your search results page. You can access it with [value banner_text] (see the value tag). You will have to modify the standard results page (or set up and specify your own) in order to display the text.

The foundation catalog does not implement banner_text in the preconfigured navigation bars.
5.7.11. link_type

Example Data

- none
- external
- internal
- simple
- complex

Link type to create in the navigation bar.

<table>
<thead>
<tr>
<th>Link type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No link</td>
</tr>
<tr>
<td>external</td>
<td>External link. The HTML specified in url will go directly into the navigation bar.</td>
</tr>
<tr>
<td>internal</td>
<td>Internal link. This will be highlit if it is the current page. If you specify both a page and a form for the link, the url field should contain &quot;page form&quot;. See the Search Engine documentation for more detail on search forms.</td>
</tr>
<tr>
<td>simple</td>
<td>Allows you to specify an Interchange search with a few values. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail.</td>
</tr>
<tr>
<td>complex</td>
<td>Allows you to fully specify an Interchange search specification. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail if you need to use these.</td>
</tr>
</tbody>
</table>

5.7.12. url

Target URL (external link or internal page/search specification). See link_type. The foundation catalog navigation bars are not set up with link types that use the url field.

5.7.13. tab

Database table file to use with 'simple' link_type (searchspec fi=tab). The foundation catalog navigation bars are not set up with link types that use the tab field.

5.7.14. page

Results page to use with 'simple' link_type (searchspec sp=page). The foundation catalog navigation bars are not set up with link types that use the page field.

5.7.15. search

Search spec used with 'complex' link_type. See the Search Engine documentation for more detail on search forms. The foundation catalog navigation bars are not set up with link types that use the search field.

5.7.16. selector

The selector that is used to scan the products table for products in the category. Used with 'simple' link_type. The foundation catalog navigation bars are not set up with link types that use the selector field.
5.7.17. link_template

Overrides the usual HTML link template for navbar entries. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file if you need to modify link templates.

The foundation catalog navigation bars are not set up with link types that use the link_template field.

5.8. banner

cat_root/products/banner.txt

The banner ad table. The foundation catalog does not implement any banner ads with this table.

You do not need to use this table to display ads served by third parties (for example, doubleclick). Since most banner ads on the internet are served by third parties and are not managed by your catalog, you probably will not need to set up banners here unless you do your own advertising.

See Banner/Ad rotation in the template documentation for a detailed description of the columns and content of the banner table. Also, see the banner tag documentation.

**Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Key for the item. If the banners are not weighted, this should be a category−specific code.</td>
</tr>
<tr>
<td>category</td>
<td>Category for set of weighted banners</td>
</tr>
<tr>
<td>weight</td>
<td>Display frequency weight for weighted banner</td>
</tr>
<tr>
<td>rotate</td>
<td>Boolean: parse banner field for banners to rotate if true (1)</td>
</tr>
<tr>
<td>banner</td>
<td>Banner name or list of banners to rotate</td>
</tr>
</tbody>
</table>

5.8.1. code

Example Data

MyBanner
MyBanner2
MyBanner3
default

See Banner/Ad rotation.

5.8.2. category

Example Data

BannerCat1

See Banner/Ad rotation.
5.8.3. weight

Example Data

1
2
7

See Banner/Ad rotation.

5.8.4. rotate

Boolean value. If true (1), rotates banners listed in banner. See Banner/Ad rotation.

5.8.5. banner

Example Data

- Default banner 1
- Default banner 2
- Default banner 3
- First MyBanner
- Second MyBanner
- Third MyBanner

See Banner/Ad rotation.

5.9. cat

cat_root/products/cat.txt

This table contains properties of product categories. Notice the similarity to the area table, since both the area and cat tables supply data for building links to results pages.

Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Unique key</td>
</tr>
<tr>
<td>sel</td>
<td>Space–delimited list of foreign keys into area table</td>
</tr>
<tr>
<td>name</td>
<td>Category name</td>
</tr>
<tr>
<td>which_page</td>
<td>Page class in which the category may appear</td>
</tr>
<tr>
<td>sort</td>
<td>Sorting prefix for entry (preempts standard alphanumeric sort)</td>
</tr>
<tr>
<td>display_type</td>
<td>How to label the category links (name, icon, url or image)</td>
</tr>
<tr>
<td>image</td>
<td>Image URL (if appropriate)</td>
</tr>
<tr>
<td>image_prop</td>
<td>HTML attributes for output &lt;img&gt; tag (if appropriate)</td>
</tr>
<tr>
<td>banner_image</td>
<td>Image name for use in target page</td>
</tr>
<tr>
<td>banner_text</td>
<td>Text for use in target page</td>
</tr>
<tr>
<td>link_type</td>
<td>Type of links in the navbar (external, internal, simple, complex)</td>
</tr>
</tbody>
</table>
Foundation Store

<table>
<thead>
<tr>
<th>url</th>
<th>Target for internal or external link_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tab</td>
<td>Database table file to use with 'simple' link_type</td>
</tr>
<tr>
<td>page</td>
<td>Results page to use with 'simple' link_type</td>
</tr>
<tr>
<td>search</td>
<td>Search spec used with 'complex' link_type</td>
</tr>
<tr>
<td>selector</td>
<td>The selector used to scan the products table for products in the category</td>
</tr>
<tr>
<td>link_template</td>
<td>Overrides template used for building links</td>
</tr>
</tbody>
</table>

5.9.1. code

Example Data

```
1
4
5
```

Unique key.

5.9.2. sel

Example Data

```
6
8 9
9
```

Space-delimited list of foreign key(s) into area table. The category will appear in each navbar section (defined by a row in the area table) where the key from cat.sel matches the area.code.

For example, the foundation catalog (tools) places Gift Certificates in more than one category of the left navbar.

5.9.3. name

Example Data

```
Breathing Protection
Eye Protection
Gift Certificate
Picks & Hatchets
Pliers
Rulers
Sandpaper
Toolboxes
```

Category name for display.

5.9.4. which_page

The page class. When building links, you can select categories matching a page class. This means you could set up your catalog to show a different list of links on page 'foo.html' than on page 'bar.html'.

5.9. cat
5.9.5. sort

Example Data

01
03

Lexographic (alphanumeric) sorting prefix. Notice use of '03' rather than '3', which would sort after '13'. You can use this to control the order of the categories in a list of links.

5.9.6. display_type

Example Data

name
cicon
turl
image

What to use for the labels in the navigation bar (for example, name, icon, url or image). The links in the foundation catalog are set up with 'name' display_type.

<table>
<thead>
<tr>
<th>display type</th>
<th>Link shown as</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Displays name only</td>
</tr>
<tr>
<td>icon</td>
<td>Displays name and specified image</td>
</tr>
<tr>
<td>image</td>
<td>Displays image only</td>
</tr>
<tr>
<td>url</td>
<td>Displays link</td>
</tr>
</tbody>
</table>

5.9.7. image

Image URL for image or icon display_type.

5.9.8. image_prop

For image or icon display_type, this contains the HTML <img ...> tag attributes for the links, for example:

```
<img src="image" alt="name" image_prop>name
```

5.9.9. banner_image

Example Data

```
promo_image.gif
```

This field is not related to banner ads. It is useful if you want to pass to your results page an image that is specific for the navbar entry (perhaps to display a banner above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' link_type), then this will add 'va=banner_image=banner_image' to the resulting search specification. This puts the contents of banner_image into the Values hash in your search results page. You can access it with [value}
5.9. banner_image

(see the value tag). You will have to modify the standard results page (or set up and specify your own) in order to display the image. For example, you might include the following in your results page:

```html
<IMG src="[value banner_image]" alt="[value banner_text]">
```

5.9.10. banner_text

This field is not related to banner ads. It is useful if you want to pass to your results page some text that is specific for the navbar entry (perhaps to display a title above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' link_type), then this will add 'va=banner_text=banner_text' to the resulting search specification. This puts the contents of banner_text into the Values hash in your search results page. You can access it with [value banner_text] (see the value tag). You will have to modify the standard results page (or set up and specify your own) in order to display the text.

5.9.11. link_type

Example Data

```
none
external
internal
simple
complex
```

The link type to create.

<table>
<thead>
<tr>
<th>Link type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No link</td>
</tr>
<tr>
<td>external</td>
<td>External link. The HTML specified in url will go directly into the link.</td>
</tr>
<tr>
<td>internal</td>
<td>Internal link. This will be highlighted if it is the current page. If you specify both a page and a form for the link, the url field should contain &quot;page form&quot;. See the Search Engine documentation for more detail on search forms.</td>
</tr>
<tr>
<td>simple</td>
<td>Allows you to specify an Interchange search with a few values. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail.</td>
</tr>
<tr>
<td>complex</td>
<td>Allows you to fully specify an Interchange search specification. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail if you need to use these.</td>
</tr>
</tbody>
</table>

5.9.12. url

Target URL (external link or internal page/search specification). See link_types above.

5.9.13. tab

Example Data

```
products
```

5.9. cat
Database table file to use with 'simple' **link_type** (searchspec fi=tab).

### 5.9.14. page

Example Data

```
swap_results
```

Results page to use with 'simple' **link_type** (searchspec sp=page).

### 5.9.15. search

Example Data

```
fi=merchandising^Msf=featured^Mse=new
fi=merchandising^Msf=featured^Mse=special^Msu=yes
```

Search spec used with 'complex' **link_type**. See the Search Engine documentation for more detail on search forms.

**Note:** The `^M` delimiters in the sample data represents a carriage return character (Control−M, or hexadecimal 0x0d).

### 5.9.16. selector

Example Data

```
category=Breathing Protection
category=Eye Protection
category=Gift Certificate
category=Picks & Hatchets
category=Pliers
category=Rulers
category=Sandpaper
category=Toolboxes
```

The element that is used to scan the `products` table for products in the category. Used with 'simple' **link_type**.

### 5.9.17. link_template

Overrides the usual HTML link template for navbar entries. See the **bar_link** subroutine in the `/dist/catalog_before.cfg` configuration file if you need to modify link templates.

### 5.10. country

A list of countries used to build select boxes, calculate tax, and set shipping mode choices based on the user's country.

<table>
<thead>
<tr>
<th>code</th>
<th>Country code</th>
<th>sorder</th>
<th>Used to determine sort order</th>
</tr>
</thead>
</table>

50
Foundation Store

region        An enumerator to group based on region
selector      The code that should be used (same as code)
shipmodes     Shipping modes to be presented for that country
name          Textual name of the country
iso           ISO three letter code for the country
isonum        ISO numeric code for the country
tax           Tax information (may redirect to a state table)

5.11. downloadable

This table controls downloadable products. The Marketing Reports data set for the foundation catalog demonstrates downloadable products. List a product's **sku** in this table if you want to deliver it through a download. A customer can then download the file specified in the **dl_location** field after checkout.

For reference, see the implementation in the following files:

- catalog_root/pages/deliver.html
- catalog_root/etc/receipt.html
- catalog_root/pages/query/order_detail.html

**Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sku</td>
<td>Unique key, matches <strong>product.sku</strong></td>
</tr>
<tr>
<td>dl_location</td>
<td>Location of downloadable file</td>
</tr>
<tr>
<td>dl_type</td>
<td>MIME type of downloadable file</td>
</tr>
</tbody>
</table>

5.11.1. **sku**

Example Data

Example Data from the Marketing Reports data set:

```
00352as
22083da
49503cg
59330rt
59402fw
73358ee
83491vp
90773sh
```

This is the unique key for this table that is also the common key into the products table.

5.11.2. **dl_location**

Example Data from `reports` catalog

```
download/00352as.pdf
download/22083da.pdf
download/49503cg.pdf
download/59330rt.pdf
download/59402fw.pdf
```
File location of downloadable product.

5.11.3. dl_type

Example Data from 'reports' catalog

application/pdf

MIME type of downloadable content.

5.12. files

A database where files (pages, etc.) can be kept instead of in the Unix filesystem. Not normally used in foundation.

5.13. inventory.txt

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sku</td>
<td>Quantity info</td>
</tr>
<tr>
<td>quantity</td>
<td>Gets decremented after each sale.</td>
</tr>
<tr>
<td>stock_message</td>
<td>The usual shipping time of the product. Example choices:</td>
</tr>
<tr>
<td></td>
<td>In stock</td>
</tr>
<tr>
<td></td>
<td>Ships in 3-5 days</td>
</tr>
<tr>
<td></td>
<td>Ships in 4-6 weeks</td>
</tr>
<tr>
<td></td>
<td>Special order</td>
</tr>
<tr>
<td>account</td>
<td>Sales account for accounting links</td>
</tr>
<tr>
<td>cogs_account</td>
<td>Inventory account for accounting links</td>
</tr>
</tbody>
</table>

5.14. locale.txt

<table>
<thead>
<tr>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_US</td>
</tr>
<tr>
<td>de_DE</td>
</tr>
<tr>
<td>fr_FR</td>
</tr>
</tbody>
</table>

Used to develop localization. Can be used in conjunction with other types of locale information. See Internationalization.

5.15. merchandising.txt

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sku</td>
<td></td>
</tr>
<tr>
<td>featured</td>
<td></td>
</tr>
<tr>
<td>banner_text</td>
<td></td>
</tr>
<tr>
<td>banner_image</td>
<td></td>
</tr>
<tr>
<td>blurb_begin</td>
<td></td>
</tr>
<tr>
<td>blurb_end</td>
<td>Closer (end text for feature display)</td>
</tr>
<tr>
<td>timed_promotion</td>
<td></td>
</tr>
</tbody>
</table>
5.16. mv_metadata

See the following sections in the icadvanced catalog for more information:

- display tag and mv_metadata
- mv_metadata.asc

5.17. options

This table contains data for implementing simple, matrix and modular options.

Simple options are options that a customer can combine arbitrarily, such as size and color. The selected options might affect price. See the accessories tag for more detail on option values for simple options.

Matrix options are preconfigured combinations of options. For example, if you sell titanium and carbon–fiber bike frames, but offer only certain combinations of frame material and color, your checkout page might include a select box with only the following entries:

- Silver Titanium: $1672
- Black Titanium: $1672
- Red Titanium: $1674
- Black Carbon Fiber: $1290
- Yellow Flame Carbon Fiber: $1300

Note that there is no Yellow Flame Titanium offering, for example.

Modular options are like a structured bill of materials, where one product is a master item and other products are subitems for that master item. The subitems can also be master items to subitems at a lower level. In addition, subitems may be designated as 'phantom', which means that they are placeholders in the hierarchy of the structured bill of materials with their own subitems, but are not actual items themselves.

The foundation catalog with the computer data set uses modular options.

For more information, see the following pages and components in the foundation catalog:

- cat_root/pages/flypage.html
- cat_root/templates/components/modular_buy
- cat_root/templates/components/modular_update

Note: Subsequent foundation catalog releases may place simple, matrix, and modular option types in separate tables.
### Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Unique ID for the product option</td>
</tr>
<tr>
<td>o_master</td>
<td>SKU of the master item for the option</td>
</tr>
<tr>
<td>sku</td>
<td>SKU for the option (foreign key into products table)</td>
</tr>
<tr>
<td>o_group</td>
<td>Product grouping code</td>
</tr>
<tr>
<td>o_sort</td>
<td>Sorting prefix for list display</td>
</tr>
<tr>
<td>phantom</td>
<td>Boolean — Item is a phantom placeholder (as in structured bill of materials) with suboptions.</td>
</tr>
<tr>
<td>o_enable</td>
<td>Boolean — enables suboptions for the option</td>
</tr>
<tr>
<td>o_matrix</td>
<td>Matrix–type option (preconfigured combinations of attributes)</td>
</tr>
<tr>
<td>o_modular</td>
<td>Modular–type option (master/subitem relationship like modular bill of materials)</td>
</tr>
<tr>
<td>o_default</td>
<td>Default selection for the option group or suboption for a phantom option</td>
</tr>
<tr>
<td>o_label</td>
<td>Short name for option display</td>
</tr>
<tr>
<td>o_value</td>
<td>Simple option values (in Interchange option format)</td>
</tr>
<tr>
<td>o_widget</td>
<td>The HTML widget to use for displaying the option group</td>
</tr>
<tr>
<td>o_footer</td>
<td>Not used in foundation catalog</td>
</tr>
<tr>
<td>o_header</td>
<td>Not used in foundation catalog</td>
</tr>
<tr>
<td>o_height</td>
<td>Height of widget (if applicable)</td>
</tr>
<tr>
<td>o_width</td>
<td>Width of widget</td>
</tr>
<tr>
<td>description</td>
<td>Option/Variant description (for description in display)</td>
</tr>
<tr>
<td>price</td>
<td>Price of this option/variant</td>
</tr>
<tr>
<td>wholesale</td>
<td>Dealer price of this option/variant</td>
</tr>
<tr>
<td>differential</td>
<td>Differential to add to the base item price when using a phantom bill of materials</td>
</tr>
<tr>
<td>weight</td>
<td>Weight difference with this option/variant (for shipping)</td>
</tr>
<tr>
<td>volume</td>
<td>Volume difference with this option/variant</td>
</tr>
<tr>
<td>mv_shipmode</td>
<td>No Description</td>
</tr>
<tr>
<td>o_exclude</td>
<td>Option groups to exclude (trumped by o_include). Modular only.</td>
</tr>
<tr>
<td>o_include</td>
<td>Option groups to include (trumps o_exclude). Modular only.</td>
</tr>
</tbody>
</table>

#### 5.17.1. code

Example Data

- 1002
- 1003
- 1004
- 1005

Unique ID for the option.
5.17.2. o_master

Example Data

00010
999000
999001
999002

SKU of the master item for the option. The master item is one level up in the modular hierarchy, and must be one of the following:

- An item in the products table (matching `products.sku`)
- Another option in the options table (matching `options.sku`)
- A phantom item in the options table.

If an option has a master item, then a customer can not choose that option without having previously selected the master item.

The price for a master item is the sum of the master item's price and the price for each of the subitems. Because the subitems are recursively defined, the top-level item reflects the top level price plus the price of all selected options.

5.17.3. sku

Example Data

00010
999000
7000015
7000030

The sku for the item or option. This may not be unique for matrix options or if an option that belongs to multiple `o_master`s is listed for each master.

5.17.4. o_group

Example Data

A
B
C
I

Product group (scanned to see whether it applies to this product or not)

5.17.5. o_sort

Example Data

01
02
Sorting prefix for listing order of options.

### 5.17.6. phantom

Modular options only.

Boolean — if true (1), then this is a phantom item acting as a placeholder for other items rather than an actual product. The item's `sku` will not match an entry in the products table, though the `o_master` will match either the `sku` of another phantom item in the `options` table or the `sku` of an item in the `products` table.

### 5.17.7. o_enable

Boolean — Enables subitems for this item or option. Note that an option with `o_enable` false may itself still be a subitem for an option or item above it.

### 5.17.8. o_matrix

Boolean. Set true (1) for matrix–type options. See the options table in the tools data set for examples of matrix options. Matrix options that are part of a set have the same value for `options.sku`.

### 5.17.9. o_modular

Specifies a modular option. See main heading for description of modular options.

### 5.17.10. o_default

Example Data

```
1
11002
7000062
7000087
```

Selects the default option for a group.

### 5.17.11. o_label

Example Data

```
Add a second hard drive
Case Color
Case color
Case style
Include tapes
Red
```
This is the short name for option display.

### 5.17.12. o_value

Example Data

```
1=One 8GB tape, \r2=Two 8GB tapes, \r==None*
a=One 8GB tape, \r\r==None*
baby=Baby Tower, \r\rmid=Mid−tower, \r\rfull=Full Tower
\ne=Baby tower, \r\rmid=Mid−tower, \r\rfull=Full tower
\n\nred=Passion Red, \r\r\r=Electric Blue, \r\rgreen=Sea Green, \r\rgrey=S...
\n\nred=Passion Red, \r\r\r=Electric Blue, \r\rgrey=Honey Yellow
```

This is an Interchange value set for a simple option. It is typically a comma−delimited list of labels and values with "*" indicating the default value. See the `accessories` tag for more detail.

Note that the "\r" characters in the above example represent carriage returns in the actual data ("\r" in perl, or Ctrl−M, or hexadecimal 0D), and the ... indicates a line too long to show.

### 5.17.13. o_widget

Example Data

```
select
```

This determines the HTML Widget type (e.g., a select box). For example, the `options` tag uses this entry when building HTML widgets in a page. See also the `accessories` tag for available widgets.

### 5.17.14. o_footer

Example Data

Allowed Values

### 5.17.15. o_header

Example Data

Allowed Values

### 5.17.16. o_height

This allows you to set the height of the HTML widget, if appropriate.

### 5.17.17. o_width

This allows you to set the width of the HTML widget, if appropriate.
5.17.18. description

Example Data

ATX Mid Tower−Grey (3)5.25 (2)3.5 & (1)3.5 Hidden
Enlight ATX Desktop Case (2)5.25 & (2)3.5
Enlight ATX Tower Case (4) 5.25 & (2)3.5
Micro ATX Tower − Honey Yellow
Micro ATX Tower − Moody Blue
Micro ATX Tower − Rage Red
Micro ATX Tower − Smoky Grey
Super Tower Case (6)5.25 & (3)3.5

Longer description to show when displaying the options.

5.17.19. price

Example Data

0.00
10
20
29
75

This sets the retail price of the option.

5.17.20. wholesale

Example Data

13
40.00

This sets the dealer price of the option.

5.17.21. differential

Example Data

−209
−40
−79

The phantom bill of materials for an option group can have a differential, which is an amount to add to the base price of the master product to get to a new base price that accommodates the phantom bill of materials. Note that the differential can be negative.

For example, in the computer data set of the foundation catalog, SKU 00011 in the products table is an $849.95 pre−configured Athlon 800MHz computer that includes a 17" monitor (in this case, SKU 7000087 in the products table).
The monitor by itself would otherwise have cost $209. It is much more convenient if you can use the same option part number and price for each item. To do this, you need a phantom option (in this case, SKU 999105 in the options table only) with a differential of −209 and the available monitors as suboptions. When you include the phantom option in the bill of materials for the computer (SKU 00011), the $−209 differential adjustment makes the price work out properly.

For instance, suppose that a $499 computer is configured as follows:

- 500 MHz Athlon -- $499
- 32 MB SDRAM -- ZERO
- 10 GB disk -- ZERO
- TOTAL -- $499

Suppose it costs $90 to upgrade the base computer to 128M of RAM and $150 for a 30 GB hard disk.

If you also sell an 128MB 800 MHz $899 computer, and the customer upgrades to the 30 GB hard disk,

- 800 MHz Athlon -- $899
- (memory differential) -- $−90
- 128 MB RAM -- $90
- 30 GB disk -- $150
- TOTAL -- $1039

If you did not have the differential, you would need a different option part number for each item make the number come out right.

With the differential, you can use the same part number for 128MB RAM no matter what the base part is. The price is always $90 — there is just a −90 differential when ordered with the 800MHz Athlon, making the effective price zero.

5.17.22. weight

Example Data

5

Shipping weight of the option. Interchange uses this to calculate shipping cost.

5.17.23. volume

Volume added by the option.

5.17.24. mv_shipmode

No Description

5.17.25. o_exclude

Modular options only.

5.17. options
Lists the option groups to exclude once the include has been done. Takes the form of a number of wildcard atoms.

5.17.26. *o_include*

Modular options only.

Lists the option groups to include with your item. Takes the form of a number of wildcard atoms.

5.18. *order_returns.txt*

```
code
order_number
session
username
rma_number
nitems
total
return_date
update_date
```

5.19. *orderline.txt*

Every line item that is actually ordered is detailed in this table. The order as a whole is one record in the transactions table.

See the page `query/check_orders.html` for how it can be used. See etc/report for how to add to it.

```
code
store_id
order_number
session
username
shipmode
sku
quantity
price
subtotal
shipping
taxable
mv_mi
mv_si
size
color
options
order_date
update_date
status
    pending = Pending
    shipped = Shipped
    backorder = Back ordered
    credit = Waiting for credit check
    canceled = Cancelled
parent
affiliate
```
5.20. pricing

This database works in conjunction with the CommonAdjust directive to allow quantity pricing for one product or for a group of products (sometimes known as mix−and−match). The fields q2, q5, q10, etc. are for the quantity levels; the price_group field selects the mix−and−match category for the product.

### Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sku</td>
<td>Unique key, shared with products table</td>
</tr>
<tr>
<td>price_group</td>
<td>Mix−and−match category</td>
</tr>
<tr>
<td>q2</td>
<td>Retail, 2 or more</td>
</tr>
<tr>
<td>q5</td>
<td>Retail, 5 or more</td>
</tr>
<tr>
<td>q10</td>
<td>Retail, 10 or more</td>
</tr>
<tr>
<td>q25</td>
<td>Retail, 25 or more</td>
</tr>
<tr>
<td>q100</td>
<td>Retail, 100 or more</td>
</tr>
<tr>
<td>w2</td>
<td>Wholesale, 2 or more</td>
</tr>
<tr>
<td>w5</td>
<td>Wholesale, 5 or more</td>
</tr>
<tr>
<td>w10</td>
<td>Wholesale, 10 or more</td>
</tr>
<tr>
<td>w25</td>
<td>Wholesale, 25 or more</td>
</tr>
<tr>
<td>w100</td>
<td>Wholesale, 100 or more</td>
</tr>
</tbody>
</table>

5.20.1. sku

Example Data

```
os28004
os28006
os28057c
os28069
```

Unique key, matching the sku for an entry in products table.

5.20.2. price_group

Example Data

```
general
```

This field determines mix−and−match categories if you want to allow mix−and−match quantity pricing (i.e., where 5 of these plus 5 of those afford the q10 price for both these and those).
5.20.3. q2

If set, this will be the price per item when the order quantity is 2 or greater.

5.20.4. q5

If set, this will be the price per item when the order quantity is 5 or greater.

5.20.5. q10

If set, this will be the price per item when the order quantity is 10 or greater.

5.20.6. q25

If set, this will be the price per item when the order quantity is 25 or greater.

5.20.7. q100

If set, this will be the price per item when the order quantity is 100 or greater.

5.20.8. w2

If set, this will be the dealer price per item when the order quantity is 2 or greater.

5.20.9. w5

If set, this will be the dealer price per item when the order quantity is 5 or greater.

5.20.10. w10

If set, this will be the dealer price per item when the order quantity is 10 or greater.

5.20.11. w25

If set, this will be the dealer price per item when the order quantity is 25 or greater.

5.20.12. w100

If set, this will be the dealer price per item when the order quantity is 100 or greater.

5.21. products

This is the main table for product data. See also 'The Product Database' section in the database documentation.

The sku is also the master key in many of the related tables.

Fields
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sku</td>
<td>Unique product ID</td>
</tr>
<tr>
<td>description</td>
<td>Short description for list display</td>
</tr>
<tr>
<td>title</td>
<td>Full title for book, CD, artwork, etc.</td>
</tr>
<tr>
<td>template_page</td>
<td>Not used in foundation catalog. No description.</td>
</tr>
<tr>
<td>comment</td>
<td>Longer description for item display (e.g., flypage.html)</td>
</tr>
<tr>
<td>thumb</td>
<td>Thumbnail image</td>
</tr>
<tr>
<td>image</td>
<td>Regular-sized image</td>
</tr>
<tr>
<td>price</td>
<td>Retail quantity one price</td>
</tr>
<tr>
<td>wholesale</td>
<td>Dealer minimum quantity price</td>
</tr>
<tr>
<td>prod_group</td>
<td>Product supercategory</td>
</tr>
<tr>
<td>category</td>
<td>Product category</td>
</tr>
<tr>
<td>nontaxable</td>
<td>Boolean. Set true (1) if nontaxable</td>
</tr>
<tr>
<td>weight</td>
<td>Weight in your units. Should match shipping table.</td>
</tr>
<tr>
<td>size</td>
<td>List of options used with accessories tag.</td>
</tr>
<tr>
<td>color</td>
<td>List of options used with accessories tag.</td>
</tr>
<tr>
<td>gift_cert</td>
<td>Boolean. Set true (1) if this is a gift certificate.</td>
</tr>
<tr>
<td>related</td>
<td>Deprecated in favor of merchandising.upsell_to</td>
</tr>
<tr>
<td>featured</td>
<td>Deprecated. Use merchandising table.</td>
</tr>
<tr>
<td>inactive</td>
<td>Boolean. Set true (1) to inactivate a product</td>
</tr>
<tr>
<td>url</td>
<td>Not Documented</td>
</tr>
</tbody>
</table>

### 5.21.1. sku

Example Data

gift_cert
os28004
os28006
os28057c

Unique identifier for the product. You should use only characters of the class A−Z a−z 0−9 _ − (i.e., matching the regular expression, '[-A-Za-z0-9_]+'). Although Interchange itself does not impose this restriction, you may have problems with SQL databases, file systems, and URL encoding if you use other characters. For example, a slash (/) can interfere with URLs and filenames, a colon (:) can interfere with database representations (or file names on some operating systems), i<etc.>

### 5.21.2. description

Example Data

Brush Set
Disposable Brush Set
Ergo Roller
Gift Certificate
Painters Brush Set
Painters Ladder
Spackling Knife
Trim Brush

A short description for the product that is used for displaying in the shopping cart, receipt, and order report.

5.21.3. title

Example Data

Brush Set
Disposable Brush Set
Ergo Roller
Gift Certificate
Painters Brush Set
Painters Ladder
Spackling Knife
Trim Brush

This column is not used in the foundation catalog. Previously used in the Art store (simple) demo for a painting title. You probably want to use description instead.

You should modify the products and other tables to suit your catalog's requirements. You might use this field if you want to show titles for books, music, or other titled merchandise. If you do not use a title that is distinct from the short description, then you probably do not need this column in the table at all.

5.21.4. template_page

Not used in foundation catalog.

No Description.

5.21.5. comment

Example Data

A must have for all painters! This spackling knife is ergon...
Enjoy the perfect feel and swing of our line of hammers. Thi...
This set includes 2" and 3" trim brushes and our ergonomical...
This set of disposable foam brushes is ideal for any stainin...

This is the field for a long description of the product. If you are using an Interchange text/gdbm database, the field size is unlimited; if using another type of database, the length will be dependent on the field type selected. If you are using a SQL database, see the appropriate cat_root/dbconf subdirectory for a place to set COLUMN_DEF values. See also the database documentation, 'Importing from an ASCII File', for details about defaults for columns that you do not define.

5.21.6. thumb

Example Data

gift_certificate.gif
os28004_b.gif
This is the filename for a small (thumbnail) image of the product.

5.21.7. image

Example Data

    gift_certificate_large.gif
    os28004.gif
    os28005.gif
    os28006.gif

This is the filename for a regular-sized image of the product, as it should appear in an HTML `<img src="image">` tag. You do not need to specify the path if the image files are in the usual Interchange image directory.

5.21.8. price

Example Data

    1.00
    12.99
    14.99
    9.99

The quantity—one price of the product. See the `wholesale` field and the `price` table for dealer and quantity pricing.

5.21.9. wholesale

Example Data

    1
    10
    11
    12

This is the minimum dealer price for the item. For quantity pricing, see the `price` table.

5.21.10. prod_group

Example Data

    Hand Tools
    Hardware
    Ladders
    Measuring Tools
    Miscellaneous
    Painting Supplies
    Safety Equipment
    Tool Storage

5.21. products
Product group (supercategory). This indicates the grouping of product categories, for example in the navigation bars created from the area table (note the match with the name data in the area table).

5.21.11. category

Example Data

<table>
<thead>
<tr>
<th>Brushes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gift Certificate</td>
</tr>
<tr>
<td>Hammers</td>
</tr>
<tr>
<td>Ladders</td>
</tr>
<tr>
<td>Nails</td>
</tr>
<tr>
<td>Paintbrushes</td>
</tr>
<tr>
<td>Putty Knives</td>
</tr>
<tr>
<td>Rollers</td>
</tr>
</tbody>
</table>

This is the category the product should appear in when you select a list. You can put a product in more than one category, but you may need to accommodate this in display and banner headings. Embedded perl is helpful for this.

5.21.12. nontaxable

Boolean value. If true (1), the sales tax calculation for an order will not include the cost of the product. See also the salestax tag.

5.21.13. weight

Example Data

1
2
3

This is a numeric value of the weight used for determining shipping costs (with UPS, for example). In the US, this is typically the weight in pounds in order to match the UPS, Fed Ex and other standard shipping tables.

5.21.14. size

Example Data

1", 2", 3"
1', 1.5'
1/4", 1/2", 3/4", 1", 2", 3"
10oz, 15oz, 20 oz
2" set
standard, metric

This is where the old Construct Something demo store kept the 'size' options for a product. The foundation catalog now uses the options table instead to handle product options (also sometimes called product attributes).
The **accessories** tag can build HTML widgets from the comma–delimited list of product options. You can use a delimiter other than comma (if compatible with the table) as long as you also set the **delimiter** in the **accessories** tag.

You probably do not need this field if you use the options table (for example, if you are building from the foundation catalog).

**5.21.15. color**

Another product option column. No longer used in the foundation catalog. See **size** above for description.

**5.21.16. gift_cert**

Boolean value. If true (1), specifies that this product is a gift certificate. See also the **gift_certs** table.

**5.21.17. related**

Used for displaying "upsells," opportunities to purchase an additional item when this one is purchased. Contains a comma–separated list of SKUs to be offered.

The foundation catalog now instead uses the **upsell_to** field of the **merchandising** table for upselling.

**5.21.18. featured**

Deprecated in favor of the **merchandising** table.

**5.21.19. inactive**

If true (1), renders the product inactive (i.e., it will not appear in the catalog).

**5.21.20. url**

Not Documented

**5.22. route.txt**

```text
  code
  attach
  continue
  commit
  commit_tables
  counter
  credit_card
  cyber_mode
  email
  empty
  encrypt
  encrypt_program
  errors_to
  increment
  inline_profile
```
5.23. salestax.asc

5.24. shipping.asc

Shipping methods table

5.25. state.txt

State/territory/county information

5.26. transactions.txt

Each individual customer order has an entry in this table. The line items are not entered here, but in the orderline table.

See the page query/check_orders.html for how it can be used. See etc/report for how to add to it.
The user database used for maintaining customer address information, account information, preferences, and more. See icdatabase for more information.
lname
address1
address2
address3
city
state
zip
Postcode
country
Country
phone_day
mv_shipmode
b_nickname
b_fname
b_lname
b_address1
b_address2
b_address3
b_city
b_state
b_zip
b_country
b_phone
Billing Phone
mv_credit_card_type
mv_credit_card_exp_month
mv_credit_card_exp_year
p_nickname
eemail
fax
phone_night
fax_order
Payment method:
(none) = Credit Card
1 = Fax or Mail
2 = Purchase order
3 = COD
address_book
accounts
preferences
carts
owner
file_acl
db_acl
order_numbers
eemail_copy
mail_list
Mailing lists the customer has joined:
offer = Special offers
newsletter = Newsletter
alert = Alerts and Recalls
upgrade = Upgrades
project_id
account_id
order_dest
credit_limit
inactive
dealer
Dealer:
(none) = No
1 = Yes
b_company
5.28. variable.txt

Configuration database

code
  Variable name
Variable
pref_group
  Preferences area
6. HTML Hypertext links

Normally, regular hypertext links are not used in Interchange pages. These kinds of links will not include the session ID. If the customer follows an external link back to the catalog, the list of products ordered so far will have been lost. The area tag is used to generate a hypertext link which includes a session ID.

Instead of:

   <A HREF="/cgi-bin/catlink/shirts">Shirts</A>

Use:

   <A HREF="[area shirts]">Shirts</A>
7. Images

Inline images are placed in Interchange pages in the normal fashion with `<IMG SRC="URL">`. But since Interchange pages are served by a CGI program, it will by default automatically rewrite relative image links as absolute ones based on the `ImageDir` and `ImageDirSecure` directives.
8. Browser Cookies

The Foundation store enables the Cookies directive so that users with cookie–capable browsers will retain session context. Then, standard HREF and Interchange page links can be intermixed without the fear of losing the shopping basket. Cookie capability is also required to use search caching, page caching, and statically generated pages. If the user's browser does not support cookies, the cache will be ignored.

If planning to use more than one host name within the same domain for naming purposes (perhaps a secure server and non–secure server), set the domain with the CookieDomain directive. This must contain at least two periods (.) as per the cookie specification, and must be located in the same server as the domain.
9. Dependencies in administration

In general, it's a good idea to leave fields empty if you don't want to use them, instead of removing them from the database altogether. That way nothing in the administration interface or the Foundation pages will break.
