



User and Developer Guide

Version 2003.06.16



Copyright 2001-2003 HostedStore Corporation. All Rights Reserved.

HostedStore Corporation retains all ownership rights to the software programs (referred to herein as "Software") offered by HostedStore Corporation and related documentation. Use of the Software and related documentation is governed by the license agreement accompanying the Software and applicable copyright law.

THIS DOCUMENTATION IS PROVIDED "AS-IS" WITHOUT WARRANTY OF ANY KIND. IN NO EVENT SHALL HOSTEDSTORE CORPORATION BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, ARISING FROM ANY ERROR IN THIS DOCUMENTATION.

HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
USA
www.hostedstore.com

June, 2003

Table of Contents

Copyright Notice	2
Table of Contents	3
Trademarks	12
End User License Agreement	13
About the Documentation	15
Resources Available	18
Lasso Professional 6 Compatibility	19
Introduction	20
Features Summary	21
Installation and Configuration	24
• Required Software	25
• Installing the HostedStore Files	26
• Creating the Database(s) and Tables	29
• Configuring Lasso Administration	30
• Loading HostedStore into Memory	33
• Store Configuration	35
• Initial Modifications	57
Upgrading	58
• 2002.12.01 to 2003.06.16	59
• 2002.11.15 to 2002.12.01	64
• 2002.07.15 to 2002.11.15	65
• 2002.07.01 to 2002.07.15	71
• Pre-2002.07.01 to 2002.07.01	74
The Customer's Perspective	75
• Storefront	76
• Home Page	77
• Product Search	79
• Product Detail	83
• Shopping Cart	88
• Checkout	93
• Help Center	100
• Email a Friend	102
• Product Review	104
• Wish Lists	108
• Account Administration	111
• Modify Password	113
• Newsletter	114
• Modify Billing Address	115
• Address Book	116
• View Orders	119
• Wish List	122
• Shopping Lists	127
• Purchased Files	129
• Logout	130
The Affiliate's Perspective	131
The Supplier's Perspective	133
• Order Queue	134
• Modify Ordered Item Status	135

<u>Store Administration</u>	137
• <u>Overview</u>	138
• <u>Logging In</u>	139
• <u>Admin User Administration</u>	140
• <u>Add New Admin User</u>	141
• <u>Find Admin User(s)</u>	144
• <u>Affiliate Administration</u>	146
• <u>Add New Affiliate</u>	148
• <u>Find Affiliate(s)</u>	151
• <u>Affiliate(s) Awaiting Payout</u>	153
• <u>Post Payout(s)</u>	155
• <u>Email Monthly Reports</u>	159
• <u>Example Uses for Affiliates</u>	160
• <u>Autoship Administration</u>	161
• <u>Add New Autoship Order</u>	162
• <u>Find Autoship Order(s)</u>	167
• <u>Edit an Autoship Order</u>	169
• <u>Manual Autoship Order Sweep</u>	172
• <u>Customer Administration</u>	175
• <u>Email Customer(s)</u>	176
• <u>Add New Customer</u>	179
• <u>Find Customer(s)</u>	181
• <u>Maintenance</u>	183
• <u>Modules</u>	187
• <u>Order Administration</u>	188
• <u>Add New Order</u>	190
• <u>Add New Return</u>	196
• <u>Find Order(s)</u>	202
• <u>Edit an Order</u>	204
• <u>Batch Settlement</u>	216
• <u>Export for Stone Edge Order Manager</u>	217
• <u>Price Group Administration</u>	220
• <u>Add New Price Group</u>	221
• <u>Find Price Group(s)</u>	223
• <u>Product Administration</u>	225
• <u>Add New Product</u>	227
• <u>Find Product(s)</u>	244
• <u>Adjust Pricing</u>	247
• <u>Adjust Inventory</u>	248
• <u>Generate Static Product Pages</u>	249
• <u>Global Price Adjustment</u>	250
• <u>File Upload</u>	251
• <u>Promotion Administration</u>	252
• <u>Add New Promotion</u>	253
• <u>Find Promotion(s)</u>	257
• <u>Example Uses for Promotions</u>	259
• <u>Reports</u>	260
• <u>Date Dependent Reports</u>	262
• <u>Sales Order Reports</u>	263
• <u>Orders by Day</u>	264
• <u>Orders by Hour</u>	265
• <u>Orders by Product</u>	266

• Orders by Category	267
• Orders by Department	268
• Orders by Manufacturer	269
• Orders by Customer	270
• Orders by New Customers	271
• Orders by Return Customers	272
• Orders by Billing Country	273
• Orders by Billing State/Province	274
• Orders by Billing Country, State/Province	275
• Orders by Shipping Country	276
• Orders by Shipping State/Province	277
• Orders by Shipping Country, State/Province	278
• Orders by Source	279
• Orders by Status	280
• Orders by Order	281
• Orders by Promotion	282
• Orders by Affiliate	283
• Gross Income Reports	284
• Gross Income by Day	285
• Gross Income by Hour	286
• Gross Income by Customer	287
• Gross Income by Customer, Itemized	288
• Gross Income by New Customers	289
• Gross Income by New Customers, Itemized	290
• Gross Income by Return Customers	291
• Gross Income by Return Customers, Itemized	292
• Gross Income by Billing Country	293
• Gross Income by Billing State/Province	294
• Gross Income by Billing Country, State/Province	295
• Gross Income by Shipping Country	296
• Gross Income by Shipping State/Province	297
• Gross Income by Shipping Country, State/Province	298
• Gross Income by Source	299
• Gross Income by Order	300
• Gross Income by Unsettled Order	301
• Gross Income by Settled Order	302
• Gross Income by Batch	303
• Gross Income by Promotion	304
• Gross Income by Affiliate	305
• Order Status Reports	306
• All Orders by Status	307
• Declined Orders, Itemized	308
• Canceled Orders, Itemized	309
• Abandoned Orders, Itemized	310
• Product Sales Reports	311
• Sales by Product	312
• Sales by Category	313
• Sales by Category, SubCategory	314
• Sales by Department	315
• Sales by Department, Product	316
• Sales by Manufacturer	317
• Sales by Manufacturer, Product	318

• Sales Tax Reports	319
• Sales Tax Collected by State/Province	320
• Discount Reports	321
• Promotions	322
• Admin Adjustments, Itemized	323
• Affiliate Reports	324
• Payout History by Affiliate, Itemized	325
• Sales by Affiliate	326
• Sales by Affiliate, Itemized	327
• Shipping Reports	328
• Drop Ship Orders by Supplier, Invoice	329
• Drop Ships Pending by Supplier, Invoice	330
• Date Independent Reports	331
• Image Library	332
• Missing Images	333
• Suggested Inventory Order	334
• Search to Sales Conversion	335
• Affiliate Referrals Conversion	336
• General Store Statistics	337
• Review Administration	338
• Find Review(s)	339
• Edit Review	341
• Store Status	343
• Supplier Administration	345
• Email Supplier(s)	346
• Add New Supplier	348
• Find Supplier(s)	351
Shipping and Taxation	353
• Shipping	354
• Taxation	357
Payment Processing	363
• Built-In Forms	364
• Credit Card Processing	366
• Offline Processing	367
• ACH Direct	368
• Authorize.Net ADC Direct Response	369
• Authorize.Net Advanced Integration Method (AIM)	371
• BluePay™	373
• DirectOne (Direct Interface)	375
• Echo	376
• eProcessing Network	377
• E-xact	378
• IO•NGATE™	379
• LinkPoint Basic	380
• NETbilling (DIRECT MODE)	382
• PayOffshore.com	383
• Planet Payment™ ADC Direct Response	385
• Plug'nPay™	387
• PSiGate	388
• SecurePay	390
• SkipJack	392
• Verisign® Payflow Link (SM)	393

• Verisign® Payflow Pro (SM)	395
• Check Processing	396
• Offline Processing	397
• ACH Direct	398
• Authorize.Net ADC Direct Response	399
• Authorize.Net Advanced Integration Method (AIM)	401
• LinkPoint Basic	403
• NETbilling (DIRECT MODE)	405
• Pay by Check	406
• Planet Payment™ ADC Direct Response	407
• Plug'nPay™	409
• Verisign® Payflow Link (SM)	410
• Verisign® Payflow Pro (SM)	412
• PayPal (IPN version 1.2)	413
• Custom Payment Processing	414
Modifying HostedStore	415
• HostedStore Custom Tags	416
• HS_ArrayRandomSort	417
• HS_FormatCurrency	418
• HS_FormatDate	419
• HS_FormatTime	420
• HS_GetFavoriteProducts	421
• HS_GetFeaturedProducts	422
• HS_GetRecommendedProducts	424
• HS_IsRequired	426
• HS_Percentage	427
• Select Lists	428
• States	429
• Countries	431
• Shipping Countries	437
• Credit Cards	442
• Months	443
• Years	444
• Cookies	445
• Store Templates	446
• Modifying Supplied Storefront Templates	447
• Creating Storefront Templates	448
• Creating Store Administration Templates	450
• Content Pages	451
• Navigation Pages	452
• Browse Products	453
• Template Footer	455
• Storefront	456
• Shopping Cart Display	457
• Shopping Cart Modify Item	459
• Recover Shopping Cart	460
• Save Shopping Cart	461
• Customer Information	462
• Customer Login Form	463
• Email a Friend	464
• Home Page	465
• Order Display	468

• Order Printable Invoice	470
• Order Result	472
• Modify Payment Method	474
• Product Detail	475
• Review Detail	479
• Review Form	481
• Review List	482
• Search Form	483
• Search List	485
• Intermediate Search Page	488
• Search SubCategories	490
• Shipping Info	491
• Modify Shipping Info	492
• Store Closed	493
• Wish List Detail	494
• Wish List List	495
• Wish List Login Form	496
• Wish List Search Form	497
• Customer Administration	498
• Address Book Entry	499
• Address Book List	500
• Modify Billing Address	501
• EAD Files List	502
• Login Form	503
• Main Menu	504
• Newsletter Subscription	505
• Order Detail	506
• Order List	508
• Change Password	509
• Shopping List Detail	510
• Shopping List List	511
• Email Wish List	512
• Wish List Menu	513
• Wish List Preferences	514
• Wish List Purchases	515
• Wish List Items	516
• Store Administration	517
• Module Menu	518
• Mailing Invoice	519
• Packing Invoice	521
• Pick List	523
• Affiliate Administration	525
• Login Form	526
• Orders List	527
• Supplier Administration	528
• Template	529
• Help Center	530
• Customer Service Inquiry Form	531
• Help Center Menu	532
• Other Pages	533
• Email Messages	534
• Storefront	535

• Customer Order Approval	536
• Email a Friend	537
• Sales Membership Purchase	538
• Customer New Account	539
• Forgotten Password	540
• Shopping Cart Quote	541
• Sales Order Approval	542
• Newsletter Subscription	543
• Customer Administration	544
• Cancel Order	545
• Forgotten Password	546
• Newsletter Subscription	547
• Newsletter Unsubscribe	548
• Email Wish List	549
• Store Administration	550
• Email Status Change	551
• Out of Stock Notification	553
• Newsletter Subscription	554
• Newsletter Unsubscribe	555
• Product In Stock Notification	556
• Affiliate Administration	557
• Forgotten Password	558
• Supplier Administration	559
• Customer Order Status Notification	560
• Store Order Status Notification	561
• Help Center	562
• Customer Service Inquiry	563
HostedStore Hooks	564
• How Hook Points Work	565
• Affiliate Administration Hooks	566
• affiliate/index.lasso	567
• Customer Administration Hooks	569
• admincustomer/address.lasso	570
• admincustomer/admin.lasso	572
• admincustomer/cancelorder.lasso	574
• admincustomer/files.lasso	575
• admincustomer/index.lasso	576
• admincustomer/logout.lasso	577
• admincustomer/shoppinglist.lasso	578
• admincustomer/wishlist.lasso	580
• Store Administration Hooks	582
• adminstore/adminuser.lasso	583
• adminstore/affiliate.lasso	584
• adminstore/affiliatepayout.lasso	585
• adminstore/affiliatepayouthistory.lasso	586
• adminstore/application.lasso	587
• adminstore/autoship.lasso	588
• adminstore/autoshipsweep.lasso	589
• adminstore/batch.lasso	590
• adminstore/customer.lasso	591
• adminstore/image.lasso	592
• adminstore/indexpage.lasso	593

• adminstore/inventory.lasso	594
• adminstore/maintenance.lasso	595
• adminstore/manualcheckout.lasso	596
• adminstore/manualorder.lasso	597
• adminstore/manualreturn_checkout.lasso	598
• adminstore/manualreturn.lasso	599
• adminstore/menu.lasso	600
• adminstore/order.lasso	601
• adminstore/ordermanager.lasso	602
• adminstore/price.lasso	603
• adminstore/priceglobaladjust.lasso	604
• adminstore/pricegroup.lasso	605
• adminstore/product.lasso	606
• adminstore/promotion.lasso	607
• adminstore/review.lasso	608
• adminstore/supplier.lasso	609
• adminstore/upload.lasso	610
• Storefront Hooks	611
• cart.lasso	612
• cartrecover.lasso	614
• cartsave.lasso	615
• checkout.lasso	616
• closed.lasso	619
• detail.lasso	620
• emailfriend.lasso	621
• helpcenter.lasso	623
• index.lasso	624
• modifyitem.lasso	626
• modifypayment.lasso	627
• modifyshipping.lasso	629
• review.lasso	631
• search.lasso	634
• wishlist.lasso	636
• Help Center Hooks	638
• customerservice.lasso	639
• Cookies	640
Creating HostedStore Modules	641
Database Schema	643
• AdminUser	645
• Affiliate	646
• AutoShipOrderQueue	647
• Customer	649
• EmailArchive	650
• GiftCertificate	651
• Invoice	652
• Item	655
• PriceGroup	657
• Product	658
• Promotion	661
• Review	663
• ShoppingList	664
• Supplier	665

• Wishlist	666
HostedStore Change Log	667
• Version 2003.06.16	668
• Version 2002.12.01	673
• Version 2002.11.15	674
• Version 2002.07.15	680
• Version 2002.07.01	684
• Version 2002.06.17	685
• Version 2002.05.27	686
• Version 2002.05.20	688
• Version 2002.05.15	690
• Version 2002.04.15	694
Manifest	695
Just Checking...	708

Trademarks

HostedStore and the HostedStore logo are trademarks of HostedStore Corporation.

Adobe, the Adobe logo, and GoLive are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Macromedia and Dreamweaver are trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries. Lasso, Lasso Professional, Lasso Studio, LDML, Lasso Service, Lasso Connector, Lasso Web Data Engine, Blue World and Blue World Communications are trademarks of Blue World Communications, Inc. MySQL is a trademark of MySQL AB. Mac OS is a registered trademark of Apple Computer, Inc. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. VeriSign, the VeriSign logo, Payflow, and Payflow Link are registered or unregistered trademarks of VeriSign and its subsidiaries in the United States and in foreign countries. Order Manager is a trademark of Stone Edge Technologies, Inc. BluePay is a trademark of Online Data Corp. Plug'nPay is a trademark of Plug and Pay Technologies, Inc. Planet Payment is a trademark of Planet Group, Inc. All other trademarks are the property of their respective owners.

Mention of products and services from companies other than HostedStore Corporation is for informational purposes only and constitutes neither an endorsement nor a recommendation. All persons listed in the examples here are purely fictitious, and any resemblance to existing persons is purely coincidental.

End User License Agreement

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT "LICENSE" CAREFULLY BEFORE INSTALLING OR USING THIS SOFTWARE. BY DOWNLOADING, INSTALLING, AND/OR USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE SOFTWARE AND (IF APPLICABLE) RETURN THE SOFTWARE TO THE PLACE WHERE YOU OBTAINED IT FOR A REFUND.

1. DEFINITIONS

- (i) "HOSTEDSTORE" is the HostedStore.LassoApp file, store template files, image files, content page files, and email files which accompany this License.
- (ii) "LICENSEE" refers to the person, entity or organization which is using "HOSTEDSTORE", and any successor or assignee of same.
- (iii) "Domain" is a fully qualified domain name in the Internet domain name space.
- (iv) "Store" is one installation of the HOSTEDSTORE software used for a single Domain.
- (v) "Original Source Code" is a collection of source codes developed by LICENSEE without using the source codes provided with HOSTEDSTORE.
- (vi) "Hooks" is a collection of functions and files that interface with the HOSTEDSTORE system.

2. PURPOSE OF LICENSE

This license is for the downloading, acquisition and use of HOSTEDSTORE and any permitted modifications of HOSTEDSTORE store templates, images, and content pages. By downloading, acquiring and using HOSTEDSTORE, LICENSEE expressly agrees to abide by this License.

3. LICENSE

This license is between LICENSEE and HostedStore Corporation. LICENSEE acknowledges that HostedStore Corporation is and has granted this license in reliance upon representations made by LICENSEE and LICENSEE warrants the truthfulness and accuracy of such representations. Any breach of this warranty of truthfulness and accuracy shall constitute a breach of this License. HostedStore Corporation has the sole and unqualified right to refuse to license any person or entity for any reason, except for reasons which are specifically prohibited by state and national laws and statutes.

4. RIGHTS GRANTED

LICENSEE is granted a non-exclusive license to use HOSTEDSTORE in conjunction with one Domain and one Store. A License to use HOSTEDSTORE with more than one Domain or with more than one Store must be separately purchased from HostedStore Corporation and is specifically not granted in this License.

5. PROPRIETARY RIGHTS

HOSTEDSTORE contains valuable proprietary rights of HostedStore Corporation. There is no transfer to LICENSEE of any title to or ownership of HOSTEDSTORE or any copyright, trade name, trade secret, source code, or other proprietary rights relating to HOSTEDSTORE, regardless of the form that HOSTEDSTORE exists. LICENSEE may not violate these rights. LICENSEE must take all appropriate steps to protect HostedStore Corporation's rights. HostedStore Corporation may at any time replace, modify, alter, improve, enhance, or change HOSTEDSTORE without prior notice.

6. LIMITED LICENSE TO CUSTOMIZE SITE

LICENSEE is granted a license to modify and customize the HOSTEDSTORE store templates, images, and content pages to implement custom features, fix bugs, and otherwise make HOSTEDSTORE useful for any particular application for LICENSEE's requirements, EXCEPT THAT, LICENSEE may not convert HOSTEDSTORE to a different computer language, either manually, or using an automated conversion tool, such that HOSTEDSTORE or any modification thereof will run under any language, software, or program other than or independent of, Lasso Professional 5.

7. NO TRANSFER

Furthermore, LICENSEE may not, except as permitted by applicable legislation, decompile, reverse engineer, disassemble, or otherwise reduce the HostedStore.LassoApp file to human-perceivable form. LICENSEE agrees that any modifications made to the HOSTEDSTORE store templates, images, and content pages are for the exclusive use of LICENSEE and may not be transferred, sold or licensed to another entity.

8. USE OF HOOKS

LICENSEE may create and license to other entities Original Source Code that interfaces with HOSTEDSTORE using the HOSTEDSTORE Hooks as those Hooks are defined by HostedStore Corporation in the documentation available from HostedStore Corporation's Web site.

9. **REVOCAION RIGHTS**

Any license, right, or permission to use HOSTEDSTORE is automatically revoked if LICENSEE breaches any term or condition of this License. LICENSEE agrees that in the event of a breach of this License, and upon the demand thereof by HostedStore Corporation, LICENSEE will immediately return any physical copies of HOSTEDSTORE to HostedStore Corporation, will cease any further use of HOSTEDSTORE, will remove the original and any modified or altered HOSTEDSTORE store templates, images, and content pages from his/her/its computers and drives, will cease his/her/its use of HOSTEDSTORE, and will cease any claimed breach of this License. LICENSEE agrees that monetary damages alone are not an adequate and just relief resulting from any breach of this License, that a court order prohibiting any further breach of this License is necessary to prevent further damages, and that LICENSEE will not oppose any reasonable request for a temporary restraining order, preliminary injunction, or other relief sought by HostedStore Corporation in the event of a breach of this License. HostedStore Corporation shall not be required to notify LICENSEE of any breach, nor make any demand or claim against LICENSEE resulting from any such breach, or for a demand to stop any use or distribution in violation of the terms of this License, and LICENSEE agrees that any breach of this License and damages resulting therefrom shall relate back to the first and earliest breach thereof. Failure of HostedStore Corporation to enforce its rights pursuant to this License shall not constitute a waiver of such rights, and shall not prejudice HostedStore Corporation in any later enforcement of its rights or rights to seek damages therefrom.

10. **DISCLAIMER**

HOSTEDSTORE CORPORATION WILL NOT BE LIABLE FOR (A) ANY BUG, ERROR, OMMISSION, DEFECT, DEFICIENCY, OR NONCONFORMITY IN HOSTEDSTORE; (B) IMPLIED MERCHANTABILITY OF FITNESS FOR A PARTICULAR PURPOSE; (C) IMPLIED WARRANTY RELATING TO COURSE OF DEALING, OR USAGE OF TRADE OR ANY OTHER IMPLIED WARRANTY WHATSOEVER; (D) CLAIM OF INFRINGEMENT; (E) CLAIM IN TORT, WHETHER OR NOT ARISING IN WHOLE OR PART FROM HOSTEDSTORE CORPORATION'S FAULT, NEGLIGENCE, STRICT LIABILITY, OR PRODUCT LIABILITY, OR (F) CLAIM FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES, OR LOSS OF DATA, REVENUE, LICENSEES GOODWILL, OR USE. IN NO CASE SHALL HOSTEDSTORE CORPORATION'S LIABILITY EXCEED THE PRICE THAT LICENSEE PAID FOR HOSTEDSTORE.

11. **JURISDICTIONAL LOCATION**

This License is granted, HOSTEDSTORE was developed, technical support is being provided in, and LICENSEE is downloading this copy of HOSTEDSTORE via miscellaneous resellers, distributors, gateways and servers, and is paying for the use of HOSTEDSTORE, in and from the County of King, State of Washington, and the parties agree that this License is to be performed in the County of King, Washington.

12. **CHOICE OF LAW**

This License Agreement shall be governed by the laws of the State of Washington, USA. If any provision of this License is found void, invalid, or unenforceable, it will not affect the validity of the balance of this License, which shall remain valid and enforceable according to its terms. In the event any party is required to seek judicial enforcement regarding any terms of this License, or to seek a judicial interpretation regarding any terms of this License, the party deemed by the court to be the prevailing party shall be entitled to, in addition to its costs and damages, reasonable attorney's fees and costs as determined by the court.

13. **EXPORT CONTROLS**

LICENSEE acknowledges that HOSTEDSTORE may be subject to United States export controls, pursuant to the Export Administrations Regulations, 15 C.F.R. Parts 768-799. LICENSEE shall comply strictly with all requirements of the Export Administration Regulations regarding use of HOSTEDSTORE.

14. **SOLE AGREEMENT**

This License Agreement constitutes the entire and sole agreement between LICENSEE and HostedStore Corporation, and is agreed to and entered into on the date LICENSEE downloads or implements HOSTEDSTORE, whichever date is earlier.

About the Documentation

The intent of this documentation is to provide an overview of HostedStore functionality, to provide details about setting up and administering a store, to provide details about the default content pages and email format files, and to explain how to alter the functionality of HostedStore through the use of hooks and modifications to default content pages and email format files. The documentation is not meant to serve as a guide to modifying HostedStore nor as a primer on Lasso programming techniques. Resources for both of those topics can be found in the [Resources Available](#) section.

While the documentation should provide all of the information necessary to make any modification or enhancements to HostedStore, it may not be enough for users that are new to Lasso programming nor for users that need a little help or encouragement to make changes. Detailed articles explaining possible modifications may be available in the [support area](#) of the HostedStore Web site. Users are encouraged to add their own articles or add to existing articles. And of course feel free to post questions to the HostedStore support list. Someone is sure to provide the answer needed or to at least provide some insight into how to approach the problem.

Excuses and Sincere Apologies

The documentation is a work in progress. Well, honestly, the documentation is more than a work in progress--it is preliminary and just downright lame. Rest assured that the documentation will be improved over time. Any feedback on how to improve the documentation for new and current users is welcomed. The support area of the HostedStore Web site (www.hostedstore.com/support/) provides an area to submit, view, and comment on documentation modification requests.

In the meantime, please ask any questions on the support list. Questions will be answered promptly and will also provide a valuable indication of where the documentation can be fleshed out and improved.

Thanks for your help, patience, and, most of all, your understanding!

Documentation Style Conventions

An attempt has been made to give different items in the documentation a special appearance to set them apart from the regular text.

- *Italic* lettering is used for references to sections in the documentation, commands, email addresses, URLs, filenames, emphasized text, and first references to terms.
- **Bold** lettering is used for names of databases, tables, fields, buttons, and hyperlinks that appear on a page.
- `Constant width` lettering is used for literals, constant values, variable names, code listings, and HTML markup.
- *Constant width italic* lettering is used for replaceable parameter and variable names and values.
- **Constant width bold** lettering is used to highlight the portion of a code listing being discussed.

Definition of Terms Used

- **Content Page** - Pages inserted into a template to provide the text, graphics, etc for a particular page. The term "default content page" is used frequently to refer to the original state or functionality of a content page when HostedStore is installed, i.e., the content page as it was provided with the HostedStore distribution.
- **Custom Tag** - A Lasso tag written using LDML.
- **Format File** - A file that contains LDML.
- **LDML** - Lasso Dynamic Markup Language. An acronym for the Lasso language.

- **Preference Variable** - A variable set in Store Configuration. More details about Store Configuration can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.
- **Server Administrator** - The person(s) with privileges to install files on the server used to host the HostedStore application.
- **Shared Hosting Environment** - Refers to hosting environments in which the user is not the server administrator. For example, this may apply to Web hosting services purchased from a hosting provider or to users in an organization where an IT department controls the servers.
- **SSL** - Secure Socket Layer. An encryption technique that provides a secure connection for transmitting data safely over the Internet. SSL connections are usually made on port 443 of the Web server, whereas standard connections are made on port 80--in case that means anything to anyone. Secure web connections begin with https in the URL, whereas a standard connection URL begins with http. However, just because a URL begins with https does not guarantee that a secure connection with the Web server has been made. A successful secure connection is generally indicated with a lock icon or similar icon in the Web browser's interface.
- **Store Administration** - The area of the store where all administration related tasks like the setting of preference variables, order processing, and product modification are performed.
- **Store Administrator** - The person(s) given access to the "behind the scenes" aspect of HostedStore. The store administrator is allowed to perform tasks such as adding or modifying products, adding affiliates, and processing orders.
- **Store Configuration File** - The *HS_datafile.lasso* file located in the */hostedstore/includes/* directory. This file stores the encrypted preference variables set in *Store Configuration*. More details about *Store Configuration* can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.
- **Storefront Template** - The file that provides the look for a store.
- **www.yourserver.here** - The generic URL used in example links. The URL should be replaced with the appropriate fully qualified domain name. This is usually the fully qualified domain name that was used for registering the HostedStore license.

Interchangeable Words/Terms

Sometimes words and terms are used interchangeably within the documentation. An attempt has been made to gather a list of such words below to help eliminate any confusion that may be caused. It is likely that a few terms were missed so when in doubt ask on the support list. If the explanations below lead to further confusion then please ignore and pretend it was never read.

- *Product* and *item* are often used interchangeably to refer to a product, service, item, etc that is entered into the **Product** table and sold in the store. It is hard to determine for what purposes HostedStore will be used so, for example, the term product may not strictly apply.
- *Order* and *invoice* are used to refer to a transaction in the store involving the purchase of product(s). Order is used more frequently, but invoice will be used on occasion because the word "order" is a reserved word in SQL (hence the naming of the table used to store orders as **Invoice** instead).
- *Store Administrator* and *Store Manager* are both used to refer to the person(s) that have access to the store administration area.

PDF Version of the Documentation

The documentation site (documentation.hostedstore.com) is captured as a PDF as a convenience for those that do not have access to the Internet, prefer PDFs, and for a handful of other reasons. There are, however, some side effects and inconveniences for the end reader as a result of this process:

- Text may not be displayed as indicated in the *Documentation Style Conventions* section above. Bummer. It makes parts of the documentation much more difficult to follow.
- Actual size images were used to make it very easy to compare what is seen in the browser to what is documented. Due to page breaks there may be some pages that contain an image and little to no text. In some cases it was necessary to scale the image to fit into a single page, but the details should still be legible.
- There may be pages that are not completely filled with content due to the page cutoffs for single page images. In some

cases there may even be blank pages (an attempt was made to remove those pages, but one or two may have been overlooked).

Documentation Updates

The documentation is in a constant state of revision as user feedback is incorporated and new features are added to HostedStore. Each release of HostedStore will contain the most recent version of the documentation. The most current version of the documentation can be found online at documentation.hostedstore.com.

The online version of the documentation used to create the documentation that is included in each release. It will contain the most up to date information--even for features that have not been released yet.

Important: If the online version of the documentation does not display any previous and next hyperlinks for the navigation, then the Web site is being used to build the documentation PDF. There are still hidden navigation links that can be used by mousing over the center of the top of the page.

The online version of the documentation also provides an interface which allows users to add their own comments to specific pages in the documentation. Any submitted comments will be used to provide additional information or clarification of existing information in the next version of the documentation.

Resources Available

Several resources are available for HostedStore and for Lasso.

- The HostedStore support list provides general support for the HostedStore application. This resource can also be used to ask questions about customizing HostedStore. It may even be possible to slip in a Lasso question too.
- The support area of the HostedStore Web site (www.hostedstore.com/support/) provides articles and code provided by HostedStore and third parties.
- The Blue World Web site (www.blueworld.com) provides several lists of Lasso related resources. If not already subscribed, we encourage subscription to the Lasso Talk list if a greater understanding of Lasso is wanted.

Lasso Professional 6 Compatibility

Every effort has been made to ensure that this version of HostedStore is compatible with Lasso Professional 6 (LP6). However, this version and all prior versions have been completely written in Lasso Professional 5 (LP5) so there is no guarantee of compatibility with LP6. That in mind, there have been users that have successfully deployed prior versions of HostedStore on LP6 without any problems so this warning should not be a deterrant to deploying HostedStore on LP6. If HostedStore is deployed using LP6 and an issue does arise that appears to be related to differences between LP5 and LP6, please provide a bug report and an attempt will be made to address the issue.

Introduction

HostedStore is an ecommerce application that can be used for a large variety of businesses, including businesses with a few products to businesses selling tens of thousands of products online, via mail order, and even in a retail location.

A complete set of content and display templates provide an easy means for store managers to get their store online with little or no modification. While a significant amount of thought went into making HostedStore feature complete for the needs of most users, it is recognized that some store managers will want to tailor HostedStore's functionality to specific needs.

The core of HostedStore uses Lasso™ Professional 5, a Web application programming language from Blue World Communications, Inc. (www.blueworld.com). Lasso was chosen specifically for its ease of use.

Virtually anyone, even those without prior programming experience, can make enhancements to HostedStore using Lasso Dynamic Markup Language (LDML) if and when it is necessary. More details about modifying HostedStore can be found in the [Modifying HostedStore](#) section.

If customization is needed and the store manager would like the aid of a developer, a list of developers offering development services for the HostedStore product line is provided on the HostedStore Web site (www.hostedstore.com). HostedStore users are also free to make a request for future features. The support area of the HostedStore Web site (www.hostedstore.com/support/) provides an area to submit, view, and comment on feature modification requests. Requests that would be of general use and would strengthen the HostedStore application will be considered for future features.

Features Summary

By the time this documentation is read the features of HostedStore are probably well known. A brief overview of HostedStore features is provided below just in case something slipped by.

Storefront

- All of the storefront content is generated from format files that can be modified by the end user.
- Customizable HTML templates are provided for the store look. Select a template or easily create a custom template following the instructions provided in the documentation.
- The store home page displays featured products and favorite products (top selling products based upon actual store sales). The number of featured and favorite products to display can be easily modified.
- Products may have related products identified to display up-sell, cross-sell, or necessary accessory items to the customer.
- Multiple templates can be used for the display of product information.
- A static cart display on all pages provides the customer with a constant summary of the number of items in the cart and the subtotal of the cart.
- The shopping cart display helps eliminate abandoned carts by providing an order total including shipping and tax.
- A built-in help center can be modified to describe all aspects of the store order process, account administration, and store policies to help eliminate the need to address recurring customer service questions.
- An address book allows customers to save multiple shipping addresses to facilitate a quicker and friendlier checkout process.
- Recurring orders can be saved as a shopping list to save time reselecting products for each order.
- Email notifications for all steps of the order process (credit card approved, order processed, order shipped, and order cancelled) increase customer confidence and simplify store administration.
- The online order tracking provides 24/7 customer service. Customers can view past/current orders, obtain shipping status, obtain tracking numbers, and cancel orders.
- Support for product meta tags and static page generation to aid in search engine positioning.

Order Processing

- Support for credit cards, checks (online and via phone/mail), purchase orders, and PayPal provide customers with multiple payment options. Accepted payment types can be configured to match the policies of the store.
- Support for real-time credit card and echeck processing via multiple payment gateways. The processing process can be easily modified to support additional payment gateways.
- The entire order process supports the use of SSL encryption so customer information is secure. Account numbers are stored encrypted and can be purged from the database when no longer needed or masked to prevent the viewing of the entire account number.
- Modifiable shipping calculation script allows for generalized calculations based upon total price, total weight, fixed rate, etc or product specific calculations based upon price, weight, fixed rate, etc. Ability to include any additional pricing to recover costs like labor for processing, packing materials, etc.
- Modifiable tax calculation script supports different tax rates for multiple cities, states/provinces/regions, countries, etc. Products can be assigned to specific tax classes that can be utilized for more complex tax scenarios.

Product Management

- Definable categories and multiple subcategories allow for complex catalog hierarchies.
- Support for multiple price levels: MSRP, standard price, sale price, and member price.
- Each product can have its own quantity ranges that equate to volume price discounts.
- Products may have unlimited attributes. Attribute names can be defined-there are not simply just size and color-get creative and use the attributes to add service plans, optional gift wrapping, monogramming, and more. Attributes can have value added pricing to add or subtract from the base price based upon the customer's selections. Attributes can also incorporate other products and have the other product's inventory properly adjusted.
- Products can be flagged for Electronic Asset Delivery. Customers can then download purchased images, software, and other digital assets.
- Product image upload feature that will upload several images simultaneously through the Web browser. A much more convenient process than FTP when just adding a few products.
- An image report matches products in the table to images on the server and provides warnings about missing images or products that have not been entered into the database.
- Products are automatically removed from the catalog of available products if a definable inventory threshold is reached. Option to override product removal action so that out of stock items are displayed to the customer, but with purchasing capability disabled. Alternatively, out of stock items can be purchased but the customer will be notified that the product is back ordered via a global or product specific out of stock message.
- Definable purchase limits per item or per customer.
- Inventory enforcement of limited stock items.
- Definable product dependencies/restrictions to dictate what other products must be purchased with an item.
- Ability to restrict the display of products to customers that do not belong to a specific price group.
- Ability to sell membership products that offer special pricing on other products to member or restricts products to members only.
- Current inventory levels for all products can be modified simultaneously with one form.
- Suggested inventory order report based upon defined build-to amounts and current inventory.

Order Management

- Line item status modification allows ordered items to be individually canceled, back ordered, or shipped.
- Notification by email when a customer has placed or cancelled an order eliminates the need to constantly log into store administration to check for new orders.
- Post order adjustments can be made to the order subtotal, shipping cost, tax cost, and promotional discount. Batch adjustments can also be made to the order when batching against a third-party report such as a bank statement or payment gateway report.

Affiliate Management

- Assigns orders to an affiliate based upon their unique definable affiliate ID.
- Different percentages or fixed monetary values can be assigned to individual affiliates.
- Management reports determine which products are being sold by affiliates and affiliate sales summaries.
- Can be used to generate unique URLs to track the success of individual marketing campaigns.
- Online affiliate management give affiliates direct access to their sales reports.
- Track affiliate payouts.

Coupon/Promotion Management

- Versatile system can be used for individual coupons or promotions, discounts for customers that have accumulated membership points, gift certificates, and tracking of marketing campaigns.
- Date/time stamp promotions to offer limited time coupon offers to customers.
- Automate marketing promotions by auto-sending coupons to new customers, customers that purchase

specific items, all customers during a time period, etc.

- Ability to limit coupons to specific products, specific customers, minimum number of order items, minimum order value, one-time use, and more.

Mailing List Management

- Customers may subscribe to a store newsletter during creation of their account or later in account management. Subscribe and unsubscribe commands are emailed directly to the list management software.
- Compose and send customer notifications within HostedStore to a specific customer, to all customers, or to newsletter subscribers.

Comprehensive Reporting

- Sales conversion reports track the number of times a product is viewed, placed in a cart, and purchased. The data can be helpful in determining the successful conversion of curious shoppers to buyers.
- Reports of top selling products, top customers, total sales, costs, profit margins, and tax liability help to maximize store profitability.
- Custom reports can be built if an external reporting application is not an option.

Modular Component Structure

- Each of the major features in HostedStore uses a modular structure that allows individual features to be toggled on or off to match current and future business needs.
- Access to the content pages allows for development of hooks into newly developed features-either your own, from a third party, or from HostedStore. Some modules are already available for download from the HostedStore Web site. See the [module directory](#) for a list of modules currently available from third parties and/or HostedStore.

Extensibility

- Custom code can be included in the content pages to provide additional functionality to the core application.
- A system of hooks is provided that allows developers to insert their custom code into the functionality of the core application. Using hooks, sections of HostedStore code can even be disabled so developers can insert their own code instead.
- Third party developers can create modules that can be inserted into the store core code and the store administration configuration pages to add functionality or alter existing functionality.

Installation and Configuration

Installing and configuring HostedStore is a rather simple process. It does require some time and some thought as to what features will be wanted. Installation may take longer if a working installation of Lasso Professional 5 is not already available. The good news is that HostedStore, once installed and configured, will run happily without need for further configuration.

Installation and configuration involves five steps:

1. Installing the files on the Web server and setting permissions (if applicable).
2. Installing or creating the database(s)/tables.
3. Configuring Lasso Administration to recognize the database(s)/tables and to provide permissions needed by HostedStore to function properly.
4. Loading the HostedStore.LassoApp file into memory.
5. Configuring preferences.
6. Making initial modifications to the content pages, email files, and select lists.

Many steps in the installation and configuration process require the permissions usually given only to server administrators. For example, setting ownership and permissions for files and directories and creating users in Lasso Administration are often performed by the server administrator. If HostedStore will be used in a shared hosting environment or in a large organization with an IT department, be sure to seek the aid of the server administrator for some of the steps that follow.

Sections

- [Required Software](#) - Minimum software necessary to use HostedStore.
- [Installing the HostedStore Files](#) - Instructions for installing the downloaded files.
- [Creating the Database\(s\) and Tables](#) - Instructions for configuring the database application to be used with HostedStore.
- [Configuring Lasso Administration](#) - Necessary security and preference settings for Lasso.
- [Loading HostedStore into Memory](#) - How to access HostedStore.
- [Store Configuration](#) - Setting store preferences.
- [Initial Modifications](#) - Suggestions for minimum modifications to meet the needs of the store.

Required Software

A working installation of Lasso Professional 5 is needed to use the HostedStore application. Consult the Blue World Web site (www.blueworld.com) for more information about obtaining and installing Lasso Professional 5. Lasso Professional 5 is also available for purchase at a discount from the HostedStore Web site (www.hostedstore.com.com/pricing.html).

A database application supported by Lasso will also be needed. HostedStore ships with a build script for a MySQL database that can be used with Lasso MySQL or a separate installation of MySQL. If a different database application will be used, please read the [Database Schema](#) section for details about the database schema expected by HostedStore. Build scripts for other Lasso compatible database applications may be available in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Due to limitations of Lasso Professional 5, a third party tag must be used to connect to some payment gateways. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

If the Verisign Payflow Pro payment gateway will be used for credit card or echeck payments, then HostedStore must be deployed on Lasso Professional 6. More information can be found in the [Payment Processing > Built-In Forms > Credit Card Processing > Verisign® Payflow Pro \(SM\)](#) and [Payment Processing > Built-In Forms > Check Processing > Verisign® Payflow Pro \(SM\)](#) sections.

Installing the HostedStore Files

HostedStore is installed by simply moving supplied files to their appropriate destination directories.

1. Move the supplied */hostedstore/* directory and the *HostedStore.LassoApp* file into the root of the Web serving directory for the domain that will host the store. Note that HostedStore will not work properly unless the */hostedstore/* directory is located at the root of the Web serving directory.

Important: The name and location of the directory containing the HostedStore files cannot be modified.

The purpose and contents of each of the directories in the */hostedstore/* directory will be explained below in the appropriate section. A list of all files distributed with HostedStore can be found in the [Manifest](#) section. A quick overview of the directory structure and purpose of the */hostedstore/* directory are provided below:

- */hostedstore/admincustomer/* - Contains two directories, *email/* and *html/* that contain the modifiable email bodies (the text of the email) and content pages, respectively, for the *Customer Account Administration* area of the store. The *Customer Account Administration* area allows the customer to view past orders, create shopping lists, maintain wish lists, etc. More details about the *Customer Account Administration* area can be found in the [The Customer's Perspective > Account Administration](#) section.
- */hostedstore/adminstore/* - Contains two directories, *email/* and *thirdparty/* that contain the modifiable email bodies (the text of the email) and a repository for third party tags, modules, etc, respectively, for the *Store Administration* area of the store. The *Store Administration* area is where products are maintained, orders are processed, reports are generated, etc. More details about the *Store Administration* area can be found in the [Store Administration](#) section.
- */hostedstore/admsupplier/* - Contains two directories, *email/* and *templates/* that contain the modifiable email bodies (the text of the email) and template files, respectively, for the *Supplier Account Administration* area of the store. The *Supplier Account Administration* area is where suppliers that drop ship products for the store can view and process orders that are awaiting drop shipment. More details about the *Supplier Account Administration* area can be found in the [The Supplier's Perspective](#) section.
- */hostedstore/affiliate/* - Contains two directories, *email/* and *html/* that contain the modifiable email bodies (the text of the email) and content pages, respectively, for the *Affiliate Account Administration* area of the store. The *Affiliate Account Administration* area allows affiliates to view orders that have been placed in the store due to the affiliate's efforts and view earned and received payouts. More details about the *Affiliate Account Administration* area can be found in the [The Affiliate's Perspective](#) section.
- */hostedstore/helpcenter/* - Contains two directories, *email/* and *html/* that contain the modifiable email bodies (the text of the email) and content pages, respectively, for the *Help Center* area of the store. The *Help Center* area is a central repository for FAQs, store policies, etc. More details about the *Help Center* area can be found in the [The Customer's Perspective > Help Center](#) section.
- */hostedstore/images/* - Contains the product images and attribute description files.

Once installed, product images can be placed in the appropriate directory:

- Product thumbnail images belong in the */hostedstore/images/products/thumbnails/* directory. Thumbnail images are displayed on the default search results content page (*/hostedstore/storefront/html/search_list.lasso*), the default email a friend content page (*/hostedstore/storefront/html/email_friend.lasso*), and the default product review content page (*/hostedstore/storefront/html/review_form.lasso*).
- Product detail images belong in the */hostedstore/images/products/detail/* directory. Detail images are displayed on the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*).
- Product zoom images belong in the */hostedstore/images/products/zoom/* directory. Zoom images are accessed via a popup window hyperlink on the default product detail content page

(*/hostedstore/storefront/html/product_detail.lasso*). Use of zoom images is an optional feature that can easily be toggled on or off with a preference variable the Store Configuration. More details about Store Configuration can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

- Product description files belong in the */hostedstore/images/products/descriptions/* directory. More details about optional product description files can be found in the [Store Administration](#) > [Product Administration](#) section.

If the file upload feature in *Product Administration* will be used, then ownership of the */hostedstore/products/images/* directory and all subdirectories should be assigned to the `lasso` user or `staff` group and `write` and `read` permissions should be granted. More details about the file upload feature can be found in the [Store Administration](#) > [Product Administration](#) section.

- */hostedstore/includes/* - Contains multiple modifiable files that are used across the entire store.
 - */hostedstore/products/* - This directory holds static product detail content pages if the option to generate the pages is used in the *Store Administration* area. More details about static product detail content pages can be found in the [Store Administration](#) > [Product Administration](#) > [Generate Static Product Pages](#) section.
 - */hostedstore/storefront/* - Contains three directories, *email/*, *gateways/*, and *html/* that contain the modifiable email bodies (the text of the email), forms for submitting information to payment gateways, and content pages, respectively, for the main storefront. The storefront is where customers shop for products, read product reviews, add items to their cart, submit payment processing information, etc.
 - */hostedstore/templates/* - Contains directories of template files that are used to define the overall "look and feel" of the store.
 - */hostedstore/thirdparty/* - A repository for third party tags, modules, etc used in the store.
2. The contents of two files, *HS_datafile.lasso* and *HS_serialnumber.lasso*, were issued with the demo serial number or purchased license serial number. Create both files with the appropriate contents, save the files, and move both files into the */hostedstore/includes/* directory. Make sure that the contents of the two files contains only the information that was issued. Nothing more, nothing less. There are two common mistakes when copying the information to the appropriate files:
- A line break or carriage return is added to the end of the contents. There should not be a line break or carriage return in the files.
 - The contents of the data file, *HS_datafile.lasso*, contains line breaks. This error is often introduced by copying the data file contents from an email that has line breaks inserted by the email client. Remove all line breaks in the data, as well as any trailing carriage return.

Important: HostedStore will not function properly unless the contents of both files are exactly as issued.

Ownership of the *HS_datafile.lasso* file should be assigned to the `lasso` user or `staff` group and `write` and `read` permissions should be granted.

Important: HostedStore will not function properly unless both of files are present and the appropriate permissions have been granted.

3. Realms should be placed on the following directories to prevent the reading of sensitive information through a Web browser. For details on creating realms consult the Web server documentation or, if applicable, seek the aid of the server administrator.
- */hostedstore/adminstore/report/* - This directory can be used to store reports exported from the *Reports* area of *Store Administration*. More details about reports can be found in the [Store Administration](#) > [Reports](#) section.
 - */hostedstore/backups/* - This directory can be used to store database backups and database exports performed in the *Maintenance* area of *Store Administration*. More details about HostedStore maintenance can be found in the [Store Administration](#) > [Maintenance](#) section.
 - */hostedstore/EADFiles/* - This directory is used to store files that are downloaded via electronic asset delivery (EAD). More details about EAD can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) > and [The Customer's Perspective](#) > [Account Administration](#) sections.
 - */hostedstore/includes/* - This directory stores the *HS_datafile.lasso* and *HS_serialnumber.lasso* and various other files. However, the intent of this realm is strictly to prevent access to the *HS_datafile.lasso* and *HS_serialnumber.lasso* files.

Important: If a shared SSL certificate will be used for the store and a realm is placed on the */hostedstore/includes/* directory, then the username and password for the realm must match the username and password used to access *Store Administration* (the `HS_AdminLogin` and `HS_AdminPassword` preference variables, respectively) or HostedStore will not function properly. More details about setting the username and password used to access *Store Administration* can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

- */hostedstore/ordermanager/* - This directory is used to store database exports for Stone Edge Order Manager. More details about exporting for Stone Edge Store Manager can be found in the [Store Administration](#) > [Order Administration](#) > [Export for Stone Edge Order Manager](#) section.

Creating the Database(s) and Tables

- **Instructions for MySQL**

Either Lasso MySQL or an installation of MySQL can be used with HostedStore.

- Create a database (or databases) using the mysql client. Substitute in the name of the database for *DatabaseName* below. If in a shared hosting environment, be sure to follow the naming conventions suggested by the server administrator.

```
CREATE DATABASE DatabaseName ;
```

HostedStore supports the ability to have each of the tables or combinations of tables in separate databases. This feature was provided for users that have existing databases with tables that can be used (like a customer table, for instance) that will be kept separate from other tables necessary for HostedStore to function. If existing databases will not be used, then it is recommended that all HostedStore tables reside in one database. This makes a lot of administrative functions easier plus it may conform better to account specifications for users in a shared hosting environment.

- Quit the mysql client and create the tables in the database using the *buildmysql.txt* file. Substitute in the name of the database for *DatabaseName* below. Note that the following statement assumes that one database is used for all of the tables, the *buildmysql.txt* file is in the same directory where the command is issued, and that a username and password are not needed to access the mysql client. Make adjustments as necessary. If additional assistance is needed, seek the aid of the server administrator or ask the appropriate question on the HostedStore support list.

```
mysql DatabaseName < buildmysql.txt
```

- **Instructions for Other Lasso Compatible Database Applications**

At this time, HostedStore does not support the use of other Lasso compatible database applications. Support for additional database applications is planned for future versions. If support is needed immediately, then please submit a modification request in the support area of the HostedStore Web site (www.hostedstore.com/support/). Aid may be sought to help troubleshoot any issues encountered while adding support for additional database applications.

A user and password must be setup for the database in order to properly secure the data. The user will also need file permissions for the root Web serving directory to facilitate the use of image related functions and database backup/export functions. The [Installation and Configuration > Configuring Lasso Administration](#) section provides a list of minimum privileges that must be granted in Lasso Administration for HostedStore to function properly. Please consult the Lasso Professional 5 Setup Guide if a detailed explanation of Lasso Administration is needed.

Configuring Lasso Administration

Two different steps are involved in configuring Lasso Administration. The first, enabling database(s) and tables, registers and enables access to the database(s) and tables to be used with HostedStore. The second, setting user privileges, assigns database, file, and tag permissions to the group(s) and user that will be used with HostedStore.

Enabling Database(s) and Tables

In a shared hosting environment some or all of these steps may need to be performed by the server administrator.

Note that some of the steps outlined below are the roundabout way of configuring Lasso Administration. This is done to make the process universal to accommodate different states in which Lasso Administration may be. If already familiar and comfortable with Lasso Administration, then feel free to use some of the shortcuts provided in the interface.

1. Click the **Setup > Data Sources > Connectors** tab in Lasso Administration.
2. If the data source connector used for the database(s) has not been enabled, then click the hyperlink for the appropriate data source connector (*Lasso Connector for Lasso MySQL* and *Lasso Connector for MySQL* are currently supported), set the status to *Enabled* and click the **Update** button. Repeat as necessary. Click the **Setup > Data Sources > Connectors** tab to return to the state in step one above.
3. If the host for the database(s) has not been added yet, then click the hyperlink for the appropriate data source connector: *Lasso Connector for Lasso MySQL* or *Lasso Connector for MySQL*. Click the **Add Host...** button. Type in the appropriate IP address, port, default username, and default password and click the **Add Host** button. Repeat as necessary. Click the **Setup > Data Sources > Connectors** tab to return to the state in step one above.
4. Click the *List Hosts* hyperlink for the appropriate data source connector.
5. Click the *List Databases* hyperlink for the appropriate data source.
6. Click the database name hyperlink for the appropriate database. Set the status to *Enabled* and click the **Update** button.
7. Click the *List Tables* hyperlink in the left panel for the appropriate database. Set the status to *Enabled* and click the **Update** button.
8. Click the table name hyperlink for a table that will be used with HostedStore. Set the status to *Enabled* and click the **Update** button. Repeat this step for each of the database tables that will be used with HostedStore. An alternative method is to click the **Enable All** button if all of the tables listed for the database should be enabled.
9. Repeat steps four through eight as appropriate until all databases and tables that will be used with HostedStore have an enabled status.

Setting User Privileges

A minimum amount of privileges must be granted for HostedStore to function properly. Due to the flexibility of Lasso Administration there are multiple ways that privileges can be granted. The minimum privileges necessary will be listed below followed by an explanation of one way to grant the needed privileges. The instructions assume a familiarity with Lasso Administration. Please consult the Lasso Professional 5 Setup Guide if a detailed explanation of Lasso Administration is needed.

At a minimum, the following privileges must be granted:

- **Search Records, Add Records, Update Records, Delete Records, and Execute SQL** permissions for the group that has been assigned the database(s) used for the store. Database permissions for a group are granted on the **Setup > Groups > Databases** tab in Lasso Administration.
- **Search Records, Add Records, Update Records, Delete Records, and Execute SQL** permissions for all tables for the group that has been assigned the database(s) used for the store. Table permissions for a group are granted on the

Setup > Groups > Tables tab in Lasso Administration.

- **Create Files, Read Files, Write Files, and Inspect Files** permissions for files for the group that has been assigned the database used for the store. The **Allow Path** should be set to at least the store directory (*/hostedstore/*). File tags permissions for a group are granted on the **Setup > Groups > Files** tab in Lasso Administration.
- Multiple tags must be enabled for the group that has been assigned the database(s) used for the store. Tags permissions for a group are granted on the **Setup > Groups > Tags** tab in Lasso Administration.
 - The **[File_Control]**, **[File_Exists]**, **[File_Read]**, **[File_Create]**, **[File_Write]**, **[File_ListDirectory]**, **[File_CurrentError]**, and **[Log]** tags in the **File** category must be set to a status of *Allow*.
- Multiple tags must be enabled for the **AnyUser** group. Tags permissions for a group are granted on the **Setup > Groups > Tags** tab in Lasso Administration.
 - The entire **Custom Tag** category must be set to a status of *Allow*.
 - The entire **Include** category must be set to a status of *Allow*.
 - The **[LassoApp_Link]** tag in the **Technical** category must be set to a status of *Allow*.
 - The entire **Utility** category must be set to a status of *Allow*.

If the built-in file upload feature is to be used (see the [Store Administration > Product Administration > File Upload](#) section for an explanation of this feature), then additional file and tag permissions must be granted. Uploading of files requires access to files outside the root Web serving directory. This may pose a security concern especially in a shared hosting environment. To address that concern, it is recommended that a special group is created specifically for uploading of files that has only the minimum permissions outlined below. Then assign the user created above to the group. Alternatively, the same group used for the permissions set above could be granted permission to access files outside of the root Web serving directory.

Important: The creation of a special group will not prevent users assigned to the group from using the **[File_Copy]**, **[File_Delete]**, and **[File_Uploads]** tags for malicious purposes. This scenario merely limits the use of the tags to users that need the privileges to use a HostedStore feature. Limiting access to the specific directory used to temporarily store uploaded files as outlined below will prevent access to the files in the Web serving directories of other users, but will not restrict access to sensitive files (if any) that may be held in the directory used to temporarily store uploaded files. Granting use of the **[File_Copy]**, **[File_Delete]**, and **[File_Uploads]** tags should be carefully considered especially in shared hosting environments.

- **Copy Files** permission must be granted for the group allowed to handle uploaded files. The **Allow Path** should be set to the directory used to temporarily hold the uploaded files (*c : /TEMP/* for the Microsoft® Windows® 2000 operating system, */tmp/* for Mac® OS X operating system software). The **Access Files Outside of Root** checkbox must be checked (Lasso Administration displays this permission in red lettering for a reason!). File permissions for a group are granted on the **Setup > Groups > Files** tab in Lasso Administration.
- The **[File_Copy]**, **[File_Delete]**, and **[File_Uploads]** tags in the **File** category must be set to a status of *Allow* for the group allowed to handle uploaded files. Tags permissions for a group are granted on the **Setup > Groups > Tags** tab in Lasso Administration.

One possible way to configure Lasso Administration with the needed security privileges is provided below.

Important: This example scenario is not necessarily the best method for configuring Lasso Administration nor the way that may apply to every store installation. In fact, it is a rather generous scenario in that it grants use of all tags (well beyond the minimum requirements for HostedStore). This example is only presented as one possible way of granting privileges. **Carefully think through any permissions granted to groups in Lasso Administration.**

- Click the **Setup > Groups > Add Group** tab and add a group with a status of *Enabled*.
- Click the **Setup > Groups > List** tab and select the new group from the listing. Click the **Databases...** button. Select the appropriate data source connector, host, and database for the database used with the store. Assign **Search Records, Add Records, Update Records, Delete Records, and Execute SQL** permissions. If multiple databases are used with HostedStore, then this process will need to be repeated for each database.
- Click the **Setup > Groups > List** tab and select the new group from the listing. Click the **Tables...** button. Assign **Search Records, Add Records, Update Records, Delete Records, and Execute SQL** permissions for each table. If multiple databases are used with HostedStore, then this process will need to be repeated for each database.
- Click the **Setup > Groups > List** tab and select the new group from the listing. Click the **Files...** button. Assign **Create Files, Write Files, Copy Files, Delete Files, and Inspect Files** permissions. Check the **Access Files Outside of Root**

checkbox.

- Click the **Setup > Groups > List** tab and select the new group from the listing. Click the **Tags...** button. Assign all tags.
- Click the **Setup > Users > Add User** tab and select the new group from the listing of groups. Choose a username and password for the user. The chosen username and password will be the values used for the `HS_Username` and `HS_Password` preference variables, respectively, in Store Configuration. Details about Store Configuration can be found in the [Installation and Configuration > Store Configuration](#) section.

Loading HostedStore into Memory

Load the *HostedStore.LassoApp* file into memory by navigating to the *HostedStore.LassoApp* file with a Web browser. For example, if the *HostedStore.LassoApp* file is placed in the root Web serving directory, then navigate a browser to <http://www.yourserver.here/HostedStore.LassoApp>, where *www.yourserver.here* is replaced with the fully qualified domain name associated with the HostedStore license.

Important: If a prior version of HostedStore has been run on the server since the last restart of Lasso Service, then it will be necessary to dump (remove from memory) the prior version of the *HostedStore.LassoApp* file. A LassoApp can be dumped using one of the following methods:

- **Restart LassoService** - This is the least recommended option because it interrupts all Lasso-based Web traffic and can also remove other LassoApps from memory. Note that while this is the least recommended option, it is the only currently available option due to a bug in the currently shipping version (5.0.0) of Lasso Professional 5.
- **Via Lasso Administration** - Log into Lasso Administration and navigate to the **Setup > LassoApps** tab. In the right panel, type the name of the LassoApp (*HostedStore.LassoApp* in this case) and click the **Dump LassoApp** button. Access to Lasso Administration requires that one has administration rights. In many cases, especially in shared hosting environments, it may be necessary to seek the aid of the Web server administrator to dump a LassoApp.
- **Programatically** - Via the `LassoApp_Dump` technical substitution tag. This approach will only be successful if use of the `LassoApp_Dump` has been granted to the **All Users** user or a privileged user in Lasso Administration. Use of the `LassoApp_Dump` tag is granted in the **Technical** category on the **Setup > Groups > Tags** tab.

If a privileged user has been granted use of the `LassoApp_Dump` tag, place the following code in a format file, make the appropriate substitutions for *Username* and *Password*, and navigate to the file with a browser:

```
[Inline: -Username='Username' ,  
        -Password='Password' ]  
[LassoApp_Dump: 'HostedStore.LassoApp' ]  
[/Inline]
```

To prevent having to manually load the *HostedStore.LassoApp* file into memory each time the server is restarted, place a copy of the *HostedStore.LassoApp* file in the *LassoStartup* directory. Consult the Lasso Professional 5 documentation for more details. If access to the *LassoStartup* directory is not available due to a shared hosting environment, seek the aid of the server administrator.

Additional LassoApps

Additional LassoApps are provided with HostedStore to provide additional functionality:

- **HostedStore_Reports.LassoApp** - All of the reports and help links in *Store Administration* are processed by this LassoApp. The *HostedStore_Reports.LassoApp* file or an alias to the *HostedStore_Reports.LassoApp* file should be placed in the `/hostedstore/adminstore/` directory if the links in *Store Administration* are to function properly.
- **HostedStore_Tags.LassoApp** - HostedStore uses custom tags throughout the application. The custom tags are defined with each content page to ensure that the tag is available for use. In theory, the process of loading each content page can occur quicker if the custom tags do not need to be defined with each page. With that in mind, a separate LassoApp, *HostedStore_Tags.LassoApp* is provided so that the tags can be loaded globally when the server is started. Doing so will eliminate the need to define the custom tags with each content page. Details about the custom tags provided with HostedStore can be found in the [Modifying HostedStore > HostedStore Custom Tags](#) section.

To load the *HostedStore_Tags.LassoApp* file into memory each time the server is restarted, place a copy of the *HostedStore_Tags.LassoApp* file in the *LassoStartup* directory. Consult the Lasso Professional 5 documentation for more

details. If access to the *LassoStartup* directory is not available due to a shared hosting environment, seek the aid of the server administrator. Each HostedStore custom tag is prefixed with **HS_** to help prevent conflicts with other globally defined custom tags. It may be necessary to relay that information to the server administrator if they are concerned about placing custom tags in the *LassoStartup* directory.

Store Configuration

Much of the content control in HostedStore is achieved through preference variables that are stored encrypted in a store configuration file, *HS_datafile.lasso* located in the */hostedstore/includes/* directory. This file contains the username and password for the Lasso user and *Store Administration*, the URL paths to the store, a list of enabled module options, settings for modules, and more. An interface to the store configuration file is provided in *Store Administration*.

Because the configuration file contains sensitive information like usernames and passwords a decision was made to store the file in encrypted form so that if someone were to inadvertently or purposefully gain access to the file they cannot ascertain the information necessary to access the database(s) and decrypt sensitive information in the database(s). The overhead created by this approach is minimal compared to the benefits received. The encryption used is internal to HostedStore and will not be divulged to anyone. Third parties that need access to some of the information like the Lasso username and password will need to create their own preferences file for that purpose. See the [HostedStore Hooks](#) section for more information.

Important: Anything that involves writing data to disk is by no means failproof. It is recommended that a backup of the store configuration file is made after each modification. This will ensure that a quick recovery is possible in case corruption occurs. The encrypted store configuration file is located at */hostedstore/includes/HS_datafile.lasso*. It is unlikely that the store configuration file will be altered frequently (it is basically a "set it and forget it" process) so this extra safety measure should not be much of a burden.

Store Configuration is located in *Store Administration* which is accessed by navigating a browser to <http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=adminstore/index.lasso>, where *www.yourserver.here* is replaced with the fully qualified domain name associated with the HostedStore license, and clicking on the **Configure** image in the upper right-hand corner. Open access will be granted to configure the store if this is the first time that the store is being configured. If the store has been configured before, then a prompt to login with the *Store Administration* username and password (the *HS_AdminLogin* and *HS_AdminPassword* preference variables, respectively) or as an admin user with proper privileges will appear if the store administrator is not already logged in.



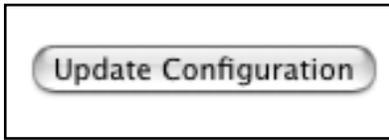
Configure button in the store administration navigation

Each item on the configuration form sets a corresponding preference variable that will be available to all of the store templates, content pages, and email files. Some preference variables are critical to the proper operation of the default store template, content pages, and email files. Others are used internally. Feel free to use any of the preference variables when customizing content, but it is recommended that none of the preference variable values are altered unless one really understands the ramifications of doing so. Each preference variable is named in the form *HS_VariableName*.

Note that all preference variables created by this file have names prefixed with *HS_* to help distinguish HostedStore preference variables from any variable that may be created when modifying HostedStore or using HostedStore in conjunction with other site content. When making modifications to HostedStore it is highly recommended to avoid variable names beginning with *HS_*, *HSH_*, *HSI_*, and *HSP_* prefixes. All four prefixes are reserved for use by HostedStore:

- *HS_* preference variables are set based on store preferences in the */hostedstore/includes/HS_datafile.lasso* store configuration file.
- *HSH_* variables are for use when making modifications to HostedStore with hook points. More details about hook points can be found in the [HostedStore Hooks](#) section.
- *HSI_* variables are for internal use and are not available for use except by code using hook points.
- *HSP_* variables are for public use in content pages and email files.

After all modifications have been made, click the **Update Configuration** button to apply the changes. If all required fields have been completed, then a success message will be returned along with all of the settings in the store configuration form. Otherwise, an error message will be returned along with an explanation of the problem(s) encountered.

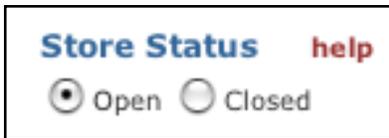


Update configuration button on the store configuration form

Each setting on the configuration form is discussed below by section along with mention of the corresponding variable:

Store Status

- **Store Status** corresponds to the `HS_StoreStatus` preference variable. If *Closed* is selected, then the customer will be shown a message that the store is closed, i.e., the customer will not be able to search for products, add products to their cart, etc. Use this feature when the store is to be closed for maintenance. The specific content page displayed to the customer is located at `/hostedstore/storefront/html/store_closed.lasso`. Details about customizing the page can be found in the [Modifying HostedStore](#) > [Content Pages](#) > [Storefront](#) > [Store Closed](#) section.



Store status area of the store configuration form

Security

- **Lasso Username** corresponds to the `HS_Username` preference variable. This is the username for the user that has been assigned database, tag, and file permissions in **Lasso Administration**. In a shared hosting environment, this value may be supplied by the hosting provider or server administrator.
- **Lasso Password** corresponds to the `HS_Password` preference variable. This is the password for the user that has been assigned database, tag, and file permissions in **Lasso Administration**. In a shared hosting environment, this value may be supplied by the hosting provider or server administrator.
- **Encryption Key** corresponds to the `HS_Encrypt` preference variable. This is the key used to encrypt data in forms and in the database(s). Keys may contain letters, numbers, and punctuation characters. A very long string that contains letters, numbers, and punctuation characters is recommended. The strength of the encryption directly corresponds to the length and complexity of the key.

Important: Sensitive information like credit card account numbers is encrypted with the key and stored in the database tables. Careful thought should be given to create a long and complex key to ensure the data is properly protected.

- **Store Admin Username** corresponds to the `HS_AdminLogin` preference variable. This is the username used to access *Store Administration*.
- **Store Admin Password** corresponds to the `HS_AdminPassword` preference variable. This is the password used to access *Store Administration*.

Security area of the store configuration form

Database Names

Multiple databases can be used to store the tables that HostedStore requires to function. The preference variables below are used to notify HostedStore of where to locate the needed tables. If the same database is used (this will likely be the scenario in most stores), then the same database name should be used for each of the preference variables below.

- **Key Field Name** corresponds to the `HS_KeyFieldName` preference variable. This is the name of the key field for all tables used by HostedStore. This preference variable will be used in future versions of HostedStore to support database applications that do not allow **ID** to be the key field name.
- **Customer** corresponds to the `HS_CustomerDB` preference variable. This is the name of the database that contains the table named **Customer**.
- **Invoice** corresponds to the `HS_InvoiceDB` preference variable. This is the name of the database that contains the table named **Invoice**.
- **Item** corresponds to the `HS_ItemDB` preference variable. This is the name of the database that contains the table named **Item**.
- **Product** corresponds to the `HS_ProductDB` preference variable. This is the name of the database that contains the table named **Product**.
- **Admin User** corresponds to the `HS_AdminUserDB` preference variable. This is the name of the database that contains the table named **AdminUser**.
- **Affiliate** corresponds to the `HS_AffiliateDB` preference variable. This is the name of the database that contains the table named **Affiliate**.
- **Autoship Order Queue** corresponds to the `HS_AutoshipOrderQueueDB` preference variable. This is the name of the database that contains the table named **AutoshipOrderQueue**.
- **Gift Certificate** corresponds to the `HS_GiftCertificateDB` preference variable. This is the name of the database that contains the table named **GiftCertificate**.
- **Price Groups** corresponds to the `HS_PriceGroupDB` preference variable. This is the name of the database that contains the table named **PriceGroup**.
- **Promotion** corresponds to the `HS_PromotionDB` preference variable. This is the name of the database that contains the table named **Promotion**.
- **Review** corresponds to the `HS_ReviewDB` preference variable. This is the name of the database that contains the table named **Review**.
- **Shopping List** corresponds to the `HS_ShoppingListDB` preference variable. This is the name of the database that contains the table named **ShoppingList**.
- **Wish List** corresponds to the `HS_WishListDB` preference variable. This is the name of the database with the table named **WishList**.
- **Gift Certificate** corresponds to the `HS_GiftCertificateDB` preference variable. This is the name of the database that contains the table named **GiftCertificate**.

Database Names		help
Key Field Name	ID	(future feature)
Customer	<input type="text"/>	
Invoice	<input type="text"/>	
Item	<input type="text"/>	
Product	<input type="text"/>	
Admin User	<input type="text"/>	
Affiliate	<input type="text"/>	
Autoship Order Queue	<input type="text"/>	
Gift Certificate	<input type="text"/>	
Email Archive	<input type="text"/>	
Price Groups	<input type="text"/>	
Promotion	<input type="text"/>	
Review	<input type="text"/>	
Shopping List	<input type="text"/>	
Supplier	<input type="text"/>	
Wish List	<input type="text"/>	

Database names area of the store configuration form

Mail Server

- **SMTP Server Name** corresponds to the `HS_Host` preference variable. This is the name of the SMTP server that will be used to send email notifications. Users in a shared hosting environment may need to ask the hosting provider or server administrator for the SMTP server name to use if that information was not provided when the account was setup.

Mail Server		help
SMTP Server Name	<input type="text"/>	

Mail server area of the store configuration form

URLs

- **Path to Site** corresponds to the `HS_PathToSite` preference variable. This is the URL for the site that contains the store. In the case of sites that only contain the store, then this URL will be the path to the store. Include the protocol in the URL. For example: `http://www.hostedstore.com`.
- **Secure Path to Store** corresponds to the `HS_SecurePathToStore` preference variable. Three selections are available:
 1. A secure URL that corresponds to the licensed domain.
 2. A non-secure URL that corresponds to the licensed domain. This URL is provided for testing purposes or for stores that will not use an SSL certificate.
 3. **Not implemented yet.**

Use Shared Certificate Path - This option should be selected for shared hosting environments where the SSL certificate of the hosting provider will be used. Selecting this option requires that the proper URL is entered into the **Shared Certificate Path** field. The URL to use should be furnished by the hosting provider.

Important: There are potential security concerns when using a shared SSL certificate. If a shared SSL certificate will be used in a shared hosting environment, it is highly recommended to check with the hosting provider that potential security issues have been properly addressed to avoid any problems.

- **Shared Certificate Path** corresponds to the `HS_SharedCertificatePath` preference variable. The secure URL to use when a shared SSL certificate is to be used. The proper path to enter should be furnished by the hosting provider or server administrator.

The image shows a section of a configuration form titled "URLs" with a "help" link. It contains three input fields:

- Path to Site:** A text input field.
- Secure Path to Store:** A dropdown menu currently showing "Use Shared Certificate Path".
- Shared Certificate Path:** A text input field.

URLs area of the store configuration form

Date Format

Date can be formatted according to different regional standards by using the preference variables below.

- **Delimiter** corresponds to the `HS_DateDelimiter` preference variable. The character/symbol to separate the month, day, and year of a date.
- **First** corresponds to the `HS_DateFirst` preference variable. Sets which part of a date displays first.
- **Second** corresponds to the `HS_DateSecond` preference variable. Sets which part of a date displays second.
- **Third** corresponds to the `HS_DateThird` preference variable. Sets which part of a date displays third.
- **Digits in Year** corresponds to the `HS_DateDigits` preference variable. Sets the number of digits to display for the year. For example, 2002 will appear as 02 if the value is set to `2` and as 2002 if the value is set to `4`.
- **Remove Leading Zeroes** corresponds to the `HS_DateRemoveZeros` preference variable. Select `Yes` if leading zeros should be removed from months and days. For example, January will appear as 1 if the value is set to `Yes` and as 01 if the value is set to `No`.

Date Format [help](#)

Delimiter Slash (/)

First Month

Second Day

Third Year

Digits in Year Four

Remove Leading Zeroes Yes No

Date format area of the store configuration form

Time Format

Time can be formatted according to different regional standards by using the preference variables below.

- **Delimiter** corresponds to the `HS_TimeDelimiter` preference variable. The character/symbol to separate the hour, minutes, and seconds of a time.
- **Hours** corresponds to the `HS_TimeHours` preference variable. Sets whether or not to use a twelve or twenty-four hour clock. Setting to `12` will automatically append either AM or PM to the time.
- **Display Seconds** corresponds to the `HS_TimeSeconds` preference variable. Select `Yes` if seconds should be displayed in the time.
- **Remove Leading Zeroes** corresponds to the `HS_TimeRemoveZeros` preference variable. If set to `Yes`, then all one digit hours will appear without a leading 0. For example, the hour for one o'clock will appear as 1 if the value is set to `Yes` and as 01 if the value is set to `No`.

Time Format [help](#)

Delimiter Colon (:)

Hours 12

Display Seconds Yes No

Remove Leading Zeros Yes No

Time format area of the store configuration form

Currency Format

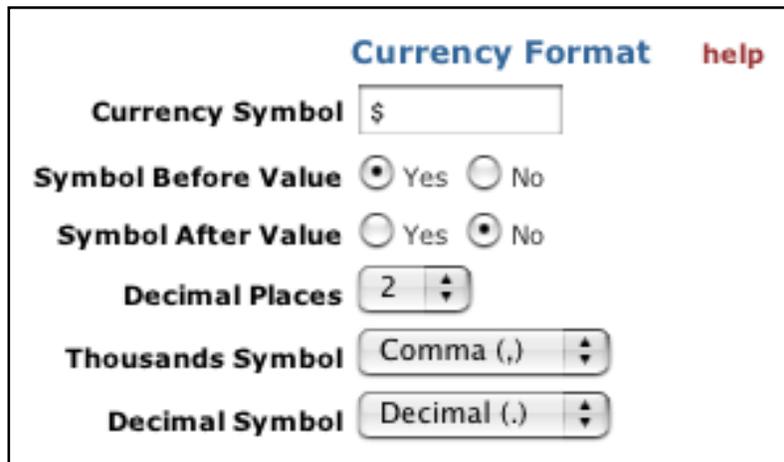
Currency can be formatted according to different regional standards by using the preference variables below.

- **Currency Symbol** corresponds to the `HS_CurrencySymbol` preference variable. The symbol used to designate currency values. The appropriate HTML entity code should be used for any symbol other than \$. HTML entity codes for common currency symbols:
 - `¢` for ¢ (cents).
 - `€` for € (euros).
 - `£` for £ (pounds).

- `¥` for ¥ (yen).

It is also possible to use an image for the currency symbol by using the appropriate code for an HTML `` tag. For example, entering `` would display the `euro.gif` image file from the `/images/` directory each time a monetary value is displayed.

- **Symbol Before Value** corresponds to the `HS_CurrencyBefore` preference variable. Select *Yes* if the currency symbol should be placed before the monetary value.
- **Symbol After Value** corresponds to the `HS_CurrencyAfter` preference variable. Select *Yes* if the currency symbol should be placed after the monetary value.
- **Decimal Places** corresponds to the `HS_CurrencyPrecision` preference variable. The number of decimal places to display in monetary values. This value also affects how many decimal places are used for monetary values in the database(s).
- **Thousands Symbol** corresponds to the `HS_CurrencyThousands` preference variable. The symbol used as a thousands place delimiter in monetary values.
- **Decimal Symbol** corresponds to the `HS_CurrencyDecimal` preference variable. The symbol used as a decimal place delimiter in monetary values.



The screenshot shows a configuration form titled "Currency Format" with a "help" link. It contains the following fields:

- Currency Symbol:** A text input field containing the dollar sign (\$) symbol.
- Symbol Before Value:** Radio buttons for "Yes" (selected) and "No".
- Symbol After Value:** Radio buttons for "Yes" and "No" (selected).
- Decimal Places:** A spinner control set to the value 2.
- Thousands Symbol:** A dropdown menu set to "Comma (,)" with up and down arrows.
- Decimal Symbol:** A dropdown menu set to "Decimal (.)" with up and down arrows.

Currency format area of the store configuration form

Percentage Format

Percentages can be formatted according to different precision requirements by using the preference variable below.

- **Decimal Places** corresponds to the `HS_PercentagePrecision` preference variable. The number of decimal places to display in percentage values. This value affects how percentages are displayed in reports generated in *Store Administration*. For example, fifty percent would be displayed as 50.00% if the value is set to `2` and as 50.000% if the value is set to `3`.



The screenshot shows a configuration form titled "Percentage Format" with a "help" link. It contains the following field:

- Decimal Places:** A spinner control set to the value 2.

Percentage format area of the store configuration form

Cookies

- **Cart Cookie Life** corresponds to the `HS_CartCookie` preference variable. The number of minutes the `HS_Cart` cookie will persist on the browser. This allows the customer to return to shopping on a different day. Leaving the

preference variable blank will expire the cart when the customer quits their browser. Some useful minute conversions:

- o 1440 minutes for 1 day.
- o 10080 minutes for 7 days.
- o 302400 minutes for 30 days.
- **Abandoned Cart** corresponds to the `HS_CartLife` preference variable. The number of days cart records in the **Item** table will persist after creation.
- **Incomplete Orders** corresponds to the `HS_OrderLife` preference variable. The number of days incomplete orders in the **Invoice** table will persist after creation.



The image shows a screenshot of a configuration form titled "Cookies" in blue text, with a "help" link in red text to its right. Below the title are three rows, each with a label and an input field:

- Cart Cookie Life (Minutes)** followed by an empty text input box.
- Abandoned Cart (Days)** followed by an empty text input box.
- Incomplete Orders (Days)** followed by an empty text input box.

Cookies area of the store configuration form

Shipping And Tax

- **Shipping Countries** corresponds to the `HS_ShipCountry` preference variable. This is a list of countries to which shipping is available if shipping is offered to a different group of countries than those contained in the included file for billing countries (`/hostedstore/includes/selectlists/countries.lasso`). Country names should be separated by two adjacent pipe characters (`| |`). The pipe character key is located in the shift position above the backslash (`\`) key (usually located above the return/enter key) on the keyboard.

Important: The value used for the country name should match the `value` parameter for the appropriate country's `<option>` tag in the included file for shipping countries (`/hostedstore/includes/selectlists/countries_shipping.lasso`). See the [Modifying HostedStore](#) > [Select Lists](#) > [Shipping Countries](#) section for more details about modifying the shipping country select list.

- **Fixed Shipping Rate** corresponds to the `HS_ShippingQuantityRate` preference variable. This is the per quantity charge for shipping. See the [Store Administration](#) > [Product Administration](#) section for more details.
- **Tax Shipping** corresponds to the `HS_TaxShipping` preference variable. Select **Yes** if shipping costs are subject to taxation.
- **Taxable States** corresponds to the `HS_TaxableStates` preference variable. A list of all states that are subject to taxation. State names should be separated by two adjacent pipe characters (`| |`). The pipe character key is located in the shift position above the backslash (`\`) key (usually located above the return/enter key) on the keyboard.
- **Tax Rates** corresponds to the `HS_TaxableStatesRates` preference variable. A list of all the tax rates for the states assigned to the `HS_TaxableStates` preference variable. Tax rates should be separated by two adjacent pipe characters (`| |`) and in the same order as the corresponding states. The pipe character key is located in the shift position above the backslash (`\`) key (usually located above the return/enter key) on the keyboard.

Shipping and Tax help

Shipping Countries

Fixed Shipping Rate

Tax Shipping Yes No

Taxable States

Tax Rates

Shipping and tax area of the store configuration form

Payment Methods Accepted

The following preference variables determine which payment method(s) will be accepted in the store.

- **Checks** corresponds to the `HS_AcceptChecks` preference variable. Select *Yes* if checks will be accepted as a payment method.
- **Check by Mail/Phone** corresponds to the `HS_AcceptOfflineChecks` preference variable. Select *Yes* if mailed checks or phoned check information will be accepted as a payment method. Note that this differs from the **Checks** option because check information is not gathered from the customer during checkout. This option could also be used for other payment methods like C.O.D. or money order.
- **Credit Cards** corresponds to the `HS_AcceptCreditCards` preference variable. Select *Yes* if credit cards will be accepted as a payment method.
- **PayPal** corresponds to the `HS_AcceptPayPal` preference variable. Select *Yes* if PayPal will be accepted as a payment method.
- **Purchase Orders** corresponds to the `HS_AcceptPurchaseOrders` preference variable. Select *Yes* if purchase orders will be accepted as a payment method. The purchase order payment option will only be displayed for customers that have been approved for purchase orders in *Customer Administration*. More details about approving customers for purchase orders can be found in the [Store Administration](#) > [Customer Administration](#) > [Add New Customer](#) section.
- **Default Purchase Order Eligibility** corresponds to the `HS_DefaultPOEligibility` preference variable. The selected value will be used as the value for the **ApprovedForPO** field in the **Customer** table when a new customer account is created. Select *Yes* if all new customers will be immediately eligible for using purchase orders as a payment method.

Payment Methods Accepted [help](#)

Checks Yes No

Check by Mail/Phone Yes No

Credit Cards Yes No

PayPal Yes No

Purchase Orders Yes No

Default Purchase Order Eligibility Yes No

Payment methods accepted area of the store configuration form

Payment Gateways

The following preference variables determine which payment gateway(s) will be used to process check and/or credit card payments.

- **Checks** corresponds to the `HS_CheckGatewayName` preference variable. The name of the gateway to use for check processing. Select the *Offline Processing* option if real-time check processing will not be used, but checks will be accepted for later offline processing. Select the *Other* option if a custom gateway processing script is to be used.
- **Credit Cards** corresponds to the `HS_GatewayName` preference variable. The name of the gateway to use for credit card processing. Select the *Offline Processing* option if real-time credit card processing will not be used, but credit cards will be accepted for later offline processing. Select the *Other* option if a custom gateway processing script is to be used.

Payment Gateway [help](#)

Checks

Credit Cards

Payment gateways area of the store configuration form

Check Gateway Parameters

- Ten preference variables are provided for passing parameters like login name, merchant ID, etc to the payment gateway. Each corresponds to the preference variables named `HS_CheckParam1` through `HS_CheckParam10`. The values to assign to the preference variables differ according to the payment gateway used. The [Built-In Forms > Check Processing](#) section provides the appropriate values, if any, to place in these fields for each payment gateway.

Check Gateway Parameters [help](#)

1

2

3

4

5

6

7

8

9

10

Check gateway parameters area of the store configuration form

Credit Card Gateway Parameters

- Ten preference variables are provided for passing parameters like login name, merchant ID, etc to the payment gateway. Each corresponds to the preference variables named `HS_CCParam1` through `HS_CCParam10`. The values to assign to the parameters differs according to the payment gateway used. The [Built-In Forms > Credit Card Processing](#) section provides the appropriate values, if any, to place in these fields for each payment gateway.

Credit Card Gateway Parameters [help](#)

1

2

3

4

5

6

7

8

9

10

Credit card gateway parameters area of the store configuration form

PayPal Parameters

- **Account Email** corresponds to the `HS_PayPalEmail` preference variable. The email address to which PayPal payments should be sent.

Paypal Parameters [help](#)

Account Email

PayPal parameters area of the store configuration form

Contact Information

All of these preference variables provide general contact information about the company running the store. These preference variables are mainly used as the signature for the default email messages.

- **Store Name** corresponds to the `HS_StoreName` preference variable.
- **Company Name** corresponds to the `HS_CompanyName` preference variable.
- **Address** corresponds to the `HS_CompanyAddress` preference variable.
- **City** corresponds to the `HS_CompanyCity` preference variable.
- **State/Province** corresponds to the `HS_CompanyState` preference variable.
- **Postal Code** corresponds to the `HS_CompanyPostalCode` preference variable.
- **Country** corresponds to the `HS_CompanyCountry` preference variable.
- **Phone** corresponds to the `HS_CompanyPhone` preference variable.
- **Fax** corresponds to the `HS_CompanyFax` preference variable.

Contact Information		help
Store Name	<input type="text"/>	
Company Name	<input type="text"/>	
Address	<input type="text"/>	
City	<input type="text"/>	
State/Province	<input type="text"/>	
Postal Code	<input type="text"/>	
Country	<input type="text"/>	
Phone	<input type="text"/>	
Fax	<input type="text"/>	

Contact information area of the store configuration form

Email Addresses

All of these preference variables determine which email addresses should receive emails sent by the system and by customers.

Important: Some mail servers may be configured with relay restrictions that only allow mail to be relayed from or to client email addresses. Check with the hosting provider or server administrator to determine if restrictions are in place. If so, then all of the email addresses listed below will need to be recognized as client email addresses by the mail server otherwise email sent from HostedStore will not be delivered.

- **Sales** corresponds to the `HS_SalesEmail` preference variable. This is the email address that will receive all notifications pertaining to orders: order placed, order canceled, etc.
- **Customer Service** corresponds to the `HS_CustomerServiceEmail` preference variable. This is the email address that will receive all customer service inquiries if the default *Help Center* customer service inquiry form is used.
- **Feedback** corresponds to the `HS_FeedbackEmail` preference variable. This is the email address that will receive all store feedback if the default *Help Center* content pages are used.
- **Webmaster** corresponds to the `HS_WebmasterEmail` preference variable. This is the email address that will receive all issues that should be reported to the webmaster if the default *Help Center* content pages are used.
- **Store Manager** corresponds to the `HS_StoreManagerEmail` preference variable. This is the email address that will receive all notices that a product is out of stock, that a customer has posted a product review, and other email notices. Note that even though this email address is required this feature does not have to be used. Email notification can be also set for individual products. Details about out of stock messages can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Proxy** corresponds to the `HS_ProxyEmail` preference variable. This is the email address that will be used to send emails for the email a friend feature on the product detail content page and to email wish list links in *Customer Account Administration*. An email address should be specified in this field if the mail server used is configured so that relaying is only allowed for emails where the to or from address is an account on the mail server. This situation may be encountered in shared hosting environments or in other environments where stringent relaying rules are in place. If these restrictions are not used for the mail server, then this field is optional.

Email Addresses		help
* Sales	<input type="text"/>	
* Customer Service	<input type="text"/>	
* Feedback	<input type="text"/>	
* Webmaster	<input type="text"/>	
* Store Manager	<input type="text"/>	
Proxy	<input type="text"/>	

Email addresses area of the store configuration form

Default Template

These preference variables are used in conjunction with the default store templates.

- **Template Name** corresponds to the `HS_TemplateName` preference variable. This is the name of the directory that contains the template that will be used for the store appearance. All templates must be stored in the `/hostedstore/templates/` directory. The templates provided with HostedStore begin with an `HS_` prefix followed by a number.
- **Page Title** corresponds to the `HS_PageTitle` preference variable. This preference variable is used by the default store templates as the default page title placed between the HTML `<title></title>` tags in the document head.
- **Keywords** corresponds to the `HS_MetaKeywords` preference variable. This preference variable is used by the default store templates as the default `content` parameter in the HTML `<meta name="keywords">` tag in the document head.
- **Description** corresponds to the `HS_MetaDescription` preference variable. This preference variable is used by the default store templates as the default `content` parameter in the HTML `<meta name="description">` tag in the document head.
- **Store Administration Template Name** corresponds to the `HS_AdminTemplateName` preference variable. This is the name of the directory that contains the template that will be used for the appearance of *Store Administration*. All templates must be stored in the `/hostedstore/adminstore/templates/` directory. The template provided with HostedStore is named `HS_1`. Use that template name if a custom template will not be used. Note that the provided template is internal to HostedStore so it will not be found in the `/hostedstore/adminstore/templates/` directory.

Default Template [help](#)

Template Name

Page Title

Keywords

Description

**Store Administration
Template Name**

Default template area of the store configuration form

Home Page

These preference variables determine what information appears on the default store home page content page.

- **Featured Items** corresponds to the `HS_FeaturedLimit` preference variable. This is the maximum number of featured products to display on the default home content page. Products can be flagged as featured in *Product Administration*. Details about featured products can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Favorites** corresponds to the `HS_DisplayFavorites` preference variable. This determines whether or not top selling products will be displayed on the default home content page.
- **Favorite Items** corresponds to the `HS_FavoritesLimit` preference variable. This is the maximum number of top selling products to display on the default home content page.

Home Page [help](#)

Featured Items

Favorites Yes No

Favorite Items

Home page area of the store configuration form

Product Search

- **Products Per Screen** corresponds to the `HS_MaxRecords` preference variable. Set this to the maximum number of products to display on the default product search results content page. If the number of matching products exceeds the value of `HS_MaxRecords`, then the user can navigate the found set with previous/next hyperlinks provided on the default product search results content page.
- **Statistics** corresponds to the `HS_UseStatistics` preference variable. Select *Yes* if the number of times a product is searched, viewed, and added to a cart is to be tracked. This preference variable may also be used in future versions of HostedStore to determine whether or not other store statistics are tracked. Tracking store statistics does add some overhead because additional database calls are necessary.



Product search area of the store configuration form

Products

- **Attributes** corresponds to the `HS_NumberOfAttributes` preference variable. This is the maximum number of attributes that will be used with products. Note that the largest value supported internally is 30. There will be some performance gains by setting this value appropriately (especially when adding/editing products in *Product Administration*), but the gains are not likely to be discernable. Details about product attributes can be found under the *Product Attributes* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Inventory Order Difference** corresponds to the `HS_InventoryOrder` preference variable. The number of units that should trigger notification to order inventory. If the difference between the quantity on hand and the build to quantity is larger than the value, then an order for the product is recommended. Details about the recommended inventory order report can be found in the [Store Administration](#) > [Reports](#) > [Date Independent Reports](#) > [Suggested Inventory Order](#) section.
- **Out of Stock Message** corresponds to the `HS_OutOfStockMessage` preference variable. The default message to display if a product is out of stock and cannot be ordered. HTML can be used. This value can be overridden for individual products. Details about product specific out of stock messages can be found under the *Inventory* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **EAD** corresponds to the `HS_UseEAD` preference variable. Select *Yes* if products that require downloading will be sold in the store. Details about EAD products can be found under the *Electronic Asset Delivery (EAD)* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Memberships** corresponds to the `HS_UseMemberships` preference variable. Select *Yes* if membership type products will be sold in the store. Details about membership products can be found under the *Memberships* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Email Signups/Renewals** corresponds to the `HS_EmailMembershipSales` preference variable. Select *Yes* if the membership email address is to receive notification when a membership product is sold in the store.
- **Membership Email Address** corresponds to the `HS_MembershipsEmail` preference variable. The email address that should receive notification when a membership is purchased in the store.

Important: Some mail servers may be configured with relay restrictions that only allow mail to be relayed from or to client email addresses. Check with the hosting provider or server administrator to determine if restrictions are in place. If so, then this email address will need to be recognized as a client email address by the mail server otherwise email sent from HostedStore will not be delivered.

- **Order Limits** corresponds to the `HS_UseOrderLimits` preference variable. Select *Yes* if products will have restrictions on the quantities that may be ordered. Details about quantity limits can be found under the *Quantity Limits* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Volume Pricing** corresponds to the `HS_UseVolumePricing` preference variable. Select *Yes* if products will have restrictions on the quantities that may be ordered. Details about volume pricing can be found under the *Pricing* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.

Products help

Attributes

Inventory Order Difference

Out of Stock Message

EAD Yes No

Memberships Yes No

Email Signups/Renewals Yes No

Membership Email Address

Order Limits Yes No

Volume Pricing Yes No

Products area of the store configuration form

Autoship Products

These preference variables are used to configure how autoship products will function. Details about autoship products can be found under the *Product Attributes* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.

- **Autoship Products** corresponds to the `HS_UseAutoShipOrders` preference variable. Select *Yes* if autoship products will be used.
- **Frequency Attribute** corresponds to the `HS_AutoShipFrequencyAttribute` preference variable. The name of the attribute whose value will be used to set the frequency at which an autoship order is generated.
- **Start Time Attribute** corresponds to the `HS_AutoShipStartTimeAttribute` preference variable. The name of the attribute whose value will be used to set the time at which an autoship order is first generated.
- **Order Action** corresponds to the `HS_AutoShipOrderAction` preference variable. Select *Process* if autoship generated orders should be automatically processed. Select *Pending* if autoship generated orders should be created with a pending status for manual processing at a later date.
- **Fail Address** corresponds to the `HS_AutoShipOrderFailEmail` preference variable. The email address that should receive notification when an autoship generated order fails due to an error.

Important: Some mail servers may be configured with relay restrictions that only allow mail to be relayed from or to client email addresses. Check with the hosting provider or server administrator to determine if restrictions are in place. If so, then this email address will need to be recognized as a client email address by the mail server otherwise email sent from HostedStore will not be delivered.

Autoship Products [help](#)

Autoship Products Yes No **(future feature)**

Frequency Attribute

Start Time Attribute

Order Action Process Pending

Fail Address

Autoship products area of the store configuration form

Product Images

The product image related code in the default content pages assumes all thumbnail, detail, and zoom (if used) images for products are the same size and use the same border pixel width. The following preference variables are used as the default parameters for the HTML `` tags:

- **Zoom Images** corresponds to the `HS_UseZoomImages` preference variable. Select *Yes* if zoom images will be used.
- **Thumbnail - Width** corresponds to the `HS_ThumbnailImageWidth` preference variable. The width of thumbnail images.
- **Thumbnail - Height** corresponds to the `HS_ThumbnailImageHeight` preference variable. The height of thumbnail images.
- **Thumbnail - Border** corresponds to the `HS_ThumbnailImageBorder` preference variable. The border size to use for thumbnail images.
- **Detail - Width** corresponds to the `HS_DetailImageWidth` preference variable. The width of detail images.
- **Detail - Height** corresponds to the `HS_DetailImageHeight` preference variable. The height of detail images.
- **Detail - Border** corresponds to the `HS_DetailImageBorder` preference variable. The border size to use for detail images.
- **Zoom - Width** corresponds to the `HS_ZoomImageWidth` preference variable. The width of zoom images.
- **Zoom - Height** corresponds to the `HS_ZoomImageHeight` preference variable. The height of zoom images.
- **Zoom - Border** corresponds to the `HS_ZoomImageBorder` preference variable. The border size to use for zoom images.

Product Images [help](#)

Zoom Images Yes No

	Width	Height	Border
Thumbnail	<input type="text"/>	<input type="text"/>	<input type="text"/>
Detail	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zoom	<input type="text"/>	<input type="text"/>	<input type="text"/>

Product images area of the store configuration form

Product Detail Page

These preference variables determine what information appears on the default product detail content page.

- **MSRP** corresponds to the `HS_DisplayMSRP` preference variable. Select *Yes* if the **MSRP** field from the product database should be displayed on the default product detail content page.
- **Sale Pricing** corresponds to the `HS_UseSalePrices` preference variable. Select *Yes* if sale pricing will be used with products.
- **Product Complements** corresponds to the `HS_UseComplements` preference variable. Select *Yes* if product complements are to be displayed on the default product detail content page.
- **Email a Friend** corresponds to the `HS_UseEmailFriend` preference variable. Select *Yes* to allow customers to email from the default product detail content page.
- **Use Restricted Products** corresponds to the `HS_UseRestrictedProducts` preference variable. Select *Yes* if some products will be restricted so that the product can be added to the cart only if another product is already in the cart. Details about restricted products can be found under the *Dependencies/Restrictions* heading of the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) > section.

The screenshot shows a configuration form titled "Product Detail Page" with a "help" link. It contains five rows, each with a preference variable name and two radio buttons labeled "Yes" and "No". The "No" radio button is selected for all five variables.

Preference Variable	Yes	No
MSRP	<input type="radio"/>	<input checked="" type="radio"/>
Sale Pricing	<input type="radio"/>	<input checked="" type="radio"/>
Product Complements	<input type="radio"/>	<input checked="" type="radio"/>
Email a Friend	<input type="radio"/>	<input checked="" type="radio"/>
Use Restricted Products	<input type="radio"/>	<input checked="" type="radio"/>

Product detail area of the store configuration form

Cart/Checkout Page

These preference variables determine which business rules are applied and what information appears on the default shopping cart and checkout content pages.

- **Enforce Inventory** corresponds to the `HS_EnforceInventory` preference variable. Select *Yes* if available inventory for some products will be verified during checkout. More information can be found under the *Inventory* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.
- **Quotes** corresponds to the `HS_UseQuotes` preference variable. Select *Yes* if customers are allowed to email a quote from the shopping cart detail page.
- **Days Valid** corresponds to the `HS_QuoteLife` preference variable. The number of days that email quotes are valid. Note that this value is meaningless to HostedStore internally. The number of days the quote is valid will appear in the emailed quote for informational purposes only. It is the responsibility of the store manager to verify and validate cart generated quotes.
- **Savable Carts** corresponds to the `HS_UseSavedCarts` preference variable. Select *Yes* if customers are allowed to save their shopping carts for later use. Note that item prices are not verified on saved carts so too high of a value for this preference variable may cause problems if product pricing changes frequently.
- **Saved Carts Expire** corresponds to the `HS_SavedCartDays` preference variable. The number of days that saved carts will be stored.
- **Address Books** corresponds to the `HS_UseAddressBooks` preference variable. Select *Yes* to allow customers to save and use multiple shipping addresses.
- **Require CC Security Code** corresponds to the `HS_UseSecurityCode` preference variable. Select *Yes* if the credit card security code is to be required for credit card purchases. The security code appears as a 3 or 4 digit number on the front of the credit card or in the signature line on the back of the credit card. The primary purpose of requiring this number is for verification that the card owner has the card in their possession when the purchase is made. Although there

is not a formula to validate the number, the entered number can be compared with the card owner's number if a dispute arises in the future. Some payment gateways will also verify the validity of the security code. This may or may not be a required value by the payment gateway.

- **Validate Credit Cards** corresponds to the `HS_ValidateCreditCards` preference variable. Select *Yes* if credit card numbers should be verified to be of the proper form. The validation only works for AMEX, MasterCard, and Visa credit cards so if another credit card is accepted this value should be set to *No*. Note that this does not validate whether or not a credit card account number is legitimate. It only verifies that the number is of the proper form.
- **Cancel Orders** corresponds to the `HS_CancelOrders` preference variable. Select *Yes* if customers will be allowed to cancel orders after completing the purchase process in the store. Note that only orders that have not had a status change will be eligible for cancellation.
- **Require Terms and Conditions** corresponds to the `HS_UseTermsAndConditions` preference variable. Select *Yes* if customers will have to agree to the store terms and conditions before being allowed to checkout and complete their order.

The screenshot shows a configuration form titled "Cart/Checkout Page" with a "help" link. The form contains the following settings:

- Enforce Inventory**: Yes No
- Quotes**: Yes No
- Days Valid**:
- Savable Carts**: Yes No
- Saved Carts Expire (Days)**:
- Address Books**: Yes No
- Require CC Security Code**: Yes No
- Validate Credit Cards**: Yes No
- Cancel Orders**: Yes No
- Require Terms And Conditions**: Yes No

Cart/checkout page area of the store configuration form

Newsletter

These preference variables are used to configure the store to use a store newsletter.

- **Use Newsletters** corresponds to the `HS_Newsletter` preference variable. Select *Yes* if newsletter subscriptions will be supported. If the default content pages are used, customers are prompted during account creation and in the customer **Account Administration** area if a newsletter subscription is wanted.
- **Subscribe Address** corresponds to the `HS_SubscribeEmail` preference variable. Enter the address that is to receive newsletter subscribe requests.

Important: Some mail servers may be configured with relay restrictions that only allow mail to be relayed from or to client email addresses. Check with the hosting provider or server administrator to determine if restrictions are in place. If so, then this email address will need to be recognized as a client email address by the mail server otherwise email sent from HostedStore will not be delivered.

- **Subscribe Subject** corresponds to the `HS_SubscribeSubject` preference variable. Enter the subject of the email sent as a newsletter subscribe request.

- **Unsubscribe Address** corresponds to the `HS_UnsubscribeEmail` preference variable. Enter the address that is to receive newsletter unsubscribe requests.
- **Unsubscribe Subject** corresponds to the `HS_UnsubscribeSubject` preference variable. Enter the subject of the email sent as a newsletter unsubscribe request.

The image shows a configuration form for newsletters. At the top right, there is a link labeled "Newsletter help". Below this, the "Use Newsletters" option is set to "No" with a selected radio button. There are four text input fields stacked vertically, labeled "Subscribe Address", "Subscribe Subject", "Unsubscribe Address", and "Unsubscribe Subject".

Newsletter area of the store configuration form

Additional Modules

These preference variables are used to configure the store to use additional features that require additional database tables.

- **Affiliate Credits** corresponds to the `HS_AffiliateCredits` preference variable. Select *Yes* if affiliates are to receive credit for each sale completed with their affiliate ID.
- **Affiliate ID** corresponds to the `HS_AffiliateIDName` preference variable. Enter the name of the parameter that will be used to pass affiliate IDs to the store.
- **Promotions** corresponds to the `HS_UsePromotions` preference variable. Select *Yes* to allow for the use of special promotion codes that offer discounts to customers.
- **Promotion ID** corresponds to the `HS_PromotionIDName` preference variable. Enter the name of the parameter that will be used to pass promotion IDs to the store
- **Product Reviews** corresponds to the `HS_UseProductReviews` preference variable. Select *Yes* to allow customers to review products on the default product detail content page.
- **Login Required to Review** corresponds to the `HS_LoginForProductReviews` preference variable. Select *Yes* if customers must log in prior to posting product reviews. This feature ensures that only prior store customers are posting reviews. Note that this feature does not verify that the customer has purchased the reviewed product from the store.
- **Auto-Publish Reviews** corresponds to the `HS_AutoPublishReviews` preference variable. Select *Yes* if customer reviews will automatically be published for viewing by other customers. Select *No* if customer reviews must be reviewed by the store administrator before being published for viewing by other customers.
- **Review Email Notification** corresponds to the `HS_EmailNotifyReviews` preference variable. Select *Yes* if the store manager should receive an email notification when a customer posts a review. Email notification is sent to the email address set in the `HS_StoreManagerEmail` preference variable.
- **Shopping Lists** corresponds to the `HS_UseShoppingLists` preference variable. Select *Yes* to allow customers to save previous orders and add to their cart quickly.
- **Wish Lists** corresponds to the `HS_UseWishLists` preference variable. Select *Yes* to allow customers to save products in a wish list.
- **Store Admin Users** corresponds to the `HS_UseAdminUsers` preference variable. Select *Yes* if other users other than the store administrator will be granted access to *Store Administration*. Details about admin users can be found in the [Store Administration > Admin User Administration](#) section.
- **Price Groups** corresponds to the `HS_UsePriceGroups` preference variable. Select *Yes* if customers will be assigned to different price groups. Details about price groups can be found in the [Store Administration > Price Group Administration](#) section.
- **Default Price Group** corresponds to the `HS_DefaultPriceGroup` preference variable. Enter the name of the price group to which new customers should be assigned.

- **Gift Certificates** corresponds to the `HS_UseGiftCertificates` preference variable. Select *Yes* if gift certificates will be sold. Details about gift certificate products can be found under the *Gift Certificates* heading in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.

Additional Modules [help](#)

Affiliate Credits Yes No

Affiliate ID

Promotions Yes No

Promotion ID

Product Reviews Yes No

Login Required to Review Yes No

Auto-Publish Reviews Yes No

Review Email Notification Yes No

Shopping Lists Yes No

Wish Lists Yes No

Store Admin Users Yes No **(future feature)**

Price Groups Yes No

Default Price Group

Gift Certificates Yes No **(future feature)**

Additional modules area of the store configuration form

Initial Modifications

While HostedStore is designed to run "out of the box," modifications to some of the files will be necessary to match the store policies. A suggested list of the files which, at a minimum, should be modified before opening the store for public access is provided below:

- The contents of the *Help Center* related content pages (located in the `/hostedstore/helpcenter/html/` directory) should be modified to reflect store policies. The *Help Center* content pages provide customers with easy access to store policies, frequently asked questions, explanations of shipping and tax charges, etc. Because store policies vary between each store, many of the supplied *Help Center* content pages are blank by default. The blank pages are provided as a way to suggest the type of content that would be useful to have in the *Help Center*. The pages that do contain content should be reviewed and altered or removed as necessary. The blank pages should be completed or removed from the *Help Center* menu if not needed. Details about the *Help Center* content pages can be found in the [Modifying HostedStore > Content Pages > Help Center](#) section.
- Select lists (located in the `/hostedstore/includes/selectlists/` directory) are used to provide selections for credit cards accepted, states/provinces/regions, billing countries, shipping countries, etc. The select lists should be reviewed and altered as necessary. Details about the select lists can be found in the [Modifying HostedStore > Select Lists](#) section.

Important: Make a backup of copy of all of the original files in the `/hostedstore/` directory before making any modifications. Writing code is often a trial and error process. A backup copy will ensure that a working copy can always be returned to for reference.

Upgrading

This section contains instructions for upgrading HostedStore from prior versions. Instructions are provided below in reverse chronological order.

HostedStore Version Numbering

Most software companies typically use a version number in the form *MajorVersionNumber.MinorVersionNumber.ReleaseNumber*. The **major version number** changes when the software company arbitrarily decides that the release has enough significant changes to warrant a new major version number. An incremental major version number usually equates to a paid upgrade. The **minor version number** changes when the software company arbitrarily decides that the release has enough new features to warrant a new minor version number. A change in the minor version number may or may not equate to a paid upgrade. The **release number** is used as a catchall for releases that fix bugs, offer minor improvements, etc, yet do not warrant an increase in the major and minor version numbers.

HostedStore version numbering does not follow that system. Each release of HostedStore is assigned a version number corresponding to the release date in *YYYY.MM.DD* format. There are several reasons why this is done:

- We thrive on doing things against the status quo.
- Version numbering is very arbitrary. Often it does not reflect the actual features of a release. In fact, major and minor version numbers are often incremented with releases to force paid upgrades depending upon where the software company finds itself in its product sales cycle.
- It can be difficult to remember exactly what dot release is currently installed when checking for updates. A date-based version may provide a more valuable reference in terms of when the last version was installed.
- All future upgrades are free for purchased HostedStore licenses so the typical arbitrary software version numbering system really has no meaning because upgrades are not tied to our product sales cycle.

Sections

- [2002.12.01 to 2003.06.16](#)
- [2002.11.15 to 2002.12.01](#)
- [2002.07.15 to 2002.11.15](#)
- [2002.07.01 to 2002.07.15](#)
- [Pre-2002.07.01 to 2002.07.01](#)

2002.12.01 to 2003.06.16

The following changes are necessary to incorporate new functionality added with version 2003.06.16. It is recommended to make the changes to prevent complications when using future versions.

Important: As with any upgrade, make backups of the current database and format files before proceeding with the upgrade steps outlined below. Something can always go wrong...

Database Modifications

- Add a new column to the **Invoice** table by issuing the following statements in the MySQL client:

```
ALTER TABLE Invoice ADD COLUMN TermsAndConditions CHAR(1);
```

- Add a new column to the **Product** table by issuing the following statements in the MySQL client:

```
ALTER TABLE Product ADD COLUMN PriceGroup TEXT AFTER RestrictTo;
```

File Modifications

There are three approaches to updating content pages and email format files:

- If modifications have not been made to any of the content pages or email format files, then upgrading the files is simply a matter of renaming the current */hostedstore/* directory and moving the new */hostedstore/* directory into the root Web serving directory. Any files that have been added into the old */hostedstore/* directory (product images, for example) will need to be copied to the appropriate location in the new */hostedstore/* directory.
- If a log of changes has been made (as recommended in the [Modifying HostedStore](#) section) and there are relatively few changes, then it may be an easier process to rename the current */hostedstore/* directory, move the new */hostedstore/* directory into the root Web serving directory, and then apply the changes from the log to the new file. Any files that have been added into the old */hostedstore/* directory (product images, for example) will need to be copied to the appropriate location in the new */hostedstore/* directory.
- Apply the changes as outlined below to the content pages and email format files in the */hostedstore/* directory.

A reasonable effort has been made to outline the incremental changes made to aid those that have made modifications to any of the content pages. This list is by no means guaranteed to be exhaustive so if something was left out please make a documentation modification request in the support area of the HostedStore Web site (www.hostedstore.com/support/).

- */hostedstore/includes/calculations/calculate_tax.lasso*, line 101

WAS:

```
$HSP_CartTaxableTotal+=$HSP_CartShippingCost;
```

NOW:

```
$HSP_CartShippingCost->SetFormat:-Precision=$HS_CurrencyPrecision;  
$HSP_CartTaxableTotal+=(Decimal:(String:$HSP_CartShippingCost));
```

- */hostedstore/includes/calculations/calculate_tax.lasso*, line 143

INSERT:

```
// Set display formatting.  
$HSP_CartSubTotal->SetFormat:-Precision=$HS_CurrencyPrecision;  
$HSP_CartShippingCost->SetFormat:-Precision=$HS_CurrencyPrecision;  
$HSP_TempTax->SetFormat:-Precision=$HS_CurrencyPrecision;
```

- */hostedstore/includes/calculations/calculate_tax.lasso*, line 148

WAS:

```
$HSP_TempTotalWithTax=$HSP_CartSubTotal  
    +$HSP_CartShippingCost  
    +$HSP_TempTax;
```

NOW:

```
$HSP_TempTotalWithTax=(Decimal:(String:$HSP_CartSubTotal))  
    +(Decimal:(String:$HSP_CartShippingCost))  
    +(Decimal:(String:$HSP_TempTax));
```

- */hostedstore/includes/navigation/browse_products.lasso*, line 10

WAS:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?  
-ResponseLassoApp=search.lasso&category=[Field:'Category']">  
[Field:'Category']</a><br>
```

NOW:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?  
-ResponseLassoApp=search.lasso&category=[Encode_URL:(Field:'Category')] ">  
[Field:'Category']</a><br>
```

- */hostedstore/storefront/html/cart_display.lasso*, line 61

WAS:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?  
-ResponseLassoApp=cart.lasso" method="post">
```

NOW:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?  
-ResponseLassoApp=cart.lasso&preventcache=[Lasso_UniqueID]" method="post">
```

- */hostedstore/storefront/html/cart_display.lasso*, line 63

WAS:

```
<form action="[$HS_PathToStore]/cart.lasso" method="post">
```

NOW:

```
<form action="[$HS_PathToStore]/cart.lasso?preventcache=[Lasso_UniqueID]"  
method="post">
```

- */hostedstore/storefront/html/payment_modify.lasso*, line 160

INSERT:

```
[If:$HS_UseTermsAndConditions=='Y']  
<tr><td colspan="2" class="subtitle">Terms and Conditions</td></tr>  
<tr>  
    <td> </td>  
    <td align="left" class="med">
```

```



```

- */hostedstore/storefront/html/shipping_info.lasso*, line 24

WAS:

```

<form action="[$HS_PathToStore]/HostedStore.LassoApp?
-ResponseLassoApp=checkout.lasso" method="post">

```

NOW:

```

<form action="[$HS_SecurePathToStore]/HostedStore.LassoApp?
-ResponseLassoApp=checkout.lasso&preventcache=[Lasso_UniqueID]" method="post">

```

- */hostedstore/storefront/html/shipping_info.lasso*, line 26

WAS:

```

<form action="[$HS_PathToStore]/checkout.lasso" method="post">

```

NOW:

```

<form action="[$HS_SecurePathToStore]/checkout.lasso?
preventcache=[Lasso_UniqueID]" method="post">

```

- */hostedstore/storefront/html/shipping_info.lasso*, line 314

INSERT:

```

[If:$HS_UseTermsAndConditions=='Y']
<tr><td colspan="2" class="subtitle">Terms and Conditions</td></tr>
<tr>
<td> </td>
<td align="left" class="med">


```

- */hostedstore/templates/HS_1/ss.css*, multiple changes
- */hostedstore/templates/HS_2/ss.css*, multiple changes
- */hostedstore/templates/HS_3/ss.css*, multiple changes
- */hostedstore/templates/HS_4/ss.css*, multiple changes
- */hostedstore/templates/HS_5/ss.css*, multiple changes
- */hostedstore/templates/HS_6/ss.css*, multiple changes
- */hostedstore/templates/HS_7/ss.css*, multiple changes

GLOBAL MODIFICATION:

Each occurrence of a space followed by a colon (:) should be replaced with just a colon.

Directory Additions

A new directory is needed with this version that was not used in prior versions.

- A */hostedstore/adminstore/report/* directory should be created with permissions that allow Lasso to write to the directory.

File Additions

Several new files are provided with this version that were not components of prior versions. Listed below are all of the files that should be copied from the new install files to the appropriate location:

- The */hostedstore/admncustomer/html/billingaddress.lasso* file should be added.
- The */hostedstore/storefront/gateways/cc_authorizenet_aim.lasso* file should be added.
- The */hostedstore/storefront/gateways/cc_verisign_payflowpro.lasso* file should be added.
- The */hostedstore/storefront/gateways/chk_authorizenet_aim.lasso* file should be added.
- The */hostedstore/storefront/gateways/chk_verisign_payflowpro.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/adminstore_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/adminstore_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/adminuser_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/adminuser_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/affiliate_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/affiliate_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/affiliatepayouthistory_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/affiliatepayouthistory_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/application_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/application_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/autoship_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/autoship_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/autoshipsweep_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/autoshipsweep_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/batch_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/batch_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/customer_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/customer_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/image_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/image_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/index_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/index_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/inventory_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/inventory_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/maintenance_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/maintenance_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualcheckout_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualcheckout_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualorder_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualorder_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualreturn_checkout_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualreturn_checkout_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualreturn_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/manualreturn_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/menu_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/menu_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/order_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/order_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/ordermanager_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/ordermanager_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/price_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/price_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/priceglobaladjust_entry.lasso* file should be added.

- The */hostedstore/thirdparty/hooks/adminstore/priceglobaladjust_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/pricegroup_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/pricegroup_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/product_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/product_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/promotion_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/promotion_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/review_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/review_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/supplier_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/supplier_exit.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/upload_entry.lasso* file should be added.
- The */hostedstore/thirdparty/hooks/adminstore/upload_exit.lasso* file should be added.

2002.11.15 to 2002.12.01

The following changes are necessary to incorporate new functionality added with version 2002.12.01. It is recommended to make the changes to prevent complications when using future versions.

Important: As with any upgrade, make backups of the current database and format files before proceeding with the upgrade steps outlined below. Something can always go wrong...

Database Modifications

- Add a new column to the **Item** table by issuing the following statement in the MySQL client:

```
ALTER TABLE Item ADD COLUMN TaxClass TEXT AFTER Taxable;
```

- Add a new column to the **Product** table by issuing the following statement in the MySQL client:

```
ALTER TABLE Product ADD COLUMN TaxClass TEXT AFTER Taxable;
```

File Modifications

No file modifications are necessary for this version.

File Additions

No file additions are necessary for this version.

2002.07.15 to 2002.11.15

The following changes are necessary to incorporate new functionality added with version 2002.11.15. It is recommended to make the changes to prevent complications when using future versions.

Important: As with any upgrade, make backups of the current database and format files before proceeding with the upgrade steps outlined below. Something can always go wrong...

Database Modifications

- Add new columns to the **AdminUser** table by issuing the following statements in the MySQL client:

```
ALTER TABLE AdminUser ADD COLUMN AdminGroup TEXT DEFAULT NULL AFTER Active;
```

```
ALTER TABLE AdminUser ADD COLUMN FirstName TEXT DEFAULT NULL AFTER AdminGroup;
```

```
ALTER TABLE AdminUser ADD COLUMN LastName TEXT DEFAULT NULL AFTER FirstName;
```

```
ALTER TABLE AdminUser ADD COLUMN Notes TEXT DEFAULT NULL AFTER Password;
```

- Add a new column to the **Customer** table by issuing the following statement in the MySQL client:

```
ALTER TABLE Customer ADD COLUMN EmailGroups TEXT AFTER ApprovedForPO;
```

- Add new columns to the **Invoice** table by issuing the following statements in the MySQL client:

```
ALTER TABLE Invoice ADD COLUMN AdminUserID BIGINT(20) UNSIGNED DEFAULT NULL AFTER CustomerID;
```

```
ALTER TABLE Invoice ADD COLUMN AutoShipOrderQueueID BIGINT(20) UNSIGNED NOT NULL AFTER TempID;
```

```
ALTER TABLE Invoice ADD COLUMN TaxLevel1 DECIMAL(10,2) DEFAULT NULL AFTER SubTotal;
```

```
ALTER TABLE Invoice ADD COLUMN TaxLevel2 DECIMAL(10,2) DEFAULT NULL AFTER TaxLevel1;
```

```
ALTER TABLE Invoice ADD COLUMN TaxLevel3 DECIMAL(10,2) DEFAULT NULL AFTER TaxLevel2;
```

```
ALTER TABLE Invoice ADD COLUMN AffiliateAdjustment DECIMAL(10,2) DEFAULT NULL AFTER SettlementAdjustment;
```

```
ALTER TABLE Invoice ADD COLUMN AffiliateTheoreticalPayout DECIMAL(10,2) DEFAULT NULL AFTER IPAddress;
```

```
ALTER TABLE Invoice ADD COLUMN AffiliatePlan TEXT AFTER AffiliateTheoreticalPayout;
```

- Add new columns to the **Item** table by issuing the following statements in the MySQL client:

```
ALTER TABLE Item ADD COLUMN ShadowProducts TEXT AFTER ProductID;
```

```
ALTER TABLE Item ADD COLUMN SupplierID VARCHAR(23) DEFAULT NULL AFTER Restricted;
```

- Add new columns to the **Product** table by issuing the following statements in the MySQL client:

```
ALTER TABLE Product ADD COLUMN ShadowProducts TEXT AFTER ProductID;
```

```
ALTER TABLE Product ADD COLUMN SupplierID BIGINT(20) DEFAULT NULL AFTER  
ContentPage;
```

```
ALTER TABLE Product ADD COLUMN SupplierNotify CHAR(1) DEFAULT NULL AFTER  
SupplierID;
```

- Add a new **AutoShipOrderQueue** table by issuing the following statement in the MySQL client:

```
CREATE TABLE AutoShipOrderQueue (  
ID BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT  
CustomerID BIGINT(20) UNSIGNED DEFAULT NULL,  
OrderID BIGINT(20) UNSIGNED DEFAULT NULL,  
DateOrdered DATETIME DEFAULT NULL,  
Active CHAR(1) DEFAULT NULL,  
DateActivated DATETIME DEFAULT NULL,  
DateDeactivated DATETIME DEFAULT NULL,  
Frequency TINYINT(3) UNSIGNED DEFAULT NULL,  
TimesExecuted TINYINT(3) UNSIGNED DEFAULT NULL,  
RestrictedDuration CHAR(1) DEFAULT NULL,  
Duration TINYINT(3) DEFAULT NULL,  
NextOrderDate DATE DEFAULT NULL,  
OrderAction TEXT,  
OrderHistory TEXT,  
Notes TEXT,  
AuditTrail TEXT,  
Item TEXT,  
FirstName TEXT,  
LastName TEXT,  
Company TEXT,  
Address1 TEXT,  
Address2 TEXT,  
City TEXT,  
State TEXT,  
Postal TEXT,  
Country TEXT,  
Phone TEXT,  
Fax TEXT,  
Email TEXT,  
PaymentType TEXT,  
PONumber TEXT,  
Account TEXT,  
Routing TEXT,  
CheckNumber INT(11) DEFAULT NULL,  
CheckSignature TEXT,  
Expiration VARCHAR(7) DEFAULT NULL,  
DateCreated DATETIME DEFAULT NULL,  
DateModified DATETIME DEFAULT NULL,  
PRIMARY KEY (ID)  
) TYPE=MyISAM;
```

- Add a new **EmailArchive** table by issuing the following statement in the MySQL client:

```
CREATE TABLE EmailArchive (  
ID BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
Type CHAR(2) DEFAULT NULL,
```

```

EmailGroup TEXT,
CustomerID BIGINT(20) UNSIGNED DEFAULT NULL,
HeaderTo TEXT,
HeaderCC TEXT,
HeaderBCC TEXT,
HeaderFrom TEXT,
HeaderSubject TEXT,
Body TEXT,
Message TEXT,
ReadByIDs TEXT,
SentByID BIGINT(20) UNSIGNED,
DateCreated DATETIME DEFAULT NULL,
DateModified DATETIME DEFAULT NULL,
PRIMARY KEY (ID)
) TYPE=MyISAM;

```

- Add a new **Supplier** table by issuing the following statement in the MySQL client:

```

CREATE TABLE Supplier (
ID BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
Status CHAR(1) DEFAULT NULL,
FirstName TEXT,
LastName TEXT,
Company TEXT,
Address1 TEXT,
Address2 TEXT,
City TEXT,
State TEXT,
Postal TEXT,
Country TEXT,
Phone TEXT,
Fax TEXT,
Email TEXT,
URL TEXT,
Password TEXT,
AllowLogin CHAR(1) DEFAULT NULL,
NotifyEmail TEXT,
NotifyEmailTemplate TEXT,
SupplierAdminTemplate TEXT,
CustomerEmailTemplate TEXT,
StoreEmailTemplate TEXT,
StoreEmailAddress TEXT,
OrderQuantity INT(11) UNSIGNED DEFAULT NULL,
OrderTotal DECIMAL(15,2) DEFAULT NULL,
AdminComments TEXT,
DateCreated DATETIME DEFAULT NULL,
DateModified DATETIME DEFAULT NULL,
PRIMARY KEY (ID),
KEY EmailPassword (Email(20),Password(8))
) TYPE=MyISAM;

```

File Modifications

There are three approaches to updating content pages and email format files:

- If modifications have not been made to any of the content pages or email format files, then upgrading the files is simply a matter of renaming the current `/hostedstore/` directory and moving the new `/hostedstore/` directory into the root Web serving directory. Any files that have been added into the old `/hostedstore/` directory (product images, for example) will need to be copied to the appropriate location in the new `/hostedstore/` directory.
- If a log of changes has been made (as recommended in the [Modifying HostedStore](#) section) and there are relatively few changes, then it may be an easier process to rename the current `/hostedstore/` directory, move the new `/hostedstore/` directory into the root Web serving directory, and then apply the changes from the log to the new file. Any files that have been added into the old `/hostedstore/` directory (product images, for example) will need to be copied to the appropriate location in the new `/hostedstore/` directory.
- Apply the changes as outlined below to the content pages and email format files in the `/hostedstore/` directory.

A reasonable effort has been made to outline the incremental changes made to aid those that have made modifications to any of the content pages. This list is by no means guaranteed to be exhaustive so if something was left out please make a documentation modification request in the support area of the HostedStore Web site (www.hostedstore.com/support/).

- `/hostedstore/helpcenter/html/customerserviceform.lasso`, line 11

WAS:

```
<form action="HostedStore.LassoApp?
-ResponseLassoApp=helpcenter/customerservice.lasso" method="post">
```

NOW:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?
-ResponseLassoApp=helpcenter/customerservice.lasso" method="post">
```

- `/hostedstore/storefront/html/cart_display.lasso`, line 194

WAS:

```
<td valign="top" align="center"><input type="text"
name="[$HSP_Cart->Get:1->Get:Loop_Count->Get:1]"
value="[$HSP_Cart->Get:1->Get:Loop_Count->Get:8]" size="3">
```

NOW:

```
<td valign="top" align="center"><input type="text"
name="[$HSP_Cart->Get:1->Get:Loop_Count->Get:1]"
value="[$HSP_Cart->Get:1->Get:Loop_Count->Get:8]" size="3"
maxlength="5">
```

- `/hostedstore/storefront/html/cart_recoverform.lasso`, line 43

WAS:

```
<td align="right" class="label">Name</td>
```

NOW:

```
<td align="right" class="label">Cart Name</td>
```

- `/hostedstore/storefront/html/cart_saveform.lasso`, line 47

WAS:

```
<td align="right" class="label">Name</td>
```

NOW:

```
<td align="right" class="label">Cart Name</td>
```

- */hostedstore/storefront/html/index.lasso*, line 13

WAS:

```

```

NOW:

```
[Inline: -Username=$HS_Username,
        -Password=$HS_Password,
  [If: (File_Exists: ($HS_FilePathToStore+' /images/products/thumbnail/'
+($HSP_Featured->Get:Loop_Count->Get:2)+' .jpg' ))]
  
  [Else]
  
[/If]
/Inline]
```

- */hostedstore/storefront/html/product_detail.lasso*, line 84

WAS:

```
[/If]
```

NOW:

```
<br>
[/If]
```

- */hostedstore/storefront/html/search_subcategories.lasso*, line 11

WAS:

```
<a href="[ $HS_PathToStore ]
-ResponseLassoApp=search.lasso&subcategory=
[ $HSP_SubCategories->Get:Loop_Count->Get:1 ]">
```

NOW:

```
<a href="[ $HS_PathToStore ]/HostedStore.LassoApp?
-ResponseLassoApp=search.lasso&subcategory=
[Encode_StrictURL: ($HSP_SubCategories->Get:Loop_Count->Get:1)]">
```

- */hostedstore/storefront/gateways/cc_bluepay.lasso*, line 5

WAS:

```
<input type="hidden" name="CC_Expires" value="[ $HSP_Processing->Get:3-
>Get:4 ]/[ $HSP_Processing->Get:3->Get:5->SubString:3,4 ]">
```

NOW:

```
<input type="hidden" name="CC_Expires" value="[ $HSP_Processing->Get:3-
>Get:4 ]/[ $HSP_Processing->Get:3->Get:5->SubString:3 ]">
```

- */hostedstore/storefront/gateways/cc_iongate.lasso*, line 12

WAS:

```
<input type="hidden" name="EXPIRES" value="[$HSP_Processing->Get:3->Get:4]
[$HSP_Processing->Get:3->Get:5->SubString:3,4]">
```

NOW:

```
<input type="hidden" name="EXPIRES" value="[$HSP_Processing->Get:3->Get:4]
[$HSP_Processing->Get:3->Get:5->SubString:3]">
```

- */hostedstore/storefront/gateways/cc_psigate.lasso*, line 14

WAS:

```
<input type="hidden" name="ExpYear" value="[$HSP_Processing->Get:3->Get:5-
>SubString:3,4]">
```

NOW:

```
<input type="hidden" name="ExpYear" value="[$HSP_Processing->Get:3->Get:5-
>SubString:3]">
```

- */hostedstore/storefront/gateways/cc_securepay.lasso*, line 17

WAS:

```
<input type="hidden" name="YEAR" value="[$HSP_Processing->Get:3->Get:5-
>SubString:3,4]">
```

NOW:

```
<input type="hidden" name="YEAR" value="[$HSP_Processing->Get:3->Get:5-
>SubString:3]">
```

- */hostedstore/storefront/gateways/verisign_payflowlink.lasso*, line 12

WAS:

```
<input type="hidden" name="EXPDATE" value="[$HSP_Processing->Get:3-
>Get:4]/[$HSP_Processing->Get:3->Get:5->SubString:3,4]">
```

NOW:

```
<input type="hidden" name="EXPDATE" value="[$HSP_Processing->Get:3-
>Get:4]/[$HSP_Processing->Get:3->Get:5->SubString:3]">
```

File Additions

Several new files are provided with this version that were not components of prior versions. Listed below are all of the files that should be copied from the new install files to the appropriate location:

- The */hostedstore/adminstore/email/email_supplier.lasso* file should be added.
- The */hostedstore/adminstore/reporttemplate_bottom.lasso* file should be added.
- The */hostedstore/adminstore/reporttemplate_top.lasso* file should be added.
- The */hostedstore/adminstore/selectlists/customer_emailgroups.lasso* file should be added.
- The */hostedstore/adminstore/templates/* directory and all of its files should be added.
- The */hostedstore/adminsupplier/* directory and all of its files should be added.
- The */hostedstore/thirdparty/hooks/admincustomer/* directory and all of its files should be added.

2002.07.01 to 2002.07.15

The following changes are necessary to incorporate new functionality added with version 2002.07.15. It is recommended to make the changes to prevent complications when using future versions.

Important: As with any upgrade, make backups of the current database and format files before proceeding with the upgrade steps outlined below. Something can always go wrong...

Database modifications

- Change the **Customer** table key field name from **CustomerID** to **ID** by issuing the following statement in the MySQL client:

```
ALTER TABLE Customer CHANGE COLUMN CustomerID ID BIGINT(20) UNSIGNED NOT NULL  
AUTO_INCREMENT;
```

- Add a field with the following specifications to the **Product** table to support forced search results ranking:

```
SearchRank TINYINT(2) DEFAULT NULL
```

The following statement can be used in the MySQL client to add the field:

```
ALTER TABLE Product ADD COLUMN SearchRank TINYINT(2) DEFAULT NULL AFTER  
Manufacturer;
```

- If saved carts will be used, add fields with the following specifications to the **Item** table:

```
CartName VARCHAR(80) DEFAULT NULL  
CartEmail VARCHAR(80) DEFAULT NULL  
CartExpires DATE DEFAULT NULL
```

The following statements can be used in the MySQL client to add the fields:

```
ALTER TABLE Item ADD COLUMN CartName VARCHAR(80) DEFAULT NULL AFTER  
WishListOwnerID;  
ALTER TABLE Item ADD COLUMN CartEmail VARCHAR(80) DEFAULT NULL AFTER CartName;  
ALTER TABLE Item ADD COLUMN CartExpires DATE DEFAULT NULL AFTER CartEmail;
```

Unless there is a compelling reason not to, it is recommended that these optional fields are added to the **Customer** table even if saved carts will not be used immediately. By keeping all database tables in their most current form there will be a reduction in possible future issues if saved carts are eventually used.

- If price groups will be used, add a table named **PriceGroup** to the appropriate database using the following statement in the MySQL client:

```
CREATE TABLE PriceGroup (  
  ID BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  Active char(1) DEFAULT NULL,  
  Name VARCHAR(80) DEFAULT NULL,  
  Type char(1) DEFAULT NULL,  
  Value decimal(8,5) DEFAULT NULL,  
  TaxExempt char(1) DEFAULT NULL,  
  DateCreated DATETIME DEFAULT NULL,  
  DateModified DATETIME DEFAULT NULL,
```

```
PRIMARY KEY (ID)
) TYPE=MyISAM;
```

Unless there is a compelling reason not to, it is recommended that the **PriceGroup** table is added to the appropriate database even if price groups will not be used immediately. By keeping all database tables in their most current form there will be a reduction in possible future issues if price groups are eventually used.

- If price groups will be used, add a field with the following specifications to the **Customer** table:

```
PriceGroupID BIGINT(20) UNSIGNED DEFAULT NULL
```

The following statement can be used in the MySQL client to add the field:

```
ALTER TABLE Customer ADD COLUMN PriceGroupID BIGINT(20) UNSIGNED DEFAULT NULL
AFTER TempCustomerID;
```

Unless there is a compelling reason not to, it is recommended that these optional fields are added to the **Customer** table even if price groups will not be used immediately. By keeping all database tables in their most current form there will be a reduction in possible future issues if price groups are eventually used.

- If store administration users will be used, add a table named **AdminUser** to the appropriate database using the following statement in the MySQL client:

```
CREATE TABLE AdminUser (
  ID BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  Active char(1) DEFAULT NULL,
  Username VARCHAR(80) DEFAULT NULL,
  Password VARCHAR(80) DEFAULT NULL,
  Permissions TEXT,
  DateCreated DATETIME DEFAULT NULL,
  DateModified DATETIME DEFAULT NULL,
  PRIMARY KEY (ID)
) TYPE=MyISAM;
```

Unless there is a compelling reason not to, it is recommended that the **AdminUsers** table is added to the appropriate database even if store administration users will not be used immediately. By keeping all database tables in their most current form there will be a reduction in possible future issues if store administration users are eventually used.

Content Page Modifications

A reasonable effort has been made to outline the incremental changes made to aid those that have made modifications to any of the content pages. This list is by no means guaranteed to be exhaustive so if something was left out please make a modification request in the support area of the HostedStore Web site (www.hostedstore.com/support/).

- Add the following two content pages to the */hostedstore/storefront/html/* directory take advantage of savable/recoverable carts.
 - *cart_recoverform.lasso*
 - *cart_saveform.lasso*
- Perform a global search and replace for all occurrences of the following strings:
 - `href="/HostedStore.LassoApp` should be changed to `href="[$HS_PathToStore]/HostedStore.LassoApp.`
 - `href="HostedStore.LassoApp` should be changed to `href="[$HS_PathToStore]/HostedStore.LassoApp.`
 - `action="/HostedStore.LassoApp` should be changed to `action="[$HS_PathToStore]/HostedStore.LassoApp.`

- o `action="HostedStore.LassoApp` should be changed to
`action="[$HS_PathToStore]/HostedStore.LassoApp.`

Pre-2002.07.01 to 2002.07.01

No upgrade instructions. Extensive changes were made to all files and the database structure so a full install of provided files is necessary.

The Customer's Perspective

This section will discuss the areas of the store visible to the customer, as well as what it means to have specific features enabled. Details about enabling and disabling store features and modules can be found in the [Installation and Configuration](#) > [Store Configuration](#) section. Discussion about the role of individual files will also be included. It is necessary to understand the function of all HostedStore pages if enhancements to the existing content and display are to be made. Details about modifying HostedStore content pages can be found in the [Modifying HostedStore](#) section.

Sections

- [Storefront](#) - A tour of storefront features from the customer's perspective.
- [Account Administration](#) - Customer account configuration options.

Storefront

The storefront refers to the area of the store where customers browse the product catalog, add products to their shopping cart, and purchase the products.

- **Browsing Products** - The column along the left side provides hyperlinks to quickly search for products by category, subcategory, manufacturer, memberships, and sale priced items. These are static hyperlinks that must be manually updated by modifying the `/hostedstore/includes/navigation/browseproducts.lasso` content page. Details can be found in the [Modifying HostedStore > Content Pages > Navigation Pages > Browse Products](#) section.
- **Searching for Products** - Products can be found via two methods. One is via a quick search that searches for the product generically across all product description fields. The second involves a search form that allows the customer to tailor their search to specific fields in the **Product** table.
 - **Quick Search** - Products can be found by typing search criteria into the **Quick Search** box at the top of the page generated by the default template files. The **Product** table is searched for matches where the search criteria are contained in the name, category, subcategory, manufacturer, or description for the product.
 - **Advanced Search** - The search form can be reached by clicking the **Advanced Search** tab at the top of the page generated by the default store template files. Several fields in the **Product** table can be searched simultaneously.
- **Product Details** - The default product detail content page has multiple features that can be toggled on or off. Product zoom images can be used if a third image size is needed. Email a friend, product reviews, and product complements can be displayed.

The default product detail content page handles products of two types: single items that are purchased as-is and products that can have multiple required or optional attributes that must be chosen by the customer prior to adding the item to the cart.

- **Help Center** - The *Help Center* is provided as a central place where files that address potential customer questions can be placed. All of the files in the *Help Center* are open for modification. HostedStore ships with some of the pages containing text. Details about modifying the *Help Center* content pages can be found in the [Modifying HostedStore > Content Pages > Help Center](#) section.

Important: Review the pages in the *Help Center* carefully to ensure that the text reflects the store policies.

A customer service inquiry form is also provided that will send email to the customer service email addressed set as the `HS_CustomerServiceEmail` preference variable in *Store Configuration*.

Sections

- [Home Page](#)
- [Product Search](#)
- [Product Detail](#)
- [Shopping Cart](#)
- [Checkout](#)
- [Help Center](#)
- [Email a Friend](#)
- [Product Review](#)
- [Wish Lists](#)

Home Page

The default store home content page displays a list of featured products and product favorites. Like all HostedStore content pages, this page can be modified to display other information in addition to or instead of the default information.

Featured Products

The *Product Administration* area in *Store Administration* is used to flag products to be featured on the home page. Featured products can be arbitrarily assigned and need not correspond to something special like sale pricing or limited quantities available. The total number of featured products displayed can be easily modified in the *Store Configuration*.

Each product displays with a thumbnail image and the name of the product, both of which provide a link to the product detail content page where more information can be provided for the customer to make a purchase decision.

Featured Items

	One Cent Test Item Test item to keep transaction costs down.
Custom Build Your Computer	Custom Built Pentium IV Computer Custom build an Intel Pentium IV computer to your own specifications.
	3 Watt Booster Amplifier SKU AMP3WATT Compatible with Nextel phone models: i1000, i1000plus, i2000, i2000plus, i280, i30sx, i370, i390, i470, i500plus, i50sx, i550plus, i55sr, i600, i700plus, i80s, i85s, i90c, r300
	Permanent Hands Free Kit SKU SGHN0000801 Compatible with models: DM110, SP110, V111

Favorites

Favorites are automatically determined by HostedStore by finding the products that have sold the highest number of units historically. Product favorites can be self-reinforcing in nature if products are purchased often due to their placement on the home page. Essentially resulting in favorite products always remaining favorite products. The total number of favorites to display can be easily modified in the *Store Configuration*.

As with featured products, each favorite displays with a thumbnail image and the name of the product, both of which provide a link to the product detail page where more information can be provided for the customer to make a purchase decision.

Favorites



One Cent Test Item
Test item to keep transaction costs down.

Custom Build Your Computer



Custom Built Intel Celeron/PIII Computer
Custom build an Intel Celeron/PIII computer to your own specifications.



HostedStore 1 Year Membership
Get access to special pricing and members only items. Get two months for free by purchasing a one year membership.

If you want to try out the HostedStore membership first, you should consider purchasing a one month membership.



Targus Leather Deluxe Backpack
Made of a full grain leather exterior, the Targus Leather Deluxe Backpack is manufactured from the highest quality materials, features more than 15 pockets/compartments and carries a lifetime warranty from the manufacturer. Take your notebook computer (along with related accessories, books or other goodies) wherever you go without worrying about bumping, dinging or scratching your prized possession.



Lasso Web Data Engine 3.6.6
Blue World's Lasso Web Data Engine 3.6.6 is a leading cross-platform web application server software that will allow you to easily tie your web site's front-end to your database back-end in order to build powerful database-driven web sites. Lasso technology has been used to build a few dozen web sites worldwide the very web site you're visiting right now!

Product Search

Customers can search for products using three different methods. Any of the methods can be eliminated by removing it from the supplied store templates or by creating a new template that does not contain the feature - although a reason why a store manager would want to make it harder for a customer to find a product does not immediately come to mind, the option is available.

Browse Products

Customers can search for products by clicking preconfigured hyperlinks in the *Browse Products* menu on the lefthand side of the store templates. The hyperlinks can point to specific product categories, subcategories, or manufacturers and even products that are currently sale priced. The browse products menu is contained in an included HTML file that can be edited and formatted as needed. Details can be found in the [Modifying HostedStore](#) > [Content Pages](#) > [Navigation Pages](#) > [Browse Products](#) section.



Browse products menu

For more complex products, the hyperlinks in the *Browse Products* menu (or other hyperlinks in the site) can link to intermediate search content pages that help the customer further refine their search. HostedStore supports multiple search criteria generated from checkbox form elements or from comma delimited (any delimiter can be specified) text in a text or textarea form element. Furthermore, the search parameters can be set to ensure that all of the multiple criteria apply (AND search) or at least one of the multiple criteria apply (OR search). This flexibility allows for the creation of search forms that will support detailed and specific searches.

Manufacturer Search: AudioVox

Please select the accessory type(s) and/or phone model(s) that interest you and then click the **Search** button. Alternatively, click on the accessory type hyperlink to view that accessory for all phone models or click the phone model hyperlink to view all accessories for that model.

Type of Accessory

<input type="checkbox"/> Cases/pouches/holsters (20) <input type="checkbox"/> Cigarette lighter adapters (12) <input type="checkbox"/> Dash-mount holders (1) <input type="checkbox"/> Desk top chargers (17) <input type="checkbox"/> Headsets/earpieces/surveillance kit (9) <input type="checkbox"/> Lithium-ion batteries (18) <input type="checkbox"/> Miscellaneous (15)	<input type="checkbox"/> Nickel-metal-hydride batteries (4) <input type="checkbox"/> PDA adapter & cables (13) <input type="checkbox"/> Permanent install hands-free kits (7) <input type="checkbox"/> Portable hands-free kits (10) <input type="checkbox"/> Software (4) <input type="checkbox"/> Travel chargers (15) <input type="checkbox"/> Wireless data kits (1)
---	---

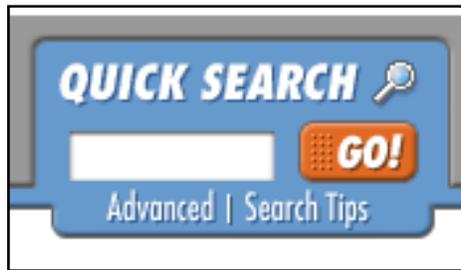
Phone Model

<input type="checkbox"/> CDM100	<input type="checkbox"/> CDM8150x	<input type="checkbox"/> CDM9200X	<input type="checkbox"/> GDX190	<input type="checkbox"/> PCX900
<input type="checkbox"/> CDM135	<input type="checkbox"/> CDM8200	<input type="checkbox"/> CMP3	<input type="checkbox"/> GDX708	<input type="checkbox"/> PCX1110xl
<input type="checkbox"/> CDM135XL	<input type="checkbox"/> CDM9100	<input type="checkbox"/> GDX110	<input type="checkbox"/> GP710	<input type="checkbox"/> PCX3600XL
<input type="checkbox"/> CDM8100	<input type="checkbox"/> CDM9150xd	<input type="checkbox"/> GDX120	<input type="checkbox"/> GSA190	<input type="checkbox"/> 9155GPX
<input type="checkbox"/> CDM8150	<input type="checkbox"/> CDM9155	<input type="checkbox"/> GDX170		<input type="checkbox"/> Maestro

Advanced search form with checkbox elements

Quick Search

The *Quick Search* form at the top of each default store template allows the customer to type in a search term or search terms without distinguishing exactly how the term is to be applied. By default the search term is matched against any products that contain the term in the product name, category, subcategory, manufacturer, or description (the **ProductName**, **Category**, **SubCategory**, **Manufacturer**, and **Description** fields in the **Product** table, respectively). Like just about everything else with HostedStore, that too can be modified.



Quick search form

Advanced Search

An *Advanced Search* form enables customers to search for products based on the fields placed in the form by the store manager. As with intermediate search content pages, fields can be searched using checkbox selections and delimited text. Experience has shown that an *Advanced Search* form is seldom used if the *Browse Products* menu provides enough product breakdown.

Advanced Product Search

Please enter criteria to search our catalog of over 2270 products.

Product Name

Category

SubCategory

Manufacturer

Description

Sale Priced

Price Range From \$ to \$

Advanced search form

Search Results

Results from any of the search methods are returned in a simple list format. If no products are found that meet the search criteria, then an appropriate message is displayed along with the *Advanced Search* form.

Each product displays with a thumbnail image and the name of the product, both of which are hyperlinks to the product detail content page where more information can be provided to help the customer make a purchase decision. A brief description and current product pricing and many more options are available to display as well.

Product Search Results

Your search found 53 selections. Click on the product image or name to view details.

<<Previous **1-20** **Next**>>

	<u>Product Name and Description</u>	<u>Price</u>
	Battery Charging Stand SKU DDC1 Compatible with models: 3320, 3360, 3390, 3395, 6360, 8260, 8290, 8390, 8890	\$33.44
	Clip On Hands Free Speaker SKU 98325 Compatible with models: V120c	\$54.99
	Clip On Hands Free Speaker & Belt Clip SKU SYN8610 Compatible with models: V60c	\$54.58

Search results

Product Detail

The product detail content page provides all of the details necessary for a customer to make an informed purchase decision. The amount of information displayed is dictated by how the product and store have been configured. As with all HostedStore content pages, the type, formatting, and location of the information can be modified.



R35 HTML webCheat Sheet

MSRP \$19.95
Our Price \$19.95

[ADD TO CART](#)

[EMAIL A FRIEND](#)

[ADD TO WISHLIST](#)

[REVIEW](#)

Product Reviews
No reviews at this time.

You might also like



R35 webColor Pad



HTML Web Magic (2nd Edition)



[click here : web communication design](#)

ZOOM

R35 HTML webCheat Sheet
Designed by Raymond Pirouz in 1999, the R35 HTML webCheat Sheet™ is a 24 X 18 inch poster that organizes and displays over 140 of the most popular, cross-browser compliant HTML 4 tags to help you cheat your way through HTML without having to memorize a line of code.

This indispensable reference chart also displays the most common HTML ASCII codes (you know, for specifying trademark symbols, curly quotes and non-breaking spaces) in an easy-to-use and attractive format.

Dimensions
18 inches high
24 inches wide

Product detail page

Product Details

A product image is displayed which when clicked will popup a new window that provides a larger image. Handy for products that may have details that need to be seen. A full description of the product is shown below the product. The full description does not necessarily need to correspond to the brief description that is shown on the search results content page. HTML can be used in both the brief and full product descriptions.

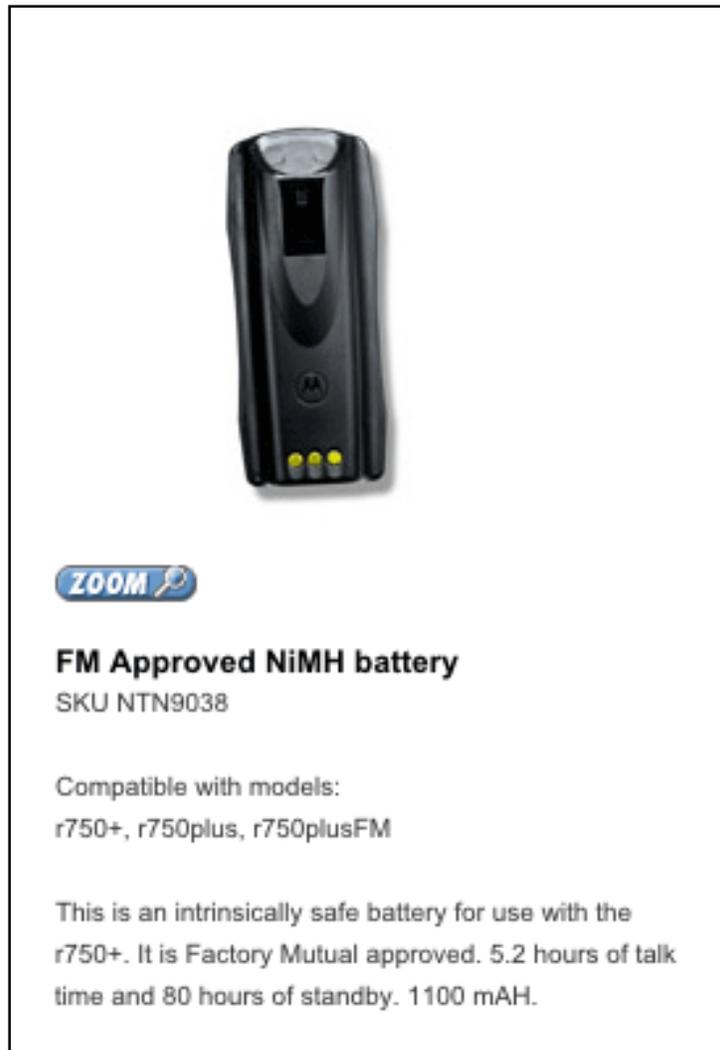


Image and description area of the product detail page

Product Pricing

Pricing details are displayed on the right side of the page. Depending upon the product and how the store is configured, the manufacturer's suggested retail price (MSRP), store price, sale price, and member price may all be shown.

MSRP \$1,180.00
Our Price \$744.95
Sale Price \$699.99
Members \$675.00

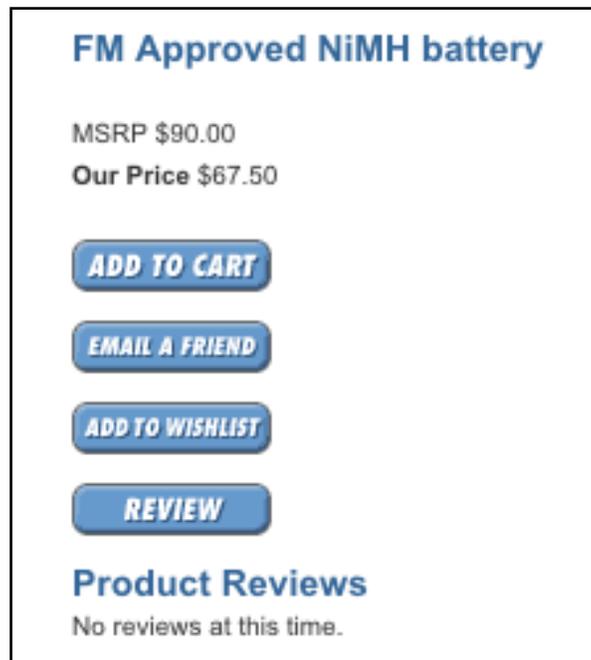
Pricing area of the product detail page

Button Actions

Below the pricing, four buttons provide the customer with choices about what to do with the product: **Add to Cart**, **Email a Friend**, **Add to Wish List**, and **Review**. If the product needs to be configured, then the Add to Cart and Add to Wishlist buttons will be displayed at the bottom of the page with a message noting that the product must be configured in place of the buttons that normally appear at the top of the page.

Details about what each of the four buttons do:

- The **Add to Cart** button will place the item in the cart or increment the current item count by one if the product is already in the cart. The shopping cart content page is displayed after adding an item to the cart.
- The **Email a Friend** button gives the customer an opportunity to enter a message to send to a friend along with a product description and hyperlink to the product detail content page.
- The **Add to Wish List** button will add the product to the customer's current wish list. Because the wish list has to be associated with the customer, the customer must be logged in to use this feature.
- The **Review** button gives the customer an opportunity to provide other customers with a commentary of their opinion and experience with the product. This feature can be optionally configured to only allow customers that are currently logged into the system to write a review. The product review is not necessarily associated with the customer - if the customer chooses not to enter their name, then the review will be posted anonymously.



Buttons area of the product detail page

Product Reviews

A summary of reviews for the product is displayed on the righthand side under the four buttons. If no reviews have been posted, then an appropriate message is displayed along with a link inviting the customer to post a review. When reviews have been posted, the number of reviews is displayed as well as the average rating of all the reviews based on a ten point scale.



Review summary area of the product detail page

Product Complements

If any complementary products have been associated with the product, each will be displayed with a

thumbnail image and product name. Up to six categories of products, each with an unlimited number of products, can be associated with each product. Clicking the image or product name will load its product detail content page. Complementary products can be used to upsell, cross sell, or accessorize a product.

You might also like



R35 webColor Pad



HTML Web Magic (2nd Edition)



click here : web communication design

Complements area of the product detail page

Configurable Products

Some products have attributes that need to be selected before the item can be added to the cart or wish list. Other products may have optional attributes like a warranty plan or gift wrapping. Each attribute may be used to change the price of the item by adding to or subtracting from the base price.

Individual attributes may need more explanation in order to help the customer make an informed choice. If an attribute has an associated image or HTML file, a hyperlink will appear next to it which will popup a new window with the image or HTML product description file when clicked.

Attributes can also be configured to alter the inventory of other products even if the other product is not offered as an item for individual sale. This feature is useful for items like custom built computers. The customer may choose their own hard drive, memory, etc and each of their choices will adjust the inventory of each of the individual items selected.

Text box attributes are also available. This type of attribute allows a customer to type in their choice for the attribute. Example uses for this feature are special instructions, monogrammed items, and greeting card messages for a gift wrapping option. Text box attributes can be configured to be optional or required fields.

Please make a selection for each option below.

Case

Processor

Cooling

Mainboard

System Memory

Hard Drive

Controller

CD/DVD-ROM

CD-Recordable

Floppy Drive

Video Card

Modem

Network Card

Sound Card

Speakers

Keyboard

Mouse

Monitor

Power

Operating System

Application

Printer

Digital Imaging

System Build

[ADD TO CART](#) [ADD TO WISHLIST](#)

Configuration area of the product detail page

Shopping Cart

The shopping cart is one of the most important parts of HostedStore because all the functionality of the store interacts with it either directly or in the background. Because of its importance, many features have been built into the customer interface to aid in the shopping process.

Shopping Cart

To change the quantity of an item in your shopping cart, enter the new quantity in the item's Quantity field and click the Recalculate button below. To remove an item from your shopping cart, enter a 0 in the item's Quantity field and click the Recalculate button below. You may modify the quantities of multiple items at the same time.

The names of items with attributes will appear as a hyperlink. Click the link to modify the attributes for the item.

Product Name	Quantity	Price	Total
Travel Charger	<input type="text" value="1"/>	\$19.24	\$19.24
FM Approved NiMH battery	<input type="text" value="1"/>	\$67.50	\$67.50

Total Without Tax			Total with Tax		
	Quantity	Total	AZ, WA Residents Only		
			State	Tax	Total
SubTotal	2	\$86.74	AZ	\$7.46	\$99.20
Shipping		\$5.00	WA	\$7.55	\$99.29
Total		\$91.74			

[Recalculate](#) [Continue Shopping](#) [Checkout →→→](#)

Shopping cart

Static Shopping Cart

All of the store templates supplied with HostedStore feature a static shopping cart on the left side of the page that summarizes the number of items and the subtotal price of the items in the cart. The static cart serves as a reminder of the cart status without having to view the entire cart.



Static shopping cart

Ordered Product Detail

The shopping cart detail content page displays the ordered product and any of the customization details if applicable. If the product has been added from the customer's wish list or another customer's wish list, there will be a small notation that designates the product as a wish list item.

If the product is configurable, then the current product configuration is detailed below the product name. The product name appears as a hyperlink which, when clicked, returns the product detail content page with the current configuration for all attributes selected. The customer may then make any necessary modifications.

As is customary with shopping carts, the customer is able to modify the quantity of an item or multiple items simultaneously in their cart. Updating quantities returns the shopping cart detail content page with new line item subtotals and recalculated shipping cost, tax, and order total.

Product Name	Quantity	Price	Total
Custom Built AMD Athlon Computer Case SuperMicro SC830W Server-Tower w/ Dual 300W Processor AMD T-Bird 1.4GHz 384k Cache 266MHz FSB Cooling AMD Duron/T-Bird Ball Bearing Fan Mainboard ASUS A7M266 Socket A 266MHz DDR System Memory 256MB 184-pin PC2100 DDR SDRAM Hard Drive Seagate 73.4GB Cheetah U160 SCSI Controller Adaptec 3400S U160 RAID Controller CD/DVD-ROM Toshiba 16x/48x EIDE DVD-ROM CD-Recordable Toshiba EIDE 4x/4x/4x/24x CD-RW/DVD-ROM Floppy Drive Teac 1.44MB Floppy Drive Video Card VisionTek GeForce3 64MB DDR AGP Modem U.S. Robotics 56k v.90 Internal ISA OEM Network Card 3COM 3C905C-TXNM 10/ 100Mbps PCI OEM Sound Card Creative SoundBlaster Live! Platinum 5.1 Speakers Monsoon MM-1000 Flat Panel Speaker System Keyboard Microsoft Natural Pro Elite USB OEM Mouse Microsoft IntelliEye PS/2 - USB Monitor ViewSonic VP181 18.1 Inch Digital LCD Power APC Back-UPS Pro 280 w/ PowerChute Bundle Operating System Microsoft Windows 2000 Server OEM Application Microsoft Office XP Professional Printer Hewlett Packard Desk Jet 930C Digital Imaging Canon PowerShot S110 System Build Build Fee	1	\$8,457.76	\$8,457.76
Lasso Web Data Engine 3.6.6 * added from a wishlist. Operating System Macintosh Data Source Module 4D	1	\$814.95	\$814.95

Ordered product detail

Shipping/Tax Detail

Many surveys of online shoppers have revealed that one of the top reasons for abandoned carts and unfinished orders is due to the surprise the customer experiences when they see the shipping costs applied to their order during the checkout process. HostedStore calculates the shipping for all of the cart items each time the shopping cart detail content page is viewed. This feature was implemented to help reduce the sticker shock that leads to abandoned carts.

The shopping cart detail content page also displays two separate order total columns. One details the subtotal, shipping, and order total for shoppers who reside in a non-taxable location. The second details the tax and total cost for each location that has been configured for taxation. This feature also helps to eliminate "checkout surprise." Some shopping carts force the user to enter their location in order to see the proper total. HostedStore's approach eliminates that inconvenience.

Details about about configuring taxable jurisdictions can be found in the [Installation and Configuration > Store Configuration](#) section.

Total Without Tax			Total with Tax		
			MA, WA Residents Only		
	<u>Quantity</u>	<u>Total</u>	<u>State</u>	<u>Tax</u>	<u>Total</u>
SubTotal	2	\$9,267.75	MA	\$471.39	\$9,899.14
Shipping		\$160.00	WA	\$763.65	\$10,191.40
Total		\$9,427.75			

Shipping/tax detail

Email Quote

To help facilitate the decision making process, the customer can email their current cart contents to their own email address or possibly to the email address of the decision maker in their company. The store manager can set how many days the quote is valid and that value is reflected in the emailed quote. Quotes include a detailed breakdown of ordered products including any customization. Details about setting the number of days a quote is valid can be found in the [Installation and Configuration > Store Configuration](#) section.

If you would like to receive a quote, please enter your email address below and click the Email Quote button.

Email Address

Email cart quote form

Savable/Recoverable Carts

Multiple shopping carts can be saved with a unique name. This feature allows for the recovery of shopping sessions at a later date. It is also a tremendous aid to customers that may shop from one computer, but would like to purchase from another computer at a later date.

Save Cart

Your cart can be saved with a name you choose and recovered at a later date. This will allow you to save an order that you wish to pay for later, possibly from a different computer. If you are saving your cart because it is a regular order, you should create a shopping list in the customer administration area instead. Shopping lists do not expire, but saved carts might.

Please choose a name for the cart and enter your email address below. These items will be needed later when you recover the cart.

Cart Name

Email Address

Save cart form

Recover Cart

Please enter the cart name and your email address to recover your cart.

Cart Name

Email Address

Empty cart before adding saved cart items.

Delete saved cart after adding items to the cart.

Recover cart form

Checkout

The checkout process is initiated by clicking the **Checkout** button on the default shopping cart display content page or any other button or hyperlink to the checkout script. If there are not any items in the shopping cart, then the checkout process will be stopped and an appropriate message will be displayed.



Checkout button on the shopping cart display page

Customer Account Creation and Login

In order to checkout, the customer must create an account if they are a new customer or login with their existing account information. An email address and password are used to begin the account creation process or to login.

Customer Login

If you are a new customer please enter your email address and choose a password to create an account. If you are a returning customer please login below and your account information will be retrieved.

I am a new customer

Email Address

Verify Email Address

Choose a Password

Verify Password

I am a returning customer

Email Address

Password

I forgot my password.

Continue

Account creation and login

Customer Billing Address

If a customer is creating a new account, then they will need to provide their billing address information. The address will be saved as an entry named *Billing* in the customer's address book if the address book feature has been enabled in *Store Configuration*. Details about enabling the address book feature can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

A checkbox is provided on the default account creation form content page that, when checked, will duplicate the billing address information into the shipping address information fields on the next form.

If the newsletter feature has been enabled in *Store Configuration*, then a checkbox will be provided on the default account creation form content page that, when checked, will subscribe the customer to the store newsletter.

Billing Information

Please complete the billing information to be associated with your account below. If you would like to skip a step and use this address for shipping, please check the checkbox below.

Important: This information should match the billing address for your credit card account or there may be problems processing your order.

* denotes an empty required field.

* **First Name**

* **Last Name**

Company

* **Address**

* **City**

* **State/Province**

* **Postal Code**

* **Country**

Email Address sales@hostedstore.com

* **Phone**

Fax

Use for shipping information.

YES! Please subscribe me to your newsletter.

Create Account

Billing information when creating an account

Order Shipping Address and Payment Information

If the address book feature has been enabled in *Store Configuration*, a select list will be provided for the customer to choose one of their address book entries for the shipping address. Every customer will have at least their *Billing* entry in the list.

A shipping address can also be hand entered and, if the address book feature is enabled in *Store Configuration*, saved as an address book entry.

Shipping Address

Use Address Book

* denotes an empty required field.

Save Address As

* **First Name**

* **Last Name**

Company

* **Address**

* **City**

* **State/Province**

* **Postal Code**

* **Country**

Email Address

Phone

Fax

Shipping information

A space is provided for customers to enter any comments, specials instructions, etc about their order.

Comments

If you have any comments or special requests that should be associated with your order, please provide them below.

Comments

Order comments

If promotions have been enabled in *Store Configuration*, the customer is provided with a box to enter the promotion/coupon code to use for their order. The promotion code will be verified to determine if it is valid for the order. Details about promotions can be found in the [Store Administration](#) > [Promotion Administration](#) section.

Order Discount

If you have a promotion or coupon code you would like to use for this order, please enter it below.

Promotion/Coupon Code

Promotion/coupon code

The payment methods that are available according to preferences set in *Store Configuration* are displayed to the customer. Details about enabling payment methods can be found in the [Installation and Configuration](#) > [Store Configuration](#) section. Clicking the **Continue to Order Review** button will proceed to the final page in the checkout process. The shipping address and payment information form is lengthy and is the one place in the checkout process where the customer is most likely to encounter errors due to incomplete information. If an error is encountered, then an appropriate message will be displayed that prompts the user to enter a valid value.

Payment Method

Check by Mail/Phone

Check

* **Check Number**

* **Transit Routing Number**

Note: This is the nine digit number located between the |: symbols at the bottom of the check.

* **Account Number**

Note: This is the remaining group of numbers at the bottom of the check.

* **Signature**

Note: Type your full name as you would sign a check.

PayPal

Purchase Order

* **PO Number**

Credit Card

* **Credit Card Type**

* **Credit Card Number**

* **Expiration Date**

Payment information

Order Review

After entering payment information, the customer is presenting with a full summary of the order including billing address, shipping address, payment information, ordered item summary, and order totals on the default order review content page. Hyperlinks are provided in several locations to modify incorrect information. Clicking the **Purchase** button will process the order according to the payment gateway (if applicable) set in *Store Configuration* for the chosen payment method. Details about payment gateways and payment methods can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

Order Review

Please verify the shipping and order information below. If the information is correct, please click the "Purchase" button below. Click on the appropriate **Modify** link to modify payment or shipping information. Click a product name to modify its attributes.

Payment: *Method* MasterCard
Modify *Account* xxxxxxxxxxxxxx0015
Expiration Date 03/2003

Ship to: Jane Doe
Modify HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Cigarette Lighter Adapter	1	\$12.07	\$12.07
Portable Hands Free Kit * purchased from a wishlist.	1	\$59.99	\$59.99

	<u>Quantity</u>	<u>Total</u>
SubTotal	2	\$72.06
Shipping		\$5.00
Tax		\$6.24
Discount		<u>(\$0.00)</u>
Total		\$83.30

Purchase

Order review

Order Confirmation and Receipt

If all goes well, an approval message and order receipt will be displayed to the customer on the default order result content page. A hyperlink to a printer friendly version of the receipt without the store navigation is also provided. If the payment processing fails, then an appropriate message is displayed to the customer.

Order Received

Your order has been received.

A summary of your order appears below. An emailed summary of the order should arrive in your inbox soon. Please print a copy of this screen for your records. [Click here](#) for a printer friendly receipt.

Bill to:

Jane Doe
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US

Ship to:

Jane Doe
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US

Payment Information

Method MasterCard
Account xxxxxxxxxxxxxx0015
Expiration Date 03/2003

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Cigarette Lighter Adapter	1	\$12.07	\$12.07
Portable Hands Free Kit * purchased from a wishlist.	1	\$59.99	\$59.99

	<u>Quantity</u>	<u>Total</u>
SubTotal	2	\$72.06
Shipping		\$5.00
Tax		<u>\$6.24</u>
Total		\$83.30

Order confirmation and receipt

Help Center

A customer *Help Center* is provided to act as a central repository for all store policies and instructions on how to perform different tasks in the store. All of the files are simple HTML files which can be modified or eliminated depending upon the needs of the store manager. Topics that are specific to how areas or functions of the store work are supplied complete with draft descriptions, whereas pages specific to store policies are left blank. All of the pages should be carefully reviewed and modified by the store manager to reflect the policies and business model of the store. Details about modifying the *Help Center* content pages can be found in the [Modifying HostedStore](#) > [Content Pages](#) > [Help Center](#) section.

Help Center

If you need help with anything in our store, please refer to the items below. If you don't find the answers you are looking for here then please use the contact information at the bottom of this page to contact us.

New Customers	Ordering
First Time Customers	Placing an Order
Frequently Asked Questions	Coupons Promotions Rebates
About This Store	Payment Options
Browser Support	Shipping and Delivery
	Returning an Item
Policies	Canceling an Order
Privacy Policy	Ordering Problems
Security Policy	
Return Policy	Your Account
Sales Tax Policy	Managing Your Account
Pricing and Availability Policy	Status of Your Order
Errors and Omissions Policy	Tracking Your Order
Review Guidelines	
Other	Search Tips
Send us Feedback	Searching
Report Site Problems	
Still need more help?	
Please fill out the Customer Service Inquiry Form .	
Still need assistance? Click here for fax, callback or email information.	

Help center menu

Customer Service Inquiries

A customer service inquiry form is also provided that will enable customers to ask pre or post sales questions. The form fields are emailed to an address that has been specified in the *Store Configuration*. Details about modifying the customer service email address can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

Customer Service Inquiry

* denotes an empty required field.

* **Name**

* **Email Address**

* **Subject**

* **Details**

[Submit Inquiry](#)

Customer service inquiry form

Help Center Access

All of the store templates included with HostedStore provide a conspicuously placed link to the *Help Center* on each page. According to surveys of online shoppers, one of the common complaints of customers or potential customers is the inability to locate the information they need when they need it. With that in mind, HostedStore provides a framework for the store manager to create a well thought out and detailed help center to eliminate customer frustration and the potential loss of sales.



Help center store template link

Email a Friend

The **Email a Friend** button on the product detail page gives the customer an opportunity to enter a message to send to a friend along with a product description and hyperlink to the product detail content page. This feature must be enabled in *Store Configuration* for the button to appear on the default product detail content page.



Email a friend button on the product detail page

Email a Friend Form

The default email a friend form content page allows the customer to enter their email address, the recipient's email address, and a personal message. The default email format file sets the body of the email message to contain the provided personal message along with the product name (**ProductName** field from the **Product** table), a brief description (**BriefDescription** field from the **Product** table), and a hyperlink to the product detail content page. Clicking the **Send Email** button will send the email to the provided recipient address if all of the required fields have been completed and the recipient email address is a valid email address.

The bottom of the form displays the thumbnail image for the product, the product name (**ProductName** field from the **Product** table), and a brief description (**BriefDescription** field from the **Product** table). The product name is displayed as a hyperlink to the product detail content page.

Email a Friend

If you want to tell a friend or colleague about the **Dual Port Docking Station** product then fill in the information below. This email address is not stored in our databases and will never be sold or used for promotional purposes. The product information and a link to the product detail page will automatically be placed in the email for you.

* denotes an empty required field.

* **Your Email Address**

* **Recipient's Email Address**

Send me a copy of the email.

Personal Message

I saw this item for sale online at HostedStore Development Store and thought you might be interested:

Product information will be included here in your email...



Dual Port Docking Station

SKU SGCD0015801

Compatible with models:

LG13000W, TP3000

[Send Email](#)

Email a friend form

Product Review

The **Review** button on the product detail page gives the customer an opportunity to write a review about a product for the benefit of other customers. This feature must be enabled in *Store Configuration* for the button to appear on the default product detail content page. Further product review related restrictions can also be set in *Store Configuration*. Details can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.



Review button on the product detail page

Post a Review

The default review form content page displays the thumbnail image for the product, the product name (**ProductName** field from the **Product** table), and a brief description (**BriefDescription** field from the **Product** table). The review form allows the customer to provide their name if they would like (otherwise the review is attributed to "Anonymous"), a numerical rating on a one to ten scale, and a review. Clicking the **Post Review** button will add the review to the database if all of the required fields have been completed.

Post Review



Dual Port Docking Station

SKU SGCD0015801

Compatible with models:
LGI3000W, TP3000

Please complete the product review form below.

* denotes an empty required field.

Your Name

* Product Rating

(10 is better than 1)

* Review

Post Review

Product review form

Review Summary

The default product detail page will provide a summary of any reviews for the product including the number of reviews and the average rating. Only those reviews that have a **Publish** field value of **Y** in the **Review** table will be listed. A hyperlink to the review list content page is provided for those customers that would like to read the review(s). If there are not any reviews, then a suitable message will be displayed.

Product Reviews

1 customer gave this product a rating of 9.0 out of 10.

[Read Review](#)

Review summary on the product detail page

Review List

The default review list content page displays the thumbnail image for the product, the product name (**ProductName** field from the **Product** table), and a brief description (**BriefDescription** field from the **Product** table). A list of reviews is also provided with a summary of each review including the date posted, author, rating, the number of times the review has been read, and the number of times the review has been marked as helpful by someone who read the review (**DateCreated**, **Author**, **Rating**, **TimesRead**, and **TimesHelpful** fields from the **Review** table, respectively). The date posted is displayed as a hyperlink to the review detail content page.

List Reviews



Dual Port Docking Station
SKU SGCD0015801

Compatible with models:
LGI3000W, TP3000

ADD TO CART

The review for the **Dual Port Docking Station** product is listed below.
Please select it to read it in detail.

<u>Date Posted</u>	<u>Author</u>	<u>Rating</u>	<u>Read</u>	<u>Helpful</u>
11/10/2002	Anonymous	9	2	1

Product review list

Review Detail

The default review detail content page displays the thumbnail image for the product, the product name (**ProductName** field from the **Product** table), and a brief description (**BriefDescription** field from the **Product** table). A summary of review activity is provided to inform the reader of how many times the review has been read and how many times it has been marked as helpful (**TimesRead** and **TimesHelpful** fields from the **Review** table, respectively) along with the review details including the date posted, author, product rating, and the actual review (**DateCreated**, **Author**, **Rating**, and **ReviewText** fields from the **Review** table, respectively).

A form is provided to allow the reader to mark the review as helpful or not helpful after reading the review.

The bottom of the page displays a list of all reviews the product with a summary of each review including the date posted, author, rating, the number of times the review has been read, and the number of times the review has been marked as helpful by someone who read the review (**DateCreated**, **Author**, **Rating**, **TimesRead**, and **TimesHelpful** fields from the **Review** table, respectively). The date posted is displayed as a hyperlink to the review detail content page.

Read Review



Dual Port Docking Station

SKU SGCD0015801

Compatible with models:
LG13000W, TP3000

[ADD TO CART](#)

Including you, the following review has been read 2 times.
Customers have rated this review as helpful 1 time.

Date Posted 11/10/2002, 9:32 PM

Author Anonymous

Product Rating 9 out of 10

Review An outstanding product. We use these organization wide to charge two phones at once.

Did you find this review helpful? Yes No

The review for the **Dual Port Docking Station** product is listed below.
Please select it to read it in detail.

<u>Date Posted</u>	<u>Author</u>	<u>Rating</u>	<u>Read</u>	<u>Helpful</u>
11/10/2002	Anonymous	9	2	1

Product review detail

Wish Lists

If the wish list feature has been enabled in *Store Configuration*, then customers will be able to search for another customer's wish list if the wish list has been set for public searching in *Customer Account Administration*. The **FirstName**, **LastName**, **Email**, **City**, **State**, and **Country** fields in the **Customer** table can all be used as search criteria. Customers without a publicly searchable wish list will have to email a link to their wish list detail in *Customer Account Administration*. Details can be found in the [The Customer's Perspective](#) > [Account Administration](#) > [Wish List](#) section.

Wish List Search

Please enter as many search criteria as you would like to find the wish list.

First Name

Last Name

Email Address

City

State

Country

Wish list search form

Wish List Search Results

The **FirstName**, **LastName**, **City**, **State**, and **Country** fields from the **Customer** table are listed for each wish list that meets the search results. Note that the customer email address can be used for searching, but it is not displayed in the search results to prevent publishing the email address of other individuals. The name of the customer is displayed as a hyperlink to the wish list detail content page.

Wish List Search Results

Your search found 1 wish list.

Click on the name of the owner to view the items in the wish list.

<u>Name</u>	<u>City</u>	<u>State</u>	<u>Country</u>
Doe, Jane	Bellevue	WA	US

Wish list search results

Wish List Detail

The wish list detail content page lists all products currently in the wish list (items that have been purchased from a wish list will not be displayed in order to eliminate duplicate purchases if a wish list is used for a gift registry). The date the product was added to the wish list, the product name, and all chosen attributes (if any) of the product are detailed. The product name is displayed as a hyperlink to the product detail content page.

Clicking the **Add to Cart** button will add the item to the shopping cart. The pricing for the item and item availability are determined before adding the item to the shopping cart. If the item is no longer available (discontinued, out of stock, etc) or if item attributes choices are no longer available, then the product will not be added to the shopping cart and an appropriate message will be displayed.

Wish List

These are the items currently in the wish list.

Clicking on the product name will show product details.

<u>Date Added</u>	<u>Product Name</u>	
-------------------	---------------------	--

11/10/2002

Custom Built AMD Athlon Computer

Case SuperMicro SC830W Server-Tower w/Dual 300W
Processor AMD T-Bird 1.4GHz 384k Cache 266MHz FSB
Cooling AMD Duron/T-Bird Ball Bearing Fan
Mainboard ASUS A7M266 Socket A 266MHz DDR
System Memory 256MB 184-pin PC2100 DDR SDRAM
Hard Drive Seagate 73.4GB Cheetah U160 SCSI
Controller Adaptec 3400S U160 RAID Controller
CD/DVD-ROM Toshiba 16x/48x EIDE DVD-ROM
CD-Recordable Toshiba EIDE 4x/4x/4x/24x CD-RW/
DVD-ROM
Floppy Drive Teac 1.44MB Floppy Drive
Video Card VisionTek GeForce3 64MB DDR AGP
Modem U.S. Robotics 56k v.90 Internal ISA OEM
Network Card 3COM 3C905C-TXNM 10/100Mbps PCI OEM
Sound Card Creative SoundBlaster Live! Platinum 5.1
Speakers Monsoon MM-1000 Flat Panel Speaker System
Keyboard Microsoft Natural Pro Elite USB OEM
Mouse Microsoft IntelliEye PS/2 - USB
Monitor ViewSonic VP181 18.1 Inch Digital LCD
Power APC Back-UPS Pro 280 w/ PowerChute Bundle
Operating System Microsoft Windows 2000 Server OEM
Application None
Printer None
Digital Imaging None
System Build Build Fee

ADD TO CART

11/10/2002

Portable Hands Free Kit

ADD TO CART

11/10/2002

Dual Port Docking Station

ADD TO CART

Wish list detail

Shopping Cart Detail

Every item that has been added from a wish list will be noted on the shopping cart display content page. Once the customer checks out and the order has been approved, then the item will be removed from the wish list and placed in a list of purchased wish list items. Details about the distinction between wish list items and purchased wish list items can be found in the [The Customer's Perspective > Account Administration > Wish List](#) section.

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Portable Hands Free Kit * added from a wishlist.	1	\$59.99	\$59.99

Shopping cart with wish list item noted

Account Administration

The *Customer Account Administration* area provides the customer with tools to administer their account. The features available to the customer are dependent upon which options were selected in *Store Configuration*. All of the possible features are described below. Details about enabling *Customer Account Administration* related features can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

- **Modify Password** - Allows the customer to change the password used to access their account.
- **Newsletter** - Allows the customer to subscribe to or unsubscribe from the store newsletter.
- **Billing Address** - Allows the customer to modify their account billing address.
- **Address Book** - Allows the customer to add, modify, or remove address book entries that are made available as shipping addresses during checkout.
- **View Orders** - Allows the customer to view past orders, obtain tracking numbers, and cancel pending orders.
- **Wish List** - Allows the customer to view items currently in their wish list and items that were in the wish list but have been purchased. The customer can also add wish list items to their cart or remove items from the wish list.
- **Shopping Lists** - Allows the customer to create, modify, or remove shopping lists. Shopping lists are created from past orders and provide a convenient way for the customer to purchase regularly purchased items without having to go through the entire shopping process.
- **Purchased Files** - Allows the customer the customer to download purchased files. A history of the number of downloads allowed or the time limit on downloads is provided for each purchased file.
- **Logout** - Allows the customer to logout from their account administration and shopping session.

Account Administration

Please select an item from the options listed below.

Modify Password	Modify password.
Newsletter	Subscribe/Unsubscribe to the newsletter.
Billing Information	Modify billing information.
View Orders	Check status of current orders. Obtain tracking numbers for shipped orders. View past order history. View details of current and past orders. Create shopping lists from orders.
Wish List	Modify public search status. Email your wish list. View items in wish list. View items purchased from wish list.
Shopping Lists	Purchase items using a Shopping list. View items in Shopping Lists. Modify Shopping Lists.
Purchased Files	View files purchased. Download purchased files.
Logout	

Customer account administration menu

Sections

- [Modify Password](#) - Change the password used to access account administration.
- [Newsletter](#) - Subscribe/unsubscribe to/from the store newsletter.
- [Address Book](#) - Add, edit, and delete address book entries.
- [View Orders](#) - View order history and create shopping lists.
- [Wish List](#) - View and delete wish list items. Purchase wish list items and view purchased wish list items.
- [Shopping Lists](#) - Edit and delete shopping lists.
- [Purchased Files](#) - View and download purchased files.
- [Logout](#) - Logout from account administration and shopping session.

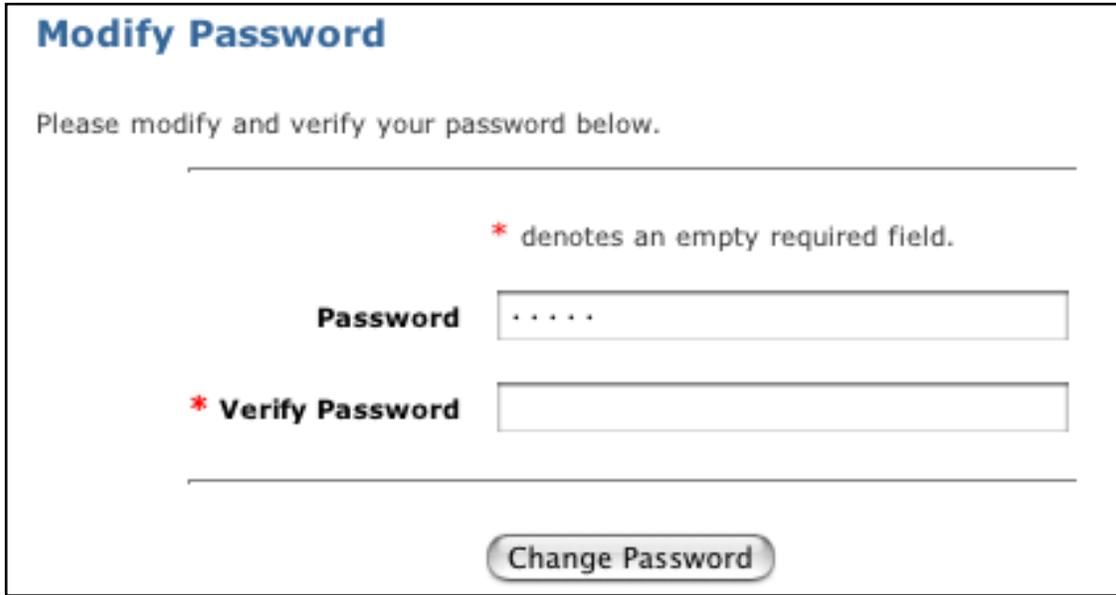
Modify Password

A customer can change the password used to access *Customer Account Administration* session by clicking the **Modify Password** hyperlink on the *Customer Account Administration* menu.



Modify password hyperlink on the customer account administration menu

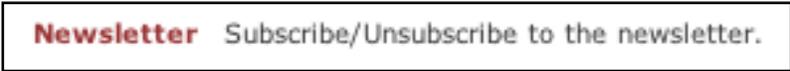
A password modification form will be displayed for the customer to provide the new password and confirm the password change by correctly typing the new password a second time. Clicking the **Change Password** button will apply the change.

A screenshot of a web form titled "Modify Password" in blue. Below the title is the instruction "Please modify and verify your password below." followed by a horizontal line. A red asterisk and the text "* denotes an empty required field." are positioned above the input fields. There are two input fields: the first is labeled "Password" and contains five dots; the second is labeled "* Verify Password" and is empty. A horizontal line is below the second field. At the bottom center is a rounded button labeled "Change Password".

Password modification form

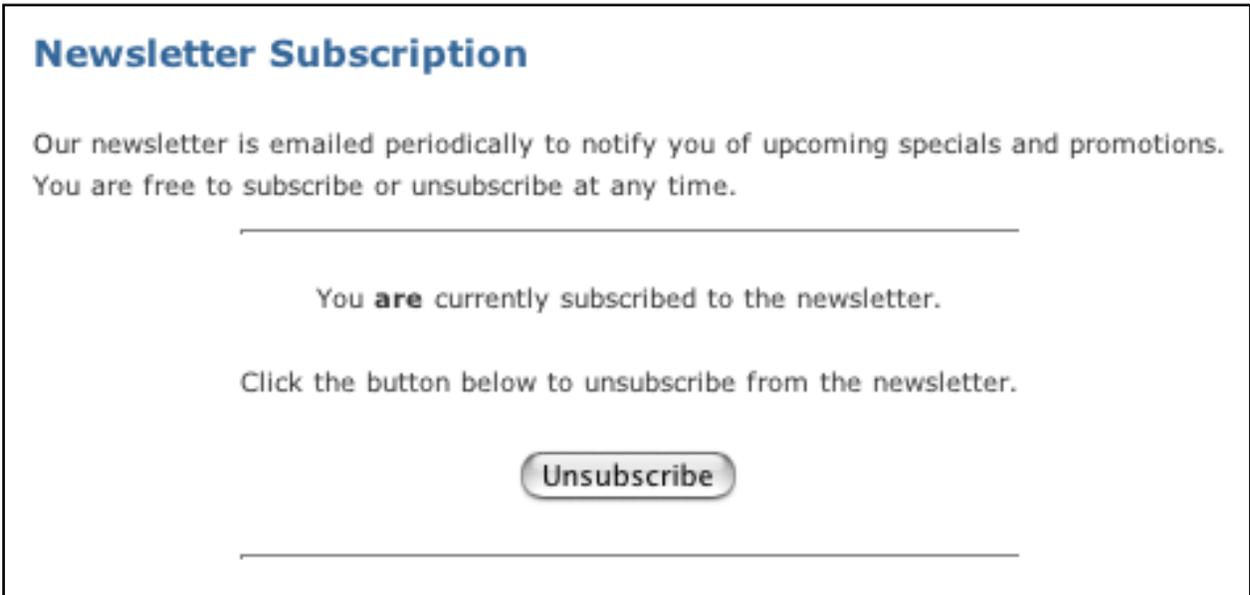
Newsletter

If the store has newsletter functionality enabled, then the customer can easily change their subscription status by clicking the **Newsletter** hyperlink on the *Customer Account Administration* menu.



Newsletter hyperlink on the customer account administration menu

A subscription modification form will be displayed showing the current subscription status along with a button that will toggle their subscription status. The appropriate subscribe or unsubscribe message will be sent to the list server if the appropriate preference variables have been configured properly in *Store Configuration*. Details about newsletter related preference variables can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

A screenshot of a web form titled "Newsletter Subscription" in bold blue text. Below the title, there is a paragraph: "Our newsletter is emailed periodically to notify you of upcoming specials and promotions. You are free to subscribe or unsubscribe at any time." This is followed by a horizontal line. Below the line, the text reads: "You **are** currently subscribed to the newsletter." Below this, it says: "Click the button below to unsubscribe from the newsletter." At the bottom of the form is a rounded rectangular button with the text "Unsubscribe" in bold black font. Another horizontal line is at the very bottom of the form area.

Newsletter subscription modification form

Billing Address

If address books have not been enabled in *Store Configuration*, then customer will have an opportunity to modify the billing address associated with their account. If address books are used in the future, then the customer's billing address will be an entry named **Billing** in the address book.

Edit the billing address by clicking the *Billing Address* hyperlink on the *Customer Account Administration* menu.

Billing Information Modify billing information.

Billing address hyperlink on the customer account administration menu

Edit Billing Address

Clicking the *Billing Address* hyperlink on the *Customer Account Administration* menu will load the customer's billing address into the edit form. Once all modifications are made, clicking the **Update** button will update the address in the table. A customer's billing address cannot be deleted because it is also used for the customer's billing information for all past orders.

Please modify the information below.

* denotes an empty required field.

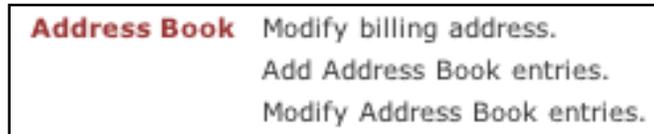
First Name	<input type="text" value="Jane"/>
Last Name	<input type="text" value="Doe"/>
Company	<input type="text" value="HostedStore Corporation"/>
Address	<input type="text" value="PO Box 6743"/> <input type="text"/>
City	<input type="text" value="Bellevue"/>
State	<input type="text" value="Washington"/>
Postal Code	<input type="text" value="98008-0743"/>
Country	<input type="text" value="US"/>
Email	<input type="text" value="sales@hostedstore.com"/>
Phone	<input type="text" value="425.644.7354"/>
Fax	<input type="text" value="425.644.7354"/>

Billing address edit form

Address Book

Customers can save multiple shipping addresses in their address book. All address book entries are available to the customer during the checkout process so that frequently used shipping addresses do not need to be retyped with each order. Every customer has an address book entry named **Billing** that contains their billing address. Address books are a feature that must be enabled in *Store Configuration*. More details about enabling address books can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

If the store has address book functionality enabled, then the customer can easily view and modify their address book entries by clicking the **Address Book** hyperlink on the *Customer Account Administration* menu.



Address book hyperlink on the customer account administration menu

Address Book List

Each address book entry is displayed along with the address assigned to the entry. The entry name appears as a hyperlink to the edit form. A hyperlink is also provided to create a new entry.

Address Book

Click an Entry Name to modify or delete.

Click here to add an entry to your Address Book.

<u>Entry Name</u>	<u>Address Information</u>
Billing	Jane Doe HostedStore Corporation PO Box 6743 Bellevue, WA 98008-0743 US Phone: 425.644.7354 sales@hostedstore.com
Grandma	Grandma Doe 123 Any Street Any Town, AL 12345 US

Address book list

Add an Entry

The add entry form contains fields for shipping address information. The customer must also assign a unique name to the entry. The customer billing address is automatically stored as an entry named **Billing** so that name cannot be used for any customer created entries. Once all required fields are completed, clicking the **Add** button will add the entry to the address book.

Add Address Book Entry

Please modify the information below.

* denotes an empty required field.

* **Entry Name**

* **First Name**

* **Last Name**

Company

* **Address**

* **City**

* **State**

* **Postal Code**

* **Country**

* **Email**

Phone

Fax

Address book entry add form

Edit an Entry

The edit entry form is loaded with the appropriate address for an entry after the entry name hyperlink is clicked in the address book listing. The edit entry form is very similar to the add entry form. Once all modifications are made, clicking the **Update** button will update the entry in the table. Clicking the **Delete** button will remove the entry from the address book. The **Billing** entry cannot be deleted because it is also used for the customer's billing information.

Modify Address Book Entry

Please modify the information below.

* denotes an empty required field.

Entry Name	<input type="text" value="Grandma"/>
First Name	<input type="text" value="Grandma"/>
Last Name	<input type="text" value="Doe"/>
Company	<input type="text"/>
Address	<input type="text" value="123 Any Street"/> <input type="text"/>
City	<input type="text" value="Any Town"/>
State	<input type="text" value="Alabama"/>
Postal Code	<input type="text" value="12345"/>
Country	<input type="text" value="United States of America"/>
* Email	<input type="text"/>
Phone	<input type="text"/>
Fax	<input type="text"/>

Address book entry edit form

View Orders

All of a customer's past orders can be viewed by clicking the **View Orders** hyperlink on the *Customer Account Administration* menu.

View Orders Check status of current orders.
Obtain tracking numbers for shipped orders.
View past order history.
View details of current and past orders.
Create shopping lists from orders.

View orders hyperlink on the customer account administration menu

Each order is listed in reverse chronological order with the date ordered, number of items in the order, the order total, and the status of the order detailed. Clicking the date ordered hyperlink will display the order detail content page.

View Orders

Click on the order date to view the details for that order.

<u>Date Ordered</u>	<u>Quantity</u>	<u>Total</u>	<u>Status</u>
10/27/2002	1	\$45.40	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending
10/27/2002	12	\$138.48	Pending

View orders

Order Detail

The order detail content page displays information about an order that may be of value to the customer. Each area of the order detail content page is explained below:

- **Order Status** displays the current status of the order in the sales process.
- **Order Dates** displays each of the dates a specific order action or status was reached. The customer will be able to see the date that the order was created, processed, back ordered, canceled, and/or shipped.
- **Shipped To** is the address to which the order will be shipped or was shipped.
- **Ordered Item Summary** provides the product name, chosen attributes (if applicable), quantity ordered, price, and extended price for each item ordered. The product name is displayed as a hyperlink to the product detail content page.
- **Order Totals Summary** shows the total number of items ordered, subtotal, shipping, tax, discount, and grand total for the order.

Order Detail

These are the details for your order placed **6/3/2002**.

Order Status Shipped

Order Dates *Shipped 7/14/2002*
Processed 7/14/2002
Posted 6/3/2002

Shipped to Brian Middendorf
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Aeron Chair Size C Frame Color Graphite Pellicle Color Carbon Casters Hard Floor/Extra Thick Carpet Armpads Black Leather	2	\$909.99	\$1,819.98

	<u>Quantity</u>	<u>Total</u>
SubTotal	2	\$1,819.98
Shipping		\$70.00
Tax		\$158.34
Discount		<u>(\$0.00)</u>
Total		\$2,048.32

Order detail

Create Shopping List

All orders can be used as the basis for a shopping list. Specifying a unique name for the shopping list and clicking the **Create Shopping List** will create a shopping list with all of the products and quantities from the

order. The shopping list can be edited later to remove any unwanted items or to update the quantities of items. More details about editing shopping lists can be found in the [The Customer's Perspective > Account Administration > Shopping Lists](#) section.

Use Order for a Shopping List

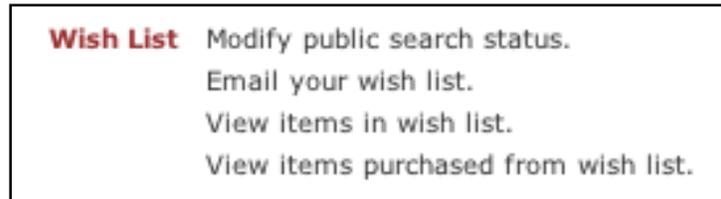
*** Shopping List Name**

Create shopping list form

Wish List

Wish lists provide a means for customers to window shop and save products in a list that may or may not be used at a later date to make purchases. Wish list based purchases can be made by the customer or by friends, family, etc of the customer. Wish lists are a feature that must be enabled in *Store Configuration*. Details about enabling wish lists can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

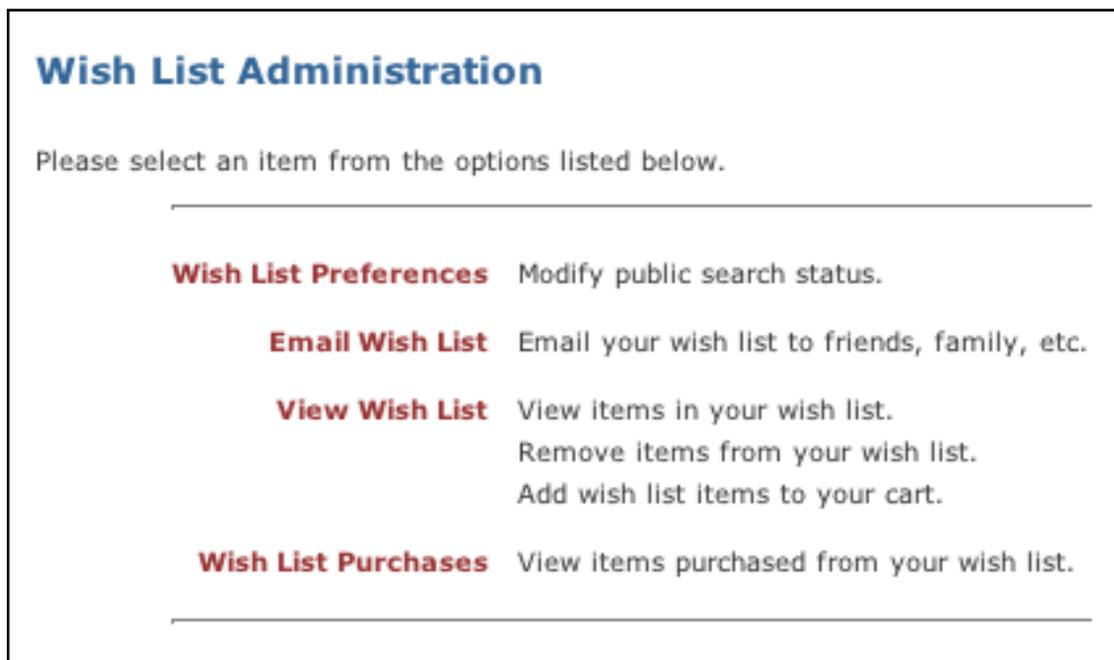
If the store has wish list functionality enabled, then the customer can view and modify their wish list entries by clicking the **Wish List** hyperlink on the *Customer Account Administration* menu.



Wish list hyperlink on the customer account administration menu

Wish List Administration Menu

Unlike many of the other areas in *Customer Account Administration*, wish lists have several administration related functions, each with a separate submenu. Each of the menu options are detailed in dedicated sections below.



Wish list administration menu

Wish List Preferences

Customers can elect whether or not their wish list is publicly searchable. By being publicly searchable, the wish list could appear in the search results listing when a site visitor is searching for a specific wish list. Customers that want to prevent their name, email address, city, state, and wish list items from being published can elect to have a private wish list.

The current search preference is displayed to the customer along with a button that will toggle the public search status of their wish list.

Wish List Preferences

Wish Lists may be configured to be searchable by the public. When a wish list is searchable, the only information that is viewable about you is your name, city, state, and country. The public can search for wish lists by owner name, email address, city, state, and country.

If you choose not to have a searchable wish list, the only way for someone to access your wish list is by you emailing a direct link to your wish list. Emailing wish lists is accomplished by selecting the Email Wish List option on the Wish List Administration page.

Your wish list **is** currently searchable by the public.

Click the button below to make your wish list not searchable.

[Change to Unsearchable](#)

Wish list preferences

View Wish List

All of the items currently in the wish list (all wish list items that have not been purchased) are listed along with the date the product was added to the wish list, the product name displayed as a hyperlink to the product detail content page, and the chosen product attributes (if applicable). Each item has an **Add to Cart** button and **Remove Item** hyperlink associated with it. Clicking the **Add to Cart** button will add the appropriate product to the shopping cart. The price and availability of each product is determined when the item is added. If a product is no longer for sale, currently out of stock, or the attributes have changed, then the product will not be added to the shopping cart and an appropriate message will be displayed to the customer. Clicking the **Remove Item** hyperlink will permanently remove the item from the wish list.

View Wish List

These are the items currently in your Wish List.

<u>Date Added</u>	<u>Product Name</u>		
7/22/2002	In-Building Repeater	ADD TO CART	Remove Item
7/5/2002	Lasso Web Data Engine 3.6.6 Operating System Macintosh Data Source Module 4D	ADD TO CART	Remove Item
5/20/2002	Aeron Chair Size C Frame Color Graphite Pellicle Color Carbon Casters Hard Floor/Extra Thick Carpet Armpads Black Leather	ADD TO CART	Remove Item
4/30/2002	R35 webColor Pad	ADD TO CART	Remove Item

[View wish list](#)

Email Wish List

Customers can notify friends, family, etc about their store wish list with a personal message and a direct link to their wish list. This feature can be a convenience to the end user because a direct wish list link prevents having to search for the wish list. This can also be a convenience to customers that want to keep their wish list private, i.e., not publicly searchable.

Email Wish List

Please complete the form below to email a direct link to your wish list. This email address is not stored in our databases and will never be sold or used for promotional purposes.

* denotes an empty required field.

* **Subject**

* **Recipient's Email Address**

Personal Message

The following text will be included here in your email...

My HostedStore Development Store wish list can be viewed at:

Link provided here

Send Email

Email wish list form

Wish List Purchases

All wish list items that have been purchased are listed along with the date the product was purchased, the product name displayed as a hyperlink to the product detail content page, and chosen product attributes (if applicable). The name of the person that purchased the wish list item is not displayed in case the item was purchased by someone for the customer as a gift.

Wish List Purchases

These are the items purchased from your Wish List.

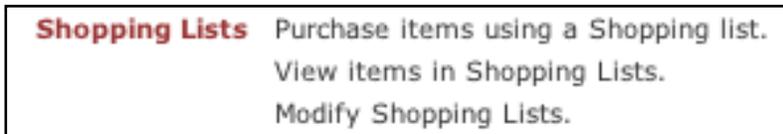
<u>Purchased</u>	<u>Product Name</u>
7/22/2002	Large Pizza Special Instructions Tomatoes after baking please. Side of Grated Parmesan Cheese No Thanks Side of Red Pepper Flakes No Thanks Topping 1 Canadian Bacon Topping 2 Pineapple Topping 3 Tomatoes, Sliced Topping 4 None Topping 5 None Topping 6 None Topping 7 None Topping 8 None Topping 9 None Topping 10 None
7/22/2002	Lasso Web Data Engine 3.6.6 Operating System Macintosh Data Source Module 4D
4/28/2002	HTML Magic Pack
4/28/2002	Custom Built Pentium IV Computer Case SuperMicro SC830W Server-Tower w/Dual 300W Processor Intel Pentium IV 1.5GHz / 400MHz FSB Cooling P4 Ball Bearing Fan Mainboard ASUS P4T Socket 423 400MHz System Memory 128MB 184-pin PC800 RDRAM Hard Drive Seagate 73.4GB Cheetah U160 SCSI Controller Adaptec 3400S U160 RAID Controller CD/DVD-ROM Toshiba 16x/48x EIDE DVD-ROM CD-Recordable Toshiba EIDE 4x/4x/4x/24x CD-RW/DVD-ROM Floppy Drive Teac 1.44MB Floppy Drive Video Card VisionTek GeForce3 64MB DDR AGP Modem U.S. Robotics 56k v.90 Internal ISA OEM Network Card 3COM 3C905C-TXNM 10/100Mbps PCI OEM Sound Card Creative SoundBlaster Live! Platinum 5.1 Speakers Monsoon MM-1000 Flat Panel Speaker System Keyboard Microsoft Natural Pro Elite USB OEM Mouse Microsoft IntelliEye PS/2 - USB Monitor ViewSonic VP181 18.1 Inch Digital LCD Power APC Back-UPS Pro 280 w/ PowerChute Bundle Operating System Microsoft Windows 2000 Server OEM Application Microsoft Office XP Professional Printer Hewlett Packard Desk Jet 930C Digital Imaging Canon PowerShot S110 System Build Build Fee

Wish list purchases

Shopping Lists

Shopping lists provide a means for customers to use past orders as a template for new orders. This feature saves time for customers that frequently order the same products or combinations of products. By saving a past order as a shopping list, all of the products in the order can be added to the shopping cart with the click of one button. More details about creating shopping lists can be found in the [The Customer's Perspective > Account Administration > View Orders](#) section. Shopping lists are a feature that must be enabled in *Store Configuration*. Details about enabling shopping lists can be found in the [Installation and Configuration > Store Configuration](#) section.

If the store has shopping lists functionality enabled, then the customer can view and modify their shopping lists by clicking the **Shopping Lists** hyperlink on the *Customer Account Administration* menu.



Shopping lists hyperlink on the customer account administration menu

View Shopping Lists

Customers can view a list of all of their shopping lists in this area of *Customer Account Administration*. The shopping lists are listed alphabetically by the name assigned by the customer along with the dates the list was created and last modified. Clicking the name of the shopping list will load the shopping list detail form content page.



A screenshot of a web page titled "View Shopping Lists" in blue. Below the title is the instruction "Click on the shopping list name to view the items in the list." followed by a table with three columns: "List Name", "Date Created", and "Date Modified". The table lists five shopping lists with their respective creation and modification dates.

<u>List Name</u>	<u>Date Created</u>	<u>Date Modified</u>
Audiovox Accessories	6/22/2002	6/22/2002
Coral Calcium	10/26/2002	10/26/2002
New Employee	4/25/2002	4/25/2002
R35 webColor Pad	4/28/2002	4/28/2002
Targus Leather Deluxe Backpack	3/13/2002	3/31/2002

View shopping lists

Shopping List Detail

The shopping list detail form content page displays each product in the shopping list and the quantity of the product to be ordered. Clicking the product name hyperlink will load the product detail content page. Three functions are available on the shopping list detail form content page:

- Clicking the **Add to Cart** will add all of the products in the shopping list to the shopping cart. The price and availability of each product is determined when the item is added. If a product is no longer for sale, currently out of stock, or the attributes have changed, then the product will not be added to the shopping cart and an appropriate message will be

displayed to the customer.

- Clicking the **Update List** button will update the shopping list with any quantity modifications that were made.
- Clicking the **Delete List** button will remove the shopping list from the table.

View Shopping List

To change the quantity of an item in your shopping list, enter the new quantity in the item's Quantity field and click the Update List button below. To remove an item from your shopping list, enter a 0 in the item's Quantity field and click the Recalculate button below. You may modify the quantities of multiple items at the same time. To completely remove the list, click the Delete List button below.

New Employee

<u>Product Name</u>	<u>Quantity</u>
Adjustable Headset	<input type="text" value="1"/>
Belt Clip	<input type="text" value="1"/>
3 Watt Booster Amplifier	<input type="text" value="1"/>
Cigarette Lighter Adapter	<input type="text" value="1"/>
Desktop Dual Charger	<input type="text" value="1"/>
Dash Mount Holder	<input type="text" value="1"/>

Add to Cart

Update List

Delete List

Shopping list detail form

Purchased Files

If the store sells downloadable files, then the customer can view and download their purchased files by clicking the **Purchased Files** hyperlink on the *Customer Account Administration* menu.



Purchased files hyperlink on the customer account administration menu

The file name, the date purchased, and a summary of file related activity are detailed for each file. The activity summary shows the download date deadline (if any), the download limit (if any), and the number of times the file has been downloaded. If the file is still eligible for download (the download date or and/or download limit have not been exceeded), then right-clicking or control-clicking on the file name hyperlink will download the file. When the file name hyperlink is clicked, the download counter is incremented. If the file is not eligible for download, then the file name will appear as text instead of a hyperlink.

Important: Once a download is initiated, the download counter is incremented. There is no guarantee that the download completes so cases may arise where a customer reaches a download limit, but the customer never receives the file. If limited downloads are used in a store, then it is advisable to have a policy and procedure in place for handling these types of events.

View Purchased Files

Important: Some browsers have difficulty properly handling the file download. The browser *Save As...* capability should be used to successfully download files.

1. Right-click (Windows OS) or Control-click (Mac OS) on the hyperlink.
2. Select the appropriate contextual menu option for saving downloaded documents/links.
3. Select an appropriate path for the downloaded file.
4. Save the file with the same name as displayed in the hyperlink.

<u>File Name</u>	<u>Purchased</u>	<u>Activity Summary</u>
<i>EAD.pdf</i>	3/16/2002	Download by: 4/15/2002 Download limit: 5 Times Downloaded: 0

Purchased file list

Logout

A customer can logout of their current *Customer Account Administration* session by clicking the **Logout** hyperlink on the *Customer Account Administration* menu.



Logout hyperlink on the customer account administration menu

All of the customer's store related cookies will be cleared and the *Customer Account Administration* login form will be displayed.

Account Administration Login

Please enter the email address and password associated with your account below. If you have forgotten your password, please enter your email address and check the checkbox. Your password will be emailed to you.

Email Address

Password

I forgot my password.

Customer account administration login form

The Affiliate's Perspective

Affiliate Account Administration provides the affiliate with the ability to login and view the history of their account. Each order that has been placed using the affiliate ID is detailed along, with the date their account was credited (ship date), the amount of the order, and the theoretical affiliate payout for the order. Note that orders will only be credited to an affiliate if the affiliate feature has been enabled in *Store Configuration*. Details about enabling the affiliate feature can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

Affiliate Account Administration is located at <http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=affiliate/index.lasso>, where *www.yourserver.here* is replaced with the fully qualified domain name associated with the HostedStore license. A link to *Affiliate Account Administration* is not provided in any of the default store template files because not all stores will have an affiliate program. It is, however, quite easy to modify any of the default store templates or content pages to provide a link if necessary. Details about customizing the default store templates and content pages can be found in the [Modifying HostedStore](#) > [Store Templates](#) and [Modifying HostedStore](#) > [Content Pages](#) sections, respectively.

Affiliate Login

The affiliate must login with their email address and password to access *Affiliate Account Administration*. The email address and password associated with an affiliate account are assigned when the affiliate account is created. Details about creating affiliate accounts can be found in the [Store Administration](#) > [Affiliate Administration](#) section.

Affiliate Login

Please enter the email address and password associated with your account below. If you have forgotten your password, please enter your email address and check the checkbox. Your password will be emailed to you.

Email Address

Password

I forgot my password.

Affiliate login form

Earnings and Payout Report

After the affiliate has successfully logged in, a report is presented that summarizes their account activity. The top part of the report summarizes the affiliate plan (the fixed rate value or percentage of order subtotal that the affiliate receives), the number of referrals from the affiliate, the number of referrals that placed orders, and the conversion rate (the number of referrals placing orders divided by the total number of referrals). The rest of the report details each order that was the result of a referral from the affiliate. The date ordered, date credited (date the order was shipped), subtotal, and theoretical payout (theoretical affiliate earnings) are shown for each order. If the affiliate has already been paid for a credited order, then the date of the payout, the check number, and the amount of the check are displayed. The cumulative subtotal and earnings are summarized for each month and for all orders.

View Orders

A summary of all orders credited to your affiliate account is provided below.
Your affiliate plan pays 25.00% of each order.

To date, you have referred 26 customers to our store.
6 of those customers placed orders for a conversion rate of 23.08%.
Orders that have shipped were credited to your account. Details are provided below.

<u>Credited</u>	<u>Ordered</u>	<u>SubTotal</u>	<u>Payout</u>	<u>Paid</u>	<u>Check</u>	<u>Amount</u>
3/17/2002	2/3/2002	\$33,524.20	\$8,381.05	7/29/2002	1500	\$8,391.05
3/17/2002	1/28/2002	\$5,224.93	\$1,306.23	5/26/2002	1345	\$1,306.23
3/31/2002	3/26/2002	\$19.95	\$4.99	5/20/2002	1234	\$54.99
		\$38,769.08	\$9,692.27			
4/28/2002	4/28/2002	\$200.00	\$50.00	5/20/2002	1234	\$54.99
		\$200.00	\$50.00			
5/26/2002	5/22/2002	\$954.95	\$238.74			
		\$954.95	\$238.74			
7/14/2002	4/24/2002	\$16,239.66	\$4,059.91			
7/14/2002	4/28/2002	\$8,646.71	\$2,161.68			
7/14/2002	5/1/2002	\$5.40	\$1.35			
7/14/2002	5/4/2002	\$909.99	\$227.50			
7/29/2002	7/29/2002	\$8,357.76	\$2,089.44			
		\$34,159.52	\$8,539.88			
Cumulative Total		\$74,083.55	\$18,520.89			

Affiliate earnings and payout report

The Supplier's Perspective

The supplier of each product in the store catalog can be assigned to the product. If the supplier has been granted the proper permissions (details can be found in the [Store Administration](#) > [Supplier Administration](#) > [Add New Supplier](#) section) and the product has been configured properly (details can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section), then the supplier will be able to log into *Supplier Account Administration* to view orders that have been processed and are ready for drop shipping.

The supplier logs in by navigating a browser to `http://www.yourserver.hereHostedStore.LassoApp?-ResponseLassoApp?-ResponseLassoApp=adminsupplier/index.lasso`, where `www.yourserver.here` is replaced with the fully qualified domain name associated with the HostedStore license. The supplier then logs in using the email address and password that have been assigned to them by the store manager.



The image shows a login form titled "Supplier Administration Login". It features two input fields: "Login" and "Password". Below the fields is a "Login" button. The form is enclosed in a rectangular border.

Supplier account administration login form

Sections

- [Order Queue](#) - List of orders with items to be drop shipped.
- [Modify Ordered Item Status](#) - Modify status of some or all of the items in an order.

The Order Queue

After successfully logging in, the supplier is presented with all orders that have been processed and are ready to be shipped in chronological order. The date/time the order was processed is displayed along with the invoice number, number of items in the order, and the city, state, postal code, and country from the shipping address.

Note that the supplier will only see the orders and the items in the order that they are to drop ship. For example, if the order has five ordered items, two from supplier A and three from supplier B (two that are drop shipped, one not), then when supplier B logs in for the first time they will see two items for the order in their queue because only two of the items are to be drop shipped by them. If they only ship one of the items during this session, then the next time they log in they will only see one item for the order in their queue. After that item is shipped, then the order will not appear in the queue the next time.

Order Queue				
<u>Date/Time Processed</u>	<u>Invoice</u>	<u>Items</u>	<u>Shipping Location</u>	<u>Postal Code</u>
7/30/2002, 11:15 AM	292	1	Bellevue, WA US	98008-0743
7/31/2002, 3:01 PM	293	5	Bellevue, WA US	98008-0743

Supplier order queue

Edit an Order

Clicking the invoice number hyperlink for an order in the queue will load the edit order form content page. The form that the supplier see functions like the edit order form the store manager sees (details can be found in the [Store Administration](#) > [Order Administration](#) > [Edit an Order](#) section), but it contains only the information that the supplier needs to ship the order. All other order information is suppressed.

Order Information

The top of the edit order form content page displays the order date, invoice number, and shipping address.

Ordered Items

The middle of the page displays a list of all the items in the order. The status of each item can be modified individually if necessary.

- **Qty** - The quantity of the item ordered. If this value is greater than one, then the item can be split into separate items.
- **Item** - Displays the product name and selected attributes (if applicable). Clicking the product name hyperlink will open a new browser window with the product detail content page.
- **Status** - The status of the item can be set to one of the following:
 - *Back Ordered* - Used for items for which shipping has been delayed because the item is not in stock.
 - *Shipped* - Used for items that have been shipped.
- **Split Item** - If more than one of an item was ordered, then a select list will be displayed that contains a list of numbers from one up to one less than the quantity ordered. Selecting a number in the list will split the item into two separate line items with identical properties and the appropriate quantities for each line item.

For example, if five of an item was ordered, then selecting **4** from the list will split the item into two items. One item will have a quantity of **1** and the second item will have a quantity of **4**. This feature allows for a different status to be applied to multiple quantity items. For example, if five of an item is ordered but only four are in stock, then the item can be split so the four items can be shipped and the other item can be back ordered.

- **Vendor** - Multiple shipping vendors are available for selection. If one of the vendors is selected, then the tracking number will be displayed as a hyperlink to the shipping tracking page on the vendor's Web site on any of the default content pages where a tracking number is displayed. If a vendor other than those listed is needed, be sure to make a feature request for the vendor to be added. In the meantime, *Other* can be used and the appropriate code modifications can be made to content pages to display the tracking number as a hyperlink if that functionality is wanted.
- **Tracking Number** - If a tracking number is entered in the text box, then a shipping vendor must be selected.

Global Order Modifications

It is possible to make global order modifications that will apply the status change to all ordered items. The status of the order can be set to one of the following:

- *Back Ordered* - Used for orders for which shipping has been delayed because ordered items are not in stock.
- *Shipped* - Used for orders for which all items have been shipped.

Multiple shipping vendors are available for selection. If one of the vendors is selected, then the tracking number will be displayed as a hyperlink to the shipping tracking page on the vendor's Web site on any of the default content pages where a tracking number is displayed. If a vendor other than those listed is needed, be sure to make a feature request for the vendor to be added. In the meantime, *Other* can be used and the appropriate code modifications can be made to content pages to display the tracking number

as a hyperlink if that functionality is wanted.

Shipping vendors currently available for selection:

- *Airborne* - Airborne
- *FedEx* - Federal Express
- *UPS* - United Parcel Service
- *USPS* - United States Postal Service

Besides *Other*, one other shipping vendor, *Pick Up*, is available for selection. That entry is provided for retail locations where the customer takes immediate delivery or for situations where the customer picks up the order instead of having it delivered.

Email Notifications

If the supplier has been configured for it, an email notification will be sent to the customer and/or the store manager when the status of the ordered items has been changed. Details about supplier configuration can be found in the [Store Administration](#) > [Supplier Administration](#) > [Add New Supplier](#) section.

Edit Order

Date/Time Created 7/31/2002, 3:01 PM	Shipping Address Jane Doe HostedStore Corporation PO Box 6743 Bellevue, WA 98008-0743 US bkm@speakeasy.org 425.644.7354 425.591.7354 (fax)
Invoice 293	

Qty	Item	Status	Vendor	Tracking Number
3	Extra Capacity NiMH AA Spare Battery (SNN4934)	<input type="text"/>	<input type="text"/>	<input type="text"/>
		Split Item <input type="text"/>		
2	Rapid Travel Charger (SPN4940)	<input type="text"/>	<input type="text"/>	<input type="text"/>
		Split Item <input type="text"/>		

Global Order Modifications

Change Status

Shipping Vendor

Tracking Number

Email status change to customer.

Edit order form

Store Administration

Store Administration provides access to all of the information pertinent to running the store. Products can be created and modified, customers and orders can be maintained, promotions and affiliates can be created and modified, and reports can be generated.

Sections

- [Overview](#) - Summary of features in *Store Administration*.
- [Logging In](#) - Accessing *Store Administration*.
- [Admin User Administration](#) - Add and edit administration users.
- [Affiliate Administration](#) - Add and edit affiliates.
- [Autoship Administration](#) - Add and edit autoship orders.
- [Customer Administration](#) - Add, edit, and email customers.
- [Maintenance](#) - Backup and export of database tables.
- [Modules](#) - Menu of installed modules.
- [Order Administration](#) - Add and edit orders.
- [Price Group Administration](#) - Add and edit price groups.
- [Product Administration](#) - Add and edit products.
- [Promotion Administration](#) - Add and edit promotions.
- [Reports](#) - Generate summary reports of store activity.
- [Review Administration](#) - Edit product reviews posted by customers.
- [Store Status](#) - Summary of orders in the queue and new orders and customers for the day.
- [Supplier Administration](#) - Add, edit, and email suppliers.

Overview

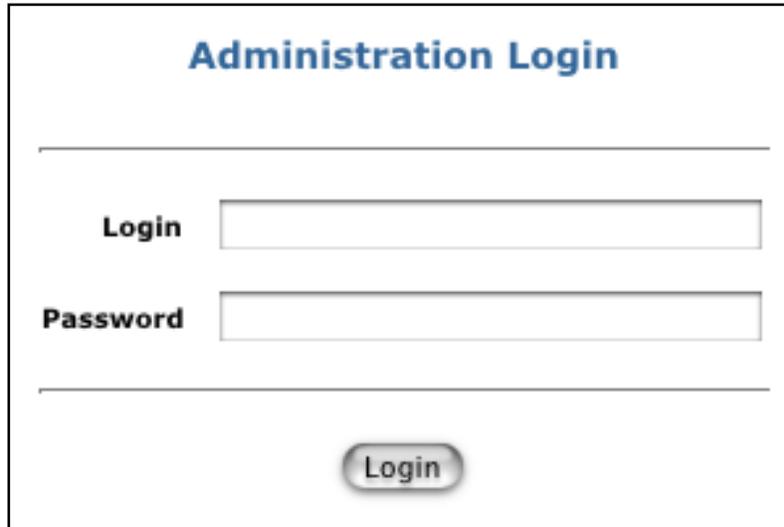
Store Administration is divided into fourteen sections: Admin Users, Affiliates, Autoship Orders, Customers, Maintenance, Modules, Orders, Price Groups, Products, Promotions, Reports, Reviews, Store Status, and Suppliers. A brief summary of each section is provided below:

- **Admin Users** - View, edit, and create users with specific *Store Administration* permissions. Users can be used to allow access to specific parts of *Store Administration* by different employees.
- **Affiliates** - View, edit, and create affiliates. Affiliates can be used to track referrals from third parties and generate payout information to compensate the affiliate. Affiliates can also be used as a means to track online marketing campaigns.
- **Autoship Orders** - View, edit, and create autoship orders. Autoship orders automatically reorder products for a customer at a specific time interval.
- **Customers** - View, edit, and create customers.
- **Maintenance** - Provides a mechanism to backup and export databases, check HostedStore versions, purge expired orders/carts, and purge checking account and credit card account information.
- **Modules** - If modules have been installed, then access to each module's administration interface (if applicable) will be available through this menu.
- **Orders** - Details for orders that have been placed. View and modify the order status for entire orders or individual ordered items. Add comments to orders and communicate with the customer about the status of their order. Create new orders placed via phone, mail order, in a retail establishment, etc.
- **Price Groups** - View, edit, and create price groups. Price groups can be used to globally modify product prices for groups of customers.
- **Products** - View, edit, and create products. Maintain the status of individual products: products for sale in the online store, featured products, products on sale, "members only" products. Determine images missing from the product catalog. Adjust prices or inventory for all products simultaneously. Upload product images or description files.
- **Promotions** - View, edit, and create promotions. Promotions are discounts that can be applied to specific products, specific customers, all products, shipping, etc.
- **Reports** - Several reports are provided to give details of store transactions. If a report is needed but not available please make a feature request.
- **Reviews** - View and edit customer submitted product reviews.
- **Store Status** - A summary overview of orders in the queue, new orders for the day, and new customers for the day.
- **Suppliers** - View, edit, and create suppliers. Suppliers allow for products to be drop shipped on behalf of the store.

Logging In

Store Administration is reached by navigating a browser to `http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=adminstore/index.lasso`, where `www.yourserver.here` is replaced with the fully qualified domain name associated with the HostedStore license. If *Store Administration* has not been accessed during the current browser session, then it will be necessary to log in.

The store administrator logs in using the username and password combination (the `HS_AdminLogin` and `HS_AdminPassword` preference variables, respectively) set in *Store Configuration*. Other store administration users (details can be found in the [Store Administration](#) > [Admin User Administration](#) section) should log in using the username and password they have been assigned.

The image shows a web form titled "Administration Login". It features a horizontal line at the top. Below the line, there are two input fields: the first is labeled "Login" and the second is labeled "Password". Both fields are empty text boxes. At the bottom of the form, there is a rounded rectangular button labeled "Login".

Store administration login form

After successfully logging in, the *Store Administration* menu will be displayed. The menu is simply a detailed listing of all of the functions available in each administration area. Note that for non-store administrator users some of the administration areas may not be available or only a subset of the functions for a given area may be enabled. The functions available will be determined by the user's privileges. Details about administration user privileges can be found in the [Store Administration](#) > [Admin User Administration](#) section.

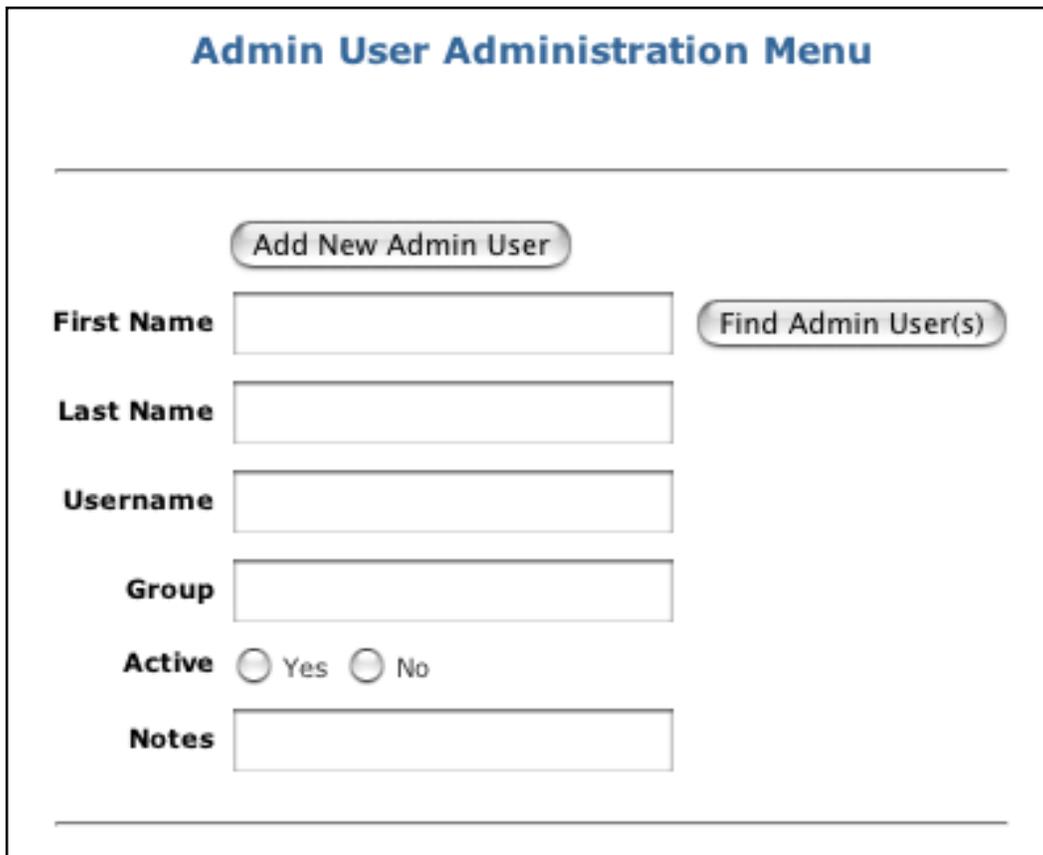
If already logged in, all subsequent returns to the *Store Administration* home page will display the store status page instead of the *Store Administration* menu. Details about the store status page can be found in the [Store Administration](#) > [Store Status](#) section. A link to the *Store Administration* menu is provided at the bottom of each page in *Store Administration*, but there is likely to be little need for the menu once the purpose of each administration area is learned because links to each area are provided in the navigation menu at the top of each page.

Admin User Administration

Administration users are users that have access to *Store Administration*. Each user can have their own set of permissions that allows access to all or a subset of the features available to the global store administrator defined in *Store Configuration*.

Administration users are useful for stores that will have multiple people performing various tasks within *Store Administration* who should not necessarily have access to all areas. For example, some users may be in charge of processing orders, but should not have access to changing admin users or products, and viewing reports.

The *Admin User Administration* menu provides for two different operations: Add New Admin User and Find Admin User(s).



The image shows a screenshot of the 'Admin User Administration Menu'. At the top, the title 'Admin User Administration Menu' is centered. Below the title is a horizontal line. There are two buttons: 'Add New Admin User' on the left and 'Find Admin User(s)' on the right. Below these buttons are several input fields: 'First Name', 'Last Name', 'Username', 'Group', 'Active' (with radio buttons for 'Yes' and 'No'), and 'Notes'. Each field is a simple rectangular box. The form is enclosed in a thin black border.

Admin user administration menu

Sections

- [Add New Admin User](#) - Create a new admin user.
- [Find Admin User\(s\)](#) - Search for and edit existing admin users.

Add New Admin User

Create a new administration user by clicking the **Add New Admin User** button. A new page with a form for adding administration user information will be loaded in the browser.



Add admin user form area of the admin user administration menu

Add Admin User Form

All of the fields on this form correspond to fields in the **AdminUser** table only. In addition to basic user information, multiple parameters are used to set *Store Administration* privileges. An explanation of each field and its corresponding field in the **AdminUser** table are detailed by group below.

After all fields have been completed, click the **Create Admin User** button at the bottom of the form to add the user to the table.



Create admin user button on the add admin user form

User Information

- **Active** corresponds to the **Status** field in the table (*REQUIRED*). Designates whether or not the admin user will be able to login to *Store Administration*.
- **Username** corresponds to the field of the same name in the table (*REQUIRED*).
- **Password** corresponds to the field of the same name in the table (*REQUIRED*).
- **Group** corresponds to the field of the same name in the table.
- **First Name** corresponds to the **FirstName** field in the table.
- **Last Name** corresponds to the **LastName** field in the table.
- **Notes** corresponds to the field of the same name in the table.

* denotes an empty required field.

* **Active** Yes No

* **Username**

* **Password**

Group

First Name

Last Name

Notes

User information area of the add admin user form

User Privileges

Each of the privileges listed below, when checked, will give the user the ability to perform all actions that are available in the corresponding area in *Store Administration*.

- **Affiliates** grants access to *Affiliate Administration*.
- **Autoship** grants access to *Autoship Administration*.
- **Customers** grants access to *Customer Administration*.
- **Maintenance** grants access to the *Maintenance* area.
- **Modules** grants access to the third-party module menu.
- **Orders** grants access to *Order Administration*.
- **Price Groups** grants access to *Price Group Administration*.
- **Products** grants access to *Product Administration*.
- **Promotions** grants access to *Promotion Administration*.
- **Reports** grants access to *Reports*.
- **Reviews** grants access to *Review Administration*.
- **Store Configuration** grants access to *Store Configuration*.
- **Store Status** grants access to the store status page.
- **Suppliers** grants access to *Supplier Administration*.
- **Users** grants access to *Admin User Administration*.

	Allow
Affiliates	<input type="checkbox"/>
Autoship	<input type="checkbox"/>
Customers	<input type="checkbox"/>
Maintenance	<input type="checkbox"/>
Modules	<input type="checkbox"/>
Orders	<input type="checkbox"/>
Price Groups	<input type="checkbox"/>
Products	<input type="checkbox"/>
Promotions	<input type="checkbox"/>
Reports	<input type="checkbox"/>
Reviews	<input type="checkbox"/>
Store Configuration	<input type="checkbox"/>
Store Status	<input type="checkbox"/>
Suppliers	<input type="checkbox"/>
Users	<input type="checkbox"/>

User privileges area of the add admin user form

Find Admin User(s)

A search form is provided to search for existing administration users. If more than one match is found, then a list of all matching administration users will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching administration user is displayed in an edit administration user form. The fields for editing an administration user are the same as the fields available when adding a new administration user.

The fields that can be searched are listed below along with the operator used in the search:

- **FirstName** - Contains (cn)
- **LastName** - Contains (cn)
- **Username** - Contains (cn)
- **Active** - Equals (eq)
- **Notes** - Contains (cn)
- **AdminGroup** - Contains (cn)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The image shows a search form for finding admin users. It consists of several input fields and a search button. The fields are labeled: First Name, Last Name, Username, Group, Active, and Notes. The Active field has two radio buttons labeled 'Yes' and 'No'. A button labeled 'Find Admin User(s)' is located to the right of the First Name field.

Admin user search form area of the admin user administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate administration user in the listing and click the **ID** hyperlink for the user. The administration user will be loaded into the edit form. The form fields and process for editing administration users is similar to the process for adding an administration user detailed in the [Store Administration](#) > [Admin User Administration](#) > [Add New Admin User](#) section.

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the

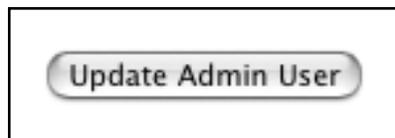
right of the column name.

Click on the username to edit the user record.

<u>Username</u> ▲▼	<u>First Name</u> ▲▼	<u>Last Name</u> ▲▼	<u>Active</u> ▲▼	<u>Group</u> ▲▼
testuser	Test	User	Y	Group1
testuser2	Test	User2	Y	Group2

Admin user search results list

After all fields have been modified, click the **Update Admin User** button at the bottom of the edit form to update the administration user record in the table.



Update admin user button on the edit admin user form

Affiliate Administration

The *Affiliate Administration* menu provides for six different operations: Add New Affiliate, Find Affiliate(s), Affiliate(s) Awaiting Payout listing, Post Payout(s), and Email Monthly Reports.

For any field requiring a monetary value, the value should be expressed as a decimal value without currency symbols. Also, be sure to use a decimal point (.) as the currency delimiter even if a different currency delimiter, a comma (,) for instance, is used for display purposes. For any field requiring a percentage value, the value should be expressed as a decimal without the percent symbol. For example, 15% would be expressed as .15.

Affiliate Administration Menu

Affiliate ID

Affiliate ID

Last Name

Company

Plan

Status

Operator And Or

Limit

Affiliate ID

Affiliate administration menu

Sections

- [Add New Affiliate](#) - Create a new affiliate.
- [Find Affiliate\(s\)](#) - Search for and edit existing affiliates.

- [Affiliate\(s\) Awaiting Payout](#) - List affiliates awaiting payout.
- [Post Payout\(s\)](#) - Post affiliate payouts.
- [Email Monthly Reports](#) - **(Future Feature)**
- [Example Uses for Affiliates](#) - Some suggestions for how to use affiliates.

Add New Affiliate

Create a new affiliate by selecting a unique affiliate ID, entering it in the **Affiliate ID** field, and clicking the **Add New Affiliate** button. The system will verify that the affiliate ID selected is unique. If not, a message will prompt for the selection of a new affiliate ID.



Add affiliate form area of the affiliate administration menu

The chosen affiliate ID should be used by the affiliate in all of their links to the store. The affiliate ID is passed as the value for the affiliate ID parameter, the `HS_AffiliateIDName` preference variable, that was set in *Store Configuration*. For example, if the affiliate ID is *HostedStore* and the affiliate ID parameter name is *AID*, then a link to the home page of the HostedStore store would look like:

<http://www.yourserver.here/HostedStore.LassoApp?AID=HostedStore>.

The affiliate ID can be passed to any page in the store. So, for example, to pass the affiliate ID in a link to the product detail content page: <http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=EncryptedProductID&AID=HostedStore>

Add Affiliate Form

The top of the add affiliate form displays the chosen affiliate ID. All of the fields on this form correspond to fields in the **Affiliate** table only. In addition to basic contact information, multiple parameters are used to determine if an affiliate is eligible for credit for a sale and how they are to be compensated. An explanation of each field and its corresponding field in the **Affiliate** table are detailed by group below.



Affiliate ID area of the add affiliate form

After all fields have been completed, click the **Create Affiliate** button at the bottom of the form to add the affiliate to the database.



Create affiliate button on the add affiliate form

Contact Information

- **First Name** corresponds to the **FirstName** field in the table (*REQUIRED*).
- **Last Name** corresponds to the **LastName** field in the table (*REQUIRED*).
- **Company** corresponds to the field of the same name in the table.
- **Address** and the second unlabeled address field correspond to the **Address1** and **Address2** fields in the table, respectively (*REQUIRED*).
- **City** corresponds to the field of the same name in the table (*REQUIRED*).
- **State** corresponds to the field of the same name in the table (*REQUIRED*).
- **Postal Code** corresponds to the **Postal** field in the table (*REQUIRED*).

- **Country** corresponds to the field of the same name in the table (*REQUIRED*).
- **Email** corresponds to the field of the same name in the table (*REQUIRED*). The username for the affiliate when logging into *Affiliate Account Administration*. It is also used to email the affiliate their monthly sales total if that function is used.
- **Phone** corresponds to the field of the same name in the table.
- **Fax** corresponds to the field of the same name in the table.
- **Password** corresponds to the field of the same name in the table (*REQUIRED*). The password for the affiliate when logging into *Affiliate Account Administration*

Contact Information

* **First Name**

* **Last Name**

Company

* **Address**

* **City**

* **State**

* **Postal Code**

* **Country**

* **Email**

Phone

Fax

* **Password**

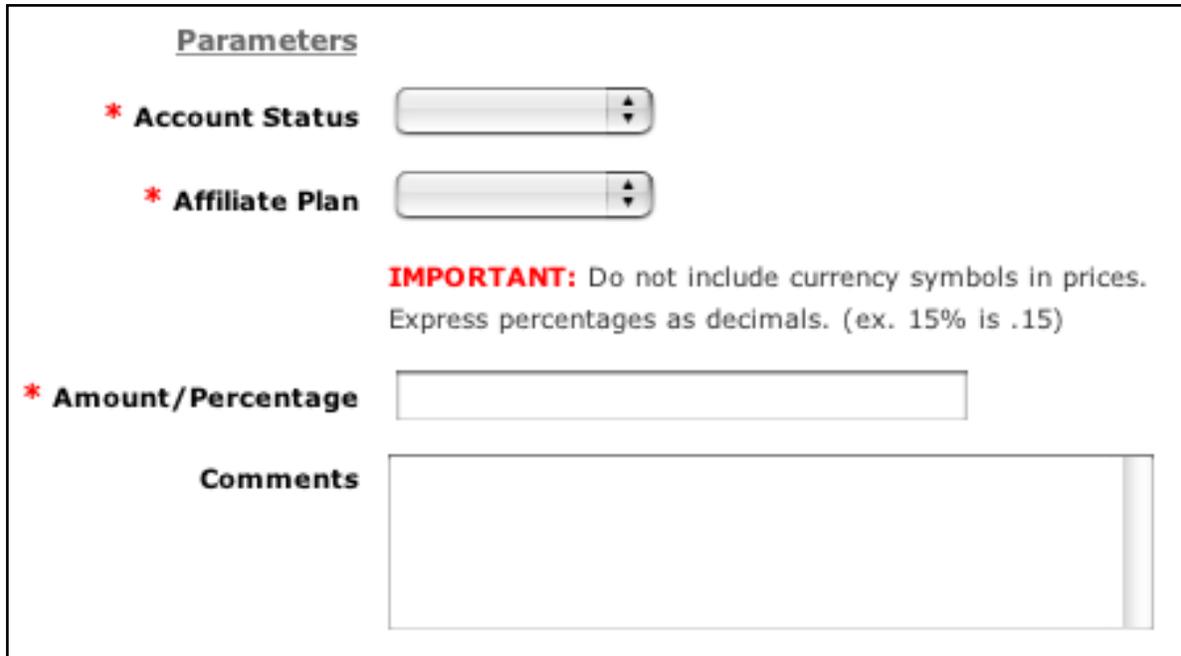
Contact information area of the add affiliate form

Parameters

- **Account Status** corresponds to the **Status** field in the table (*REQUIRED*). The current status of the affiliate. Only those affiliates with a status of *Active* will be credited with a sale even if a customer navigates to the store by way of an affiliate's hyperlink.
- **Affiliate Plan** corresponds to the **CompensationPlan** field in the table (*REQUIRED*). Specifies how the affiliate is to be compensated for their customer referral. *Fixed* means that the affiliate receives a

flat monetary value for their referral and *Percentage* means the affiliate receives a percentage of the order subtotal.

- **Amount/Percentage** corresponds to the **CompensationAmount** field in the table (*REQUIRED*). Sets the compensation amount. Use either a fixed monetary value or a percentage as determined by the value chosen for **Affiliate Plan**. Fixed monetary values should be expressed as a decimal value without currency symbols. Also, be sure to use a decimal point (.) as the currency delimiter even if a different currency delimiter, a comma (,) for instance, is used for display purposes. Percentages should be expressed as a decimal value. For example, 15% is expressed as .15 and 100% is expressed as 1.00.
- **Comments** corresponds to the **AdminComments** field in the table. Provided to keep notes, if needed, about that specific affiliate. The affiliate is not able to see the contents of this field--it is strictly for internal use only.



The screenshot shows a form titled "Parameters" with the following fields and instructions:

- * Account Status**: A dropdown menu.
- * Affiliate Plan**: A dropdown menu.
- IMPORTANT:** Do not include currency symbols in prices. Express percentages as decimals. (ex. 15% is .15)
- * Amount/Percentage**: A text input field.
- Comments**: A large text area for notes.

Parameters area of the add affiliate form

Find Affiliate(s)

A search form is provided to search for existing affiliates. If more than one match is found, then a list of all matching affiliates will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching affiliate is displayed in an edit affiliate form. The form fields for editing an affiliate are the same as the fields available when adding a new affiliate.

The fields that can be searched are listed below along with the operator used in the search:

- **AffiliateID** - Equals (eq)
- **LastName** - Contains (cn)
- **Company** - Contains (cn)
- **AffiliatePlan** - Equals (eq)
- **Status** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The screenshot shows a search form with the following elements:

- Affiliate ID**: A text input field.
- Last Name**: A text input field.
- Company**: A text input field.
- Plan**: A dropdown menu.
- Status**: A dropdown menu.
- Operator**: Radio buttons for **And** (selected) and **Or**.
- Limit**: A text input field.
- Find Affiliate(s)**: A button.

Affiliate search form area of the affiliate administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate affiliate in the listing and click the **ID** hyperlink for the affiliate. The affiliate will be loaded into the edit form. The form fields and process for editing affiliates is similar to the process for adding an affiliate detailed in the [Store Administration](#) > [Affiliate Administration](#) > [Add New Affiliate](#) section.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.

Affiliate found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the affiliate name to edit.

<u>Affiliate ID'</u> ▲▼	<u>Last Name</u> ▲▼	<u>Company</u> ▲▼	<u>Plan</u> ▲▼	<u>Status</u> ▲▼
HostedStore	Middendorf	HostedStore Corporation	Percentage	Active
SampleAffiliate	Test		Fixed	Active

Affiliate search results list

After all fields have been modified, click the **Update Affiliate** button at the bottom of the edit form to update the affiliate record in the table.

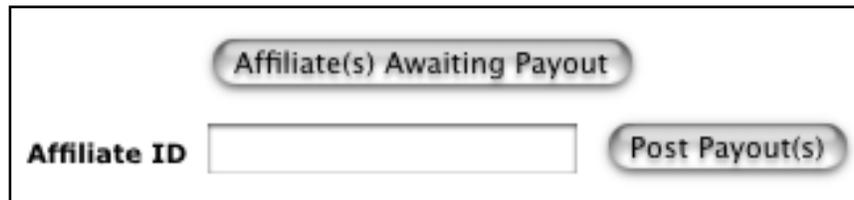


Update affiliate button on the edit affiliate form

Affiliate(s) Awaiting Payout

For large affiliate programs it would be cumbersome to individually enter each affiliate ID to process payouts. The affiliates awaiting payout listing addresses that problem by providing a summary of each affiliate with orders that qualify for payout. Using the listing, payouts can be quickly processed for only those affiliates with qualifying orders.

The affiliates awaiting payout listing is loaded by clicking the **Affiliate(s) Awaiting Payout** button on the affiliate administration menu.



A screenshot of a web interface showing a button labeled "Affiliate(s) Awaiting Payout" at the top. Below it is a text input field labeled "Affiliate ID" and another button labeled "Post Payout(s)".

Payout area of the affiliate administration menu

The listing is split into two sections. The top section displays a list of *Affiliates Awaiting Payout*. An affiliate has an order awaiting payout if the order has a status of *Shipped* and the affiliate has not been compensated for the order yet (the **AffiliatePayout** field is empty). The **Invoice** table is also checked for affiliates with orders that have been completed (orders that are not in the shopping cart), but have not been shipped or canceled. Affiliates that meet the criteria are listed in the *Affiliates with Orders in the Queue* section.

Affiliates Awaiting Payout

The affiliate ID, compensation plan, compensation amount, a count of all orders awaiting payout, the cumulative subtotal, and the calculated cumulative payout based on the affiliate's compensation plan are listed for all affiliates that have orders awaiting payout. The affiliate ID is hyperlinked to the affiliate payout form for the affiliate. Details about the affiliate payout form can be found in the [Store Administration](#) > [Affiliate Administration](#) > [Post Payout\(s\)](#) section.



A screenshot of a table titled "Affiliates Awaiting Payout". Below the title is a instruction: "Click the affiliate ID to post payout(s) for the affiliate." The table has six columns: Affiliate ID, Plan, Amount, Orders, SubTotal, and Payout. One row is visible with the following data: HostedStore, Percentage, 25.00%, 3, \$35,389.14, \$8,847.28.

<u>Affiliate ID</u>	<u>Plan</u>	<u>Amount</u>	<u>Orders</u>	<u>SubTotal</u>	<u>Payout</u>
HostedStore	Percentage	25.00%	3	\$35,389.14	\$8,847.28

Affiliates awaiting payout listing

Affiliates with Orders in the Queue

The affiliate ID, compensation plan, compensation amount, a count of all orders in the queue, the cumulative subtotal, and the calculated cumulative payout based on the affiliate's current compensation plan are listed for all affiliates that have orders in the queue.

Affiliates with Orders in the Queue

<u>Affiliate ID</u>	<u>Plan</u>	<u>Amount</u>	<u>Orders</u>	<u>SubTotal</u>	<u>Payout</u>
HostedStore	Percentage	25.00%	4	\$25,801.76	\$6,450.44

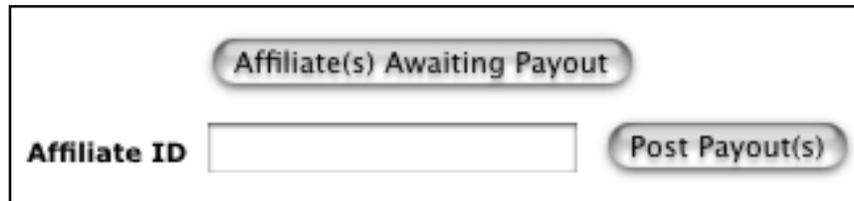
Affiliates with orders in the queue listing

Important: There is no concept of time with regards to affiliate compensation plans when payouts are calculated. The calculated cumulative payout is only an estimate based on the affiliate's current compensation plan. If the plan has changed when the order is shipped, then the payout for all orders that ship after the plan change will be based on the new plan, not the old plan that was in effect when the order was placed. That makes sense according to the business rules under which HostedStore operates. Orders do not qualify for payout (and hence the actual payout amount is not calculated) until shipped even if the order was placed several weeks prior and was placed on backorder. If this does not sync with the business rules under which the store operates, then once the order is shipped the proper payout can be made by issuing a check with the adjusted amount rather than the amount calculated based on the current compensation plan. More details about issuing payouts can be found in the [Store Administration](#) > [Affiliate Administration](#) > [Post Payout\(s\)](#) section.

Post Payout(s)

Compensation checks sent out to affiliates can be tracked by posting each payout. Note that this feature is not meant to serve any accounting type functions. This feature is meant solely for informational purposes only. By entering in the checks sent to affiliates, affiliates can see which orders they have already been paid for and which are outstanding on the default affiliate earnings and payout report content page (*/hostedstore/affiliate/html/orders_list.lasso*).

The affiliate payout form is loaded by entering the affiliate's ID in the **Affiliate ID** field on the affiliate administration menu and clicking the **Post Payout(s)** button.

A screenshot of a web form for affiliate payouts. At the top, there is a rounded rectangular button labeled "Affiliate(s) Awaiting Payout". Below this, on the left, is the text "Affiliate ID" followed by a rectangular text input field. To the right of the input field is another rounded rectangular button labeled "Post Payout(s)".

Payout area of the affiliate administration menu

The payout form is split into two sections. The top section displays a list of *Orders Awaiting Payout*. Orders are awaiting payout if the order has a status of *Shipped* and the affiliate has not been compensated for the order yet (the **AffiliatePayout** field is empty). The **Invoice** table is also checked for all orders attributed to the affiliate that have been completed (the orders are not in the shopping cart), but have not been shipped or canceled. Orders that meet the criteria are listed as *Orders in the Queue*.

Orders Awaiting Payout

A count of all orders awaiting payout, the cumulative subtotal, and the calculated cumulative payout based on the affiliate's compensation plan is presented at the top of the form followed by a list of all orders awaiting payout. Each order is listed with its invoice number, subtotal, and calculated payout. To issue a payout for the affiliate:

1. Check the checkbox for each order that will be paid with the check issued for this payout.
2. Enter the date, check number, and amount for the check. Note that the check number is stored as text so if special formatting is wanted be sure to include all of the appropriate information in the field. The check amount value should be expressed as a decimal value without currency symbols. Also, be sure to use a decimal point (.) as the currency delimiter even if a different currency delimiter, a comma (,) for instance, is used for display purposes. The appropriate formatting as set in *Store Configuration* will be used to display the value.

Affiliate ID HostedStore
Plan Percentage (25.00%)

Orders Awaiting Payout

Orders 3
SubTotal \$35,389.14
Payout \$8,847.28

Check the orders that are to be paid by the check.

	<u>Invoice</u>	<u>SubTotal</u>	<u>Payout</u>
<input type="checkbox"/>	71	\$33,524.20	\$8,381.05
<input type="checkbox"/>	211	\$909.99	\$227.50
<input type="checkbox"/>	216	\$954.95	\$238.74

* **Date Issued**

* **Check Number**

* **Amount**

Orders awaiting payout section of payout form

Important: There is no concept of time with regards to affiliate compensation plans when payouts are calculated. For example, if an affiliate is paid 10% of each order, has two qualifying orders, and then two more orders are added after their compensation plan is changed to 15%, then four orders will be listed as awaiting payout under the 15% compensation plan. To prevent this problem, it is recommended that all orders awaiting payout are paid prior to changing an affiliate's compensation plan. Note that this will not address the orders that are in the queue (details can be found under the *Orders in the Queue* heading below) under the old plan. Once shipped, those orders will be listed under the new compensation plan. That makes sense according to the business rules under which HostedStore operates. Orders do not qualify for payout until shipped even if the order was placed several weeks prior and was placed on backorder. If this does not sync with the business rules under which the store operates, then the proper payout can be made by issuing a check with the adjusted amount rather than the calculated amount shown in the list of orders awaiting payout. Note that proper adjustments should probably be made to the affiliates order list (see below) otherwise the affiliate may be confused as to why the check received does not match the calculated payout shown in their orders list.

The payout information entered is stored in the **AffiliatePayout** field as values separated by two adjacent pipe characters (| |). The order of the information stored is detailed below (the number corresponds to the location in the delimited string):

1. The date the check was issued in YYYY-MM-DD format.
2. The check number.
3. The amount of the check.
4. The affiliate compensation plan (the **CompensationPlan** field from the *Affiliate* table). This value is tracked because the affiliate plan may change at a later date without record of what plan was used to calculate the payout for the order.

5. The affiliate compensation amount (the **CompensationAmount** field from the *Affiliate* table). This value is tracked because the affiliate compensation amount may change at a later date without record of what compensation amount was used to calculate the payout for the order.
6. The calculated payout for the order.

The payout information is available to affiliates on the default orders list content page (*/hostedstore/affiliate/html/orders_list.lasso*). Details about the orders list can be found in the [The Affiliate's Perspective](#) section.

View Orders

A summary of all orders credited to your affiliate account is provided below.
Your affiliate plan pays 25.00% of each order.

To date, you have referred 25 customers to our store.
3 of those customers placed orders for a conversion rate of 12.00%.
Orders that have shipped were credited to your account. Details are provided below.

<u>Credited</u>	<u>Ordered</u>	<u>SubTotal</u>	<u>Payout</u>	<u>Paid</u>	<u>Check</u>	<u>Amount</u>
3/17/2002	2/3/2002	\$33,524.20	\$8,381.05			
3/17/2002	1/28/2002	\$5,224.93	\$1,306.23	5/26/2002	1345	\$1,306.23
3/31/2002	3/26/2002	\$19.95	\$4.99	5/20/2002	1234	\$54.99
		\$38,769.08	\$9,692.27			
4/28/2002	4/28/2002	\$200.00	\$50.00	5/20/2002	1234	\$54.99
		\$200.00	\$50.00			
5/26/2002	5/22/2002	\$954.95	\$238.74			
		\$954.95	\$238.74			
Cumulative Total		\$39,924.03	\$9,981.01			

Affiliate orders list

Orders in the Queue

A list of orders in the queue is also provided to serve as a convenient summary of how many current orders will be eligible for payout once shipped.

Orders in the Queue

Orders 4

SubTotal \$25,801.76

Payout \$6,450.44

The orders listed below have not shipped yet so are not eligible for payout.

<u>Invoice</u>	<u>SubTotal</u>	<u>Payout</u>
185	\$16,239.66	\$4,059.91
196	\$8,646.71	\$2,161.68
200	\$5.40	\$1.35
201	\$909.99	\$227.50

Orders in the queue section of payout form

Email Monthly Reports

(Future Feature)

Example Uses for Affiliates

The affiliate system in HostedStore is very flexible and can be used in creative ways to address many needs outside of a standard affiliate program. Instead of a classic affiliate program, think of the affiliate feature as a hyperlink generated sales tracking tool. Here are some examples of its uses:

- **Banner Ad Campaigns** - Sales generated from banner ads can be tracked by creating an affiliate for each campaign and passing the ID in the URL that links to the store from the ad. If the entry point will be the home page, the `HS_AffiliateIDName` preference variable is set to `AID`, and the `AffiliateID` is `Campaign1`, then the hyperlink would look like:
`http://www.yourserver.here/HostedStore.LassoApp?AID=Campaign1`, where `www.yourserver.here` is replaced by the fully qualified domain name associated with the HostedStore license.
- **Internal Link Tracking** - If the store is used as part of a site with non-store content, hyperlinks to the store from specific pages in the site can each have their own affiliate ID. Sales generation from different site pages/content can then be determined to aid in analysis of what triggers the action for customers to enter the store and purchase. If an affiliate ID is already registered for the customer then it will not be overwritten. This feature allows a customer to follow one hyperlink into the store, exit the store to non-store content, and then go back to the store via a new hyperlink while preserving the affiliate ID from the hyperlink which initially prompted them to enter the store.
- **Newsletter/Email Response Tracking** - If periodic newsletters or emails are sent to customers to announce new product offerings, new site features, etc, use a unique affiliate to track the number of sales that result from the news. The number of sales or cumulative monetary sales volume can be compared to the number of announcements sent out to determine the success of the campaign.
- **Market Tests** - Prepare multiple versions of images, copy, newsletters, etc that will be used in the same location (a few different banner ads at the same site, a couple versions of a newsletter where one version is mailed to one set of customers and the other is mailed to the rest, etc) and assign each its own affiliate ID. Compare the success of each version to gain insight into what entices people to visit the store and purchase.

Note that none of the suggestions above will measure the source of traffic generated by a specific affiliate. The affiliate feature only tracks sales that result from affiliate hyperlinks. If traffic analysis is needed, then compare hyperlinks with affiliate IDs that appear in the Web server traffic logs. That information can be very useful to determine conversion rates, i.e., which affiliate hyperlinks generate traffic to the store and what percentage of those store visits result in a sale.

Autoship Administration

The Autoship Administration menu provides for three different operations: Add New Autoship Order, Find Autoship Order(s), and Manual Autoship Order Sweep.

Autoship Administration Menu

Customer ID

Customer ID

Active Yes No

Date Ordered

Next Order

Last Name

City

State

Country

Operator And Or

Autoship administration menu

Sections

- [Add New Autoship Order](#) - Create an autoship order.
- [Find Autoship Order\(s\)](#) - Search for and edit existing autoship orders.
- [Edit an Autoship Order](#) - Edit an existing autoship order.
- [Manual Autoship Order Sweep](#) - Create orders from autoship orders.

Add New Autoship Order

Autoship orders can be entered manually to facilitate the use of HostedStore for mail order sales, phone sales, retail sales, and any other sales process where the customer will not be placing the order via the online store.

Enter the ID for the customer that is placing the autoship order and click the **Add New Autoship Order** button. If the order is for a new order, then the customer will need to be created first in the *Customer Administration*. Details about adding a new customer can be found in the [Store Administration](#) > [Customer Administration](#) > [Add New Customer](#) section.



The image shows a rectangular form area with a thin black border. On the left side, the text "Customer ID" is displayed in a bold, black font. To the right of this text is a white rectangular input field with a thin grey border. Further to the right is a rounded rectangular button with a grey gradient and the text "Add New Autoship Order" in white.

Add autoship order form area of the autoship administration menu

Autoship Parameters

Several parameters must be set for an autoship order to function properly. Explanations for each form field and its corresponding field in the **AutoShipOrderQueue** table are detailed below:

- **Active** corresponds to the field of the same name in the table. Select the *Yes* radio button if the autoship order is still to be used to generate new orders. Otherwise, select the *No* radio button.
- **Frequency** corresponds to the field of the same name in the table. Insert the day interval at which the autoship order is to be executed. After an autoship order is executed, the value will be added to the date in the **Next Order Date** field.
- **Restricted Duration** corresponds to the **RestrictedDuration** field in the table. If the autoship order is to be executed a set number of times, then select the *Yes* radio button. If the autoship order is to continue indefinitely, then select the *No* radio button.
- **Duration** corresponds to the field of the same name in the table. If the autoship order has a restricted duration, then specify the number of times remaining that the order is to be executed. This value will decrement by one each time the autoship order is executed.
- **Next Order Date** corresponds to the **NextOrderDate** field in the table. Set the month, day, and year to the next date that the autoship order is to be executed.
- **Order Action** corresponds to the **OrderAction** field in the table. If the autoship order is to be automatically processed through the payment gateway, then select the *Process* radio button. If the autoship order is to be saved as a pending order, then select the *Pending* radio button.

Important: Automatic processing of orders (selecting the *Process* radio button) is not supported in this version of HostedStore. All autoship generated orders will be created with a pending status.

- **Notes** corresponds to the field of the same name in the table. This field can be used for notes about the autoship order that will be used for internal use only. The notes do not affect the autoship order nor are the notes viewable by the customer.

Parameters

* **Active** Yes No

* **Frequency** days

* **Restricted Duration** Yes No

Duration times

* **Next Order Date**

* **Order Action** Process Pending

Notes

Parameters area of the add autoship order form

Ordered Item(s)

This field is used to store the information used to generate the ordered items for the order. It is important that the information is in a specific format that is strictly adhered to otherwise the order generation will fail. The information is stored in the **Item** field in the table.

Each item for the order must be entered into the form field in the following order:

```

ProductID
Quantity
Price
Attribute 1 Name||Attribute 1 Value||Attribute 1 Price Adjustment||Attribute 1 Part
Number||Attribute 1 Part Quantity
...
Attribute n Name||Attribute n Value||Attribute n Price Adjustment||Attribute n Part
Number||Attribute n Part Quantity
***END ITEM***

```

A few important items that may not be clear from the way the explanation renders in the browser or PDF:

- Each of the lines above should be followed by a carriage return in the form field with the exception of the very last *****END ITEM***** entry. It should not be followed by a carriage return.
- All of the information for a single attribute should appear on one line followed by a carriage return.
- Multiple items can be added by repeating the pattern after the *****END ITEM***** entry for the preceding item.

Example:

```

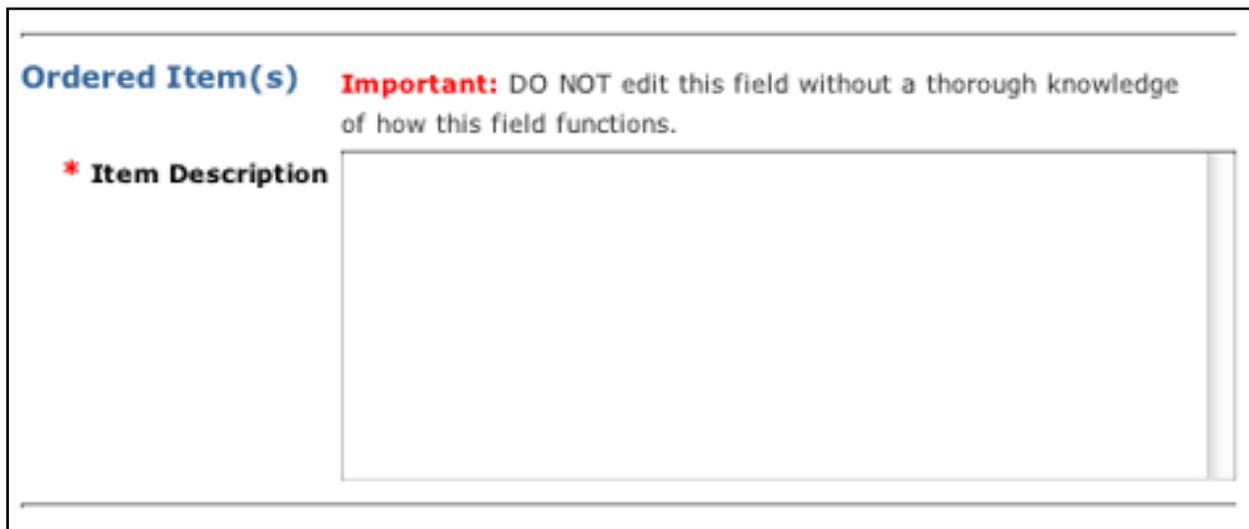
tshirt
2
10.00
Size||Large|||||

```

```
Color|Black|||tshirt-black|1
***END ITEM***
harddrive
1
100.00
***END ITEM***
```

The example above illustrates an autoship order with two items:

- A product with the product ID *tshirt* for which *2* have been ordered at a price of *10.00*. The first attribute for the product is named *Size* and has a value of *Large*. The second attribute for the product is named *Color*, has a value of *Black*, and decreases the product with a product ID of *tshirt-black* by a quantity of *1*. The end of the item information is signaled with an *****END ITEM***** entry.
- The second product's information begins on the line immediately following the previous item's ending *****END ITEM***** entry. The second product has a product ID of *harddrive* for which *1* has been ordered at a price of *100.00*. The product does not have any attributes. The end of the item information is signaled with an *****END ITEM***** entry without a following carriage return.



Ordered item(s) area of the add autoship order form

Shipping Address

The shipping address fields are used as the shipping address for the generated order (the billing address fields for the generated order come from the customer's billing address information in the **Customer** table). Explanations for each form field and its corresponding field in the **AutoShipOrderQueue** table are detailed below:

- **First Name** corresponds to the **FirstName** field in the table.
- **Last Name** corresponds to the **LastName** field in the table.
- **Company** corresponds to the field of the same name in the table.
- **Address** and the second unlabeled address field correspond to the **Address1** and **Address2** fields in the table, respectively.
- **City** corresponds to the field of the same name in the table.
- **State/Province** corresponds to the **State** field in the table.
- **Postal Code** corresponds to the **Postal** field in the table.
- **Country** corresponds to the field of the same name in the table.
- **Email Address** corresponds to the **Email** field in the table.
- **Phone** corresponds to the field of the same name in the table.

- **Fax** corresponds to the field of the same name in the table.

Shipping Address

* denotes an empty required field.

* **First Name**

* **Last Name**

Company

* **Address**

* **City**

* **State/Province**

* **Postal Code**

Country

Email Address

* **Phone**

Fax

Shipping address area of the add autoship order form

Payment Information

The payment information fields are used to generate the appropriate payment information for the generated order. Note that only checks and credit cards can be used as a payment method because each is the only payment method that make sense for a recurring order. If a customer changes the payment method they would like to use, then it is possible to edit the autoship order. More details about editing an autoship order can be found in the [Store Administration](#) > [Autoship Administration](#) > [Edit an Autoship Order](#) section.

Explanations for each form field and its corresponding field in the **AutoShipOrderQueue** table are detailed below:

- **Account Number** corresponds to the **Account** field in the table. If this field has a value, then the *Check* radio button should be selected.
- **Routing Number** corresponds to the **Routing** field in the table. If this field has a value, then the *Check* radio button should be selected.
- **Check Number** corresponds to the **CheckNumber** field in the table. If this field has a value, then the *Check* radio button should be selected.
- **Credit Card Type** corresponds to the **PaymentType** field in the table. If this field has a value, then the *Credit Card* radio button should be selected.
- **Credit Card Number** corresponds to the **Account** field in the table. If this field has a value, then the *Credit Card*

radio button should be selected.

- **Expiration Date** corresponds to the **Expiration** field in the table. The value chosen from the *Month* and *Year* select lists are concatenated together so that the value stores in the table is in *MMYYYY* format. If these select lists have values selected, then the *Credit Card* radio button should be selected.

Click the **Create Autoship Order** button to complete the autoship order and return to the *Autoship Administration* menu.

Important: Autoship orders must be structured properly in order to successfully generate orders. Carefully check all of the form fields and make sure that each is complete and has information in the proper format. This is especially true of the ordered items field.

The screenshot shows a form titled "Payment Information" with two radio button options: "Check" and "Credit Card". The "Check" option is currently selected. Under "Check", there are three text input fields: "Account Number", "Routing Number", and "Check Number". Under "Credit Card", there is a dropdown menu for "Credit Card Type", a text input field for "Credit Card Number", and two dropdown menus for "Expiration Date" labeled "Month" and "Year". At the bottom of the form is a button labeled "Create Autoship Order".

Payment information area of the add autoship order form

Find Autoship Order(s)

A search form is provided to search for existing autoship orders. If more than one match is found, then a list of all matching autoship orders will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching order is displayed in an edit autoship order form. The form fields and process for editing an autoship order are detailed in the [Store Administration](#) > [Autoship Administration](#) > [Edit an Autoship Order](#) section.

The fields that can be searched are listed below along with the operator used in the search:

- **CustomerID** - Equals (eq)
- **Active** - Equals (eq)
- **DateOrdered** - Equals (eq)
- **NextOrderDate** - Equals (eq)
- **LastName** - Contains (cn)
- **City** - Contains (cn)
- **State** - Equals (eq)
- **Country** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.

The screenshot shows a search form with the following fields and controls:

- Customer ID**: A text input field.
- Find Autoship Order(s)**: A button to execute the search.
- Active**: Radio buttons for Yes and No.
- Date Ordered**: Three dropdown menus for Month, Day, and Year.
- Next Order**: Three dropdown menus for Month, Day, and Year.
- Last Name**: A text input field.
- City**: A text input field.
- State**: A dropdown menu.
- Country**: A dropdown menu.
- Operator**: Radio buttons for And (selected) and Or.

Autoship order search form area of the autoship administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate autoship order in the listing and click the **ID** hyperlink for the autoship order. The autoship order will be loaded into the edit form. The form fields and process for editing autoship orders are detailed in the [Store Administration](#) > [Autoship Administration](#) > [Edit an Autoship Order](#) section.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.

<<Previous 1-25 of 2351 Next>>

Autoship order found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

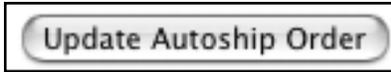
Click on the ID to edit the record.

<u>ID'</u> ▲▼	<u>Customer ID</u> ▲▼	<u>Active</u> ▲▼	<u>Next Order Date</u> ▲▼	<u>Last Name</u> ▲▼	<u>City</u> ▲▼	<u>State</u> ▲▼	<u>Country</u> ▲▼
11	15	Y	11/22/2002	Doe	Bellevue	WA	US
12	15	Y	11/22/2002	Doe	Bellevue	WA	US
13	15	Y	11/22/2002	Doe	Bellevue	WA	US
17	15	Y	11/22/2002	Doe	Bellevue	WA	US

Autoship order search results list

Edit an Autoship Order

The edit autoship order form is nearly identical to the add autoship order form. The [Store Administration > Autoship Administration > Add New Autoship Order](#) section for an explains each of the fields on the form. After making the appropriate modifications, click the Update Autoship Order button at the bottom of the form.



Update autoship order button on the edit autoship order form

Summary Information

The top of the edit form provides summary information about the autoship order. Explanations for each summary component and its corresponding field in the **AutoShipOrderQueue** table are detailed below:

- **QueueID** corresponds to the **ID** field in the table.
- **Created** corresponds to the **DateCreated** field in the table. The date that the initial order containing the autoship product(s) was ordered by the customer or the date the autoship order was manually created in *Autoship Administration*.
- **Last Modified** corresponds to the **DateModified** field in the table. The date and time that the autoship order was last modified in *Autoship Administration*.
- **CustomerID** corresponds to the field of the same name in the table. The ID for the customer (**ID** field in the **Customer** table) with which the autoship order is associated.
- **Source OrderID** corresponds to the **OrderID** field in the table. The invoice number (**ID** field in the **Invoice** table) for the initial order that contained the autoship product(s). This value will not be shown if the autoship order was manually created in *Autoship Administration*.
- **Date Ordered** corresponds to the **DateOrdered** field in the table. The date of the initial order (**DatePosted** field in the **Invoice** table) that contained the autoship product(s). This value will not be shown if the autoship order was manually created in *Autoship Administration*.
- **Date Activated** corresponds to the **DateActivated** field in the table. If the autoship order was automatically created, then this value will be the same as **Date Ordered**. If the autoship order was manually created in *Autoship Administration*, then this value will be the same as **Created**. If the autoship order was deactivated and then reactivated, then this value will be the date when the autoship order was last activated.
- **Times Executed** corresponds to the **TimesExecuted** field in the table. The number of times that the autoship order has been used to generate a new order.

Edit Autoship Order

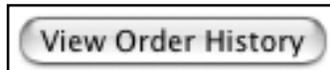
Queue ID 14
Created 10/26/2002, 7:05 AM
Last Modified 10/26/2002, 7:29 AM
CustomerID 15
Source OrderID 328
Date Ordered 10/26/2002, 7:04 AM
Date Activated 10/26/2002, 7:05 AM
Times Executed 1

[View Order History](#)
[View Audit Trail](#)

Autoship summary area of the edit autoship order form

Order History

The order history for an autoship order can be viewed by clicking the **View Order History** button located in the autoship summary area.



Order history button in the autoship summary area of the edit autoship order form

A new page will be loaded that details the date/time and invoice number for each order that has been generated from the autoship order.

Autoship Order History

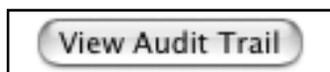
10/26/2002, 7:29 AM
Generated invoice number 332

[Back to Autoship Order Detail](#)

Order history for an autoship order

Audit Trail

Modifications to an autoship order can be viewed by clicking the **View Audit Trail** button located in the autoship summary area.



Audit trail button in the autoship summary area of the edit autoship order form

The audit trail details the date/time of each modification to the autoship order made in *Autoship Administration* as well as the user that made the modification. The initial value and final value for each modified field are also detailed for each modification.

Because autoship orders involve the automatic billing of customer checking accounts or credit cards, this feature is provided as a way to investigate any unwarranted changes that were made to an autoship order.

Autoship Order Audit Trail

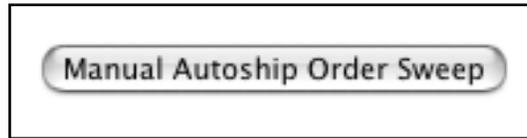
10/26/2002, 7:21 AM
Modified by Store Administrator.
Order Action was: Process
Order Action now: Pending
Next Order Date was: 2002-11-08
Next Order Date now: 2002-10-08
TOTAL MODIFICATIONS: 2

[Back to Autoship Order Detail](#)

Audit trail for an autoship order

Autoship Order Sweep

Orders can be generated from the **AutoShipOrderQueue** table by clicking the **Manual Autoship Order Sweep** button on the *Autoship Administration* menu.



Manual autoship order sweep button on the autoship administration menu

All active autoship orders that have a **NextOrderDate** field value with the current date or any date in the past will be used to generate new orders. A message will be returned that indicates how many autoship orders were generated, if any, as well as the number of orders successfully generated and the number of orders that failed. If an order fails to generate, then an email will be sent to the email address set as the `HS_AutoShipOrderFailEmail` preference variable in *Store Configuration*.

Automatic Autoship Order Generation

Autoship orders can be automatically generated by scheduling an event in Lasso Administration or programmatically scheduled with the **Event_Schedule** tag.

Important: Access to event scheduling in Lasso Administration or access to the **Event_Schedule** tag may not be possible in a shared hosting environment. Check with the hosting provider about their policies. It may be necessary to send the appropriate parameters to the webmaster so that they can schedule the event.

Scheduling Events with Lasso Administration

Events are scheduled on the **Monitor > Events > Schedule Event** tab in Lasso Administration. The appropriate values for each form fields are detailed below:

- **Event URL** - `http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=adminstore/autoshipsweep.lasso`, where `www.yourserver.here` is replaced with the appropriate server name, domain name, and domain extension.
- **Start Date/Time** - The date and time at which the event should be executed for the first time. The date and time should be specified in the format MM/DD/YYYY HH:MM:SS.
- **End Date/Time** - The date and time at which the event should stop executing. The date and time should be specified in the format MM/DD/YYYY HH:MM:SS. This field can be left blank unless for some reason the event should not continue indefinitely.
- **Repeat Delay (in Minutes)** - The frequency at which the event should be repeated. For example, if a sweep for autoship orders should occur daily, then `10080` (7 days times 24 hours/day times 60 minutes/hour) should be entered.

Note that while it is possible to generate autoship orders less frequently than daily, this could impact the date at which a customer expects to receive shipment. For example, if a customer signs up to receive a product every 30 days, then, depending upon the date the order was placed, the auto shipped order could be shipped a week later than the customer anticipates. If this approach will be used, then it is recommended to have an autoship policy available on the Web site so that customers understand that the day frequency they specify may not be an exact representation of how often they will receive their auto shipped merchandise.

- **Continue After Restart** - Set to *Yes* if the event should continue to be executed even if the Web server is restarted. Set to *No* if the event should be deactivated with a server restart. It is probably best to set this value to *Yes* because it is often unknown if a server has been restarted, especially in a shared hosting environment.
- **Username** - The username parameter is not needed to access the **Event URL** so this field can be left blank.

- **Password** - The username parameter is not needed to access the **Event URL** so this field can be left blank.

Scheduling Events Programmatically

If access to Lasso Administration is not available (in a shared hosting environment, for example), then it is possible to programmatically schedule an event. Note that this does require that permission to use the **Event_Schedule** tag has been granted. Check with the server administrator to determine if this tag is available. Use of the tag may require the use of the username and password of a privileged user (the tag is not available to the **All Users** user). Both approaches will be detailed below.

The syntax of the **Event_Schedule** tag is:

```
[Event_Schedule: -URL='URL',
                 -Start='Start Date/Time',
                 -End='End Date/Time',
                 -Delay='Delay Minutes',
                 -Restart='True/False',
                 -Username='Username for URL',
                 -Password='Password for URL']
```

The appropriate values for each of the tag parameters are detailed below:

- **-Event** - `http://www.yourserver.here/HostedStore.LassoApp?-ResponseLassoApp=adminstore/autoshipsweep.lasso`, where `www.yourserver.here` is replaced with the appropriate server name, domain name, and domain extension.
- **-Start** - The date and time at which the event should be executed for the first time. The date and time should be specified in the format MM/DD/YYYY HH:MM:SS.
- **-End** - The date and time at which the event should stop executing. The date and time should be specified in the format MM/DD/YYYY HH:MM:SS. This field can be left blank unless for some reason the event should not continue indefinitely.
- **-Delay** - The frequency at which the event should be repeated. For example, if a sweep for autoship orders should occur daily, then `10080` (7 days times 24 hours/day times 60 minutes/hour) should be entered.

Note that while it is possible to generate autoship orders less frequently than daily, this could impact the date at which a customer expects to receive shipment. For example, if a customer signs up to receive a product every 30 days, then, depending upon the date the order was placed, the autoshipped order could be shipped a week later than the customer anticipates. If this approach will be used, then it is recommended to have an autoship policy available on the Web site so that customers understand that the day frequency they specify may not be an exact representation of how often they will receive their autoshipped merchandise.

- **-Restart** - Set to *Yes* if the event should continue to be executed even if the Web server is restarted. Set to *No* if the event should be deactivated with a server restart. It is probably best to set this value to *Yes* because it is often unknown if a server has been restarted, especially in a shared hosting environment.
- **-Username** - The username parameter is not needed to access the **Event URL** so this field can be left blank.
- **-Password** - The username parameter is not needed to access the **Event URL** so this field can be left blank.

If Lasso Administration has been configured so that the **All Users** user does not have permission to use the **Event_Schedule** tag, but a privileged user has been granted permission to use the tag, then the tag must be executed between an **Inline** container tag:

```
[Inline: -Username='Username for the Privileged User',
        -Password='Password for the Privileged User',
  [Event_Schedule: -URL='URL',
                  -Start='Start Date/Time',
                  -End='End Date/Time',
                  -Delay='Delay Minutes',
                  -Restart='True/False',
```

```
-Username='Username for URL',  
-Password='Password for URL']
```

```
[/Inline]
```

When scheduling an event programmatically, it is only necessary to execute the format file with the code once. If it is executed multiple times, then the event will be scheduled multiple times. All that does is unnecessarily consume server resources.

Customer Administration

Three different functions are available for customer records: Email Customer(s), Add New Customer, and Find Customer(s). Note that even though the **Customer** table contains all of the customer addresses (billing address and any address book entries if that functionality is enabled) all of these functions are for customer billing records only (the **EntryName** field contains a value of **Billing**).

Customer Administration Menu

Customer ID	<input type="text"/>	<input type="button" value="Find Customer(s)"/>
Last Name	<input type="text"/>	
Company	<input type="text"/>	
City	<input type="text"/>	
State	<input type="text"/>	
Country	<input type="text"/>	
Price Group	<input type="text" value="None"/>	
Operator	<input checked="" type="radio"/> And <input type="radio"/> Or	
Limit	<input type="text"/>	

Customer administration menu

Sections

- [Email Customer\(s\)](#) - Send email to individual customers, all customers, or newsletter subscribers.
- [Add New Customer](#) - Create a new customer.
- [Find Customer\(s\)](#) - Search for and edit existing customers.

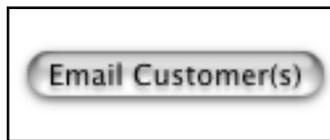
Email Customer(s)

Used to email all customers, customers subscribed to the newsletter, or an individual customer. This function allows for email-based communication with customers without having to use an email client application.

Important: None of the correspondence sent through this interface is saved in the table so if a copy of the email is wanted be sure to check the checkbox to send a copy to the sending email address or BCC an email address that is used for archiving email communication with customers.

This interface is very handy for stores that do not have access to list management software. The interface can be used to send a newsletter to all newsletter subscribers. It is not recommended to use this interface regularly to email hundreds of customers. List management software is better suited for that task. However, for small customer lists or for an occasional quick note to all customers or newsletter subscribers this feature should work fine.

Click the **Email Customer(s)** button to load a new page with a form for creating and sending the email.



Email customers form area of the customer administration menu

The fields of the email customers form are described below:

- **From** - Automatically loaded with the customer service email address set as the `HS_CustomerServiceEmail` preference variable in *Store Configuration*. The field is a text box so the sending email address can be changed if necessary.
- **To** - A select list is loaded with two options plus a single entry for each customer.
 - *All newsletter subscribers* will send the email to all customers that have elected to subscribe to the store newsletter. This corresponds to a `Y` value for the **Newsletter** field in the **Customer** table.
 - *All customers* will send the email to all customers in the **Customer** table.
 - The last set of entries is an alphabetized list of customers using fields from the **Customer** table in the following format: **LastName, FirstName (ID)** Selecting one of the entries will send the email to that customer only. Note that if the store has thousands of customers this form will probably not load because the select list will be too large. So don't do it!
- **BCC** - Enter an email address to send a blind carbon copy of the email to the specified recipient.
- **Send me a copy of the email** - Check the checkbox to send a blind carbon copy to the email address in the **To** field.
- **Subject** - Subject line of the email.
- **Message** - The body of the email. Note that only text email is supported in this interface so any HTML in the body will render as text.

Email Customer(s)

* denotes an empty required field.

* **From**

* **To**

BCC

Send me a copy of the email.

* **Subject**

* **Message**

Send Email

Email customers form

Add New Customer

Unless a store is configured otherwise, new customers will create their own customer record during the checkout process. An interface is provided for creating a customer record because there are some cases where it may be necessary to create a customer record instead of relying on a customer to do so:

- A new customer places an order via mail order, phone, or in a retail location.
- Memberships are used for reseller type customers so a customer must be created manually with the appropriate membership data.

Click the **Add New Customer** button to load a new page with a form for creating the customer.



Add new customer form area of the customer administration menu

Explanations for each form field and its corresponding field in the **Customer** table are detailed below:

- **First Name** corresponds to the **FirstName** field in the table.
- **Last Name** corresponds to the **LastName** field in the table.
- **Address** and the second unlabeled address field correspond to the **Address1** and **Address2** fields in the table, respectively.
- **City** corresponds to the field of the same name in the table.
- **State** corresponds to the field of the same name in the table.
- **Postal Code** corresponds to the **Postal** field in the table.
- **Country** corresponds to the field of the same name in the table.
- **Email** corresponds to the field of the same name in the table.
- **Phone** corresponds to the field of the same name in the table.
- **Fax** corresponds to the field of the same name in the table.
- **Password** corresponds to the field of the same name in the table.
- **Newsletter** corresponds to the field of the same name in the table. Selecting *Yes* subscribes the customer to the newsletter if one is provided for the store.
- **Public Wish List** corresponds to the **PublicWishList** field in the table. Selecting *Yes* will make the customer's wish list (if they choose to create one) searchable by the public. Selecting *No* will keep the wish list private so the customer will need to email a hyperlink to the wish list from *Customer Account Administration* if they choose to share the wish list with somebody.
- **Approved for Purchase Orders** corresponds to the **ApprovedForPO** field in the table. Selecting *Yes* will allow the customer to use a purchase order number during checkout if purchase orders have been enabled as a payment option with the `HS_AcceptPurchaseOrders` preference variable in *Store Configuration*.
- **Price Group** corresponds to the **PriceGroupID** field in the table. This field will only appear if price groups have been enabled in *Store Configuration* (the `HS_UsePriceGroups` preference variable has a value of *Y*). The select list will display a list of all price groups that are active. Select *None* if the customer is not assigned to a price group, i.e., HostedStore should use its default calculation method for determining product pricing for the customer.
- **Membership Created** corresponds to the **MembershipCreated** field in the table. Three select lists allow for the **Month**, **Day**, and **Year** of membership initiation to be selected. This field is only meaningful if memberships have been enabled with the `HS_UseMemberships` preference variable in *Store Configuration*.
- **Membership Expires** corresponds to the **MembershipExpires** field in the table. Three select lists allow for the **Month**, **Day**, and **Year** of membership termination to be selected. This field is only meaningful if memberships have been enabled with the `HS_UseMemberships` preference variable in *Store Configuration*.

Create Customer

* denotes an empty required field.

* First Name

* Last Name

Company

* Address

* City

State

* Postal Code

* Country

* Email

Phone

Fax

* Password

* Newsletter Yes No

* Public Wish List Yes No

Approved for Purchase Orders Yes No

Price Group

Membership Created

Membership Expires

Create Customer

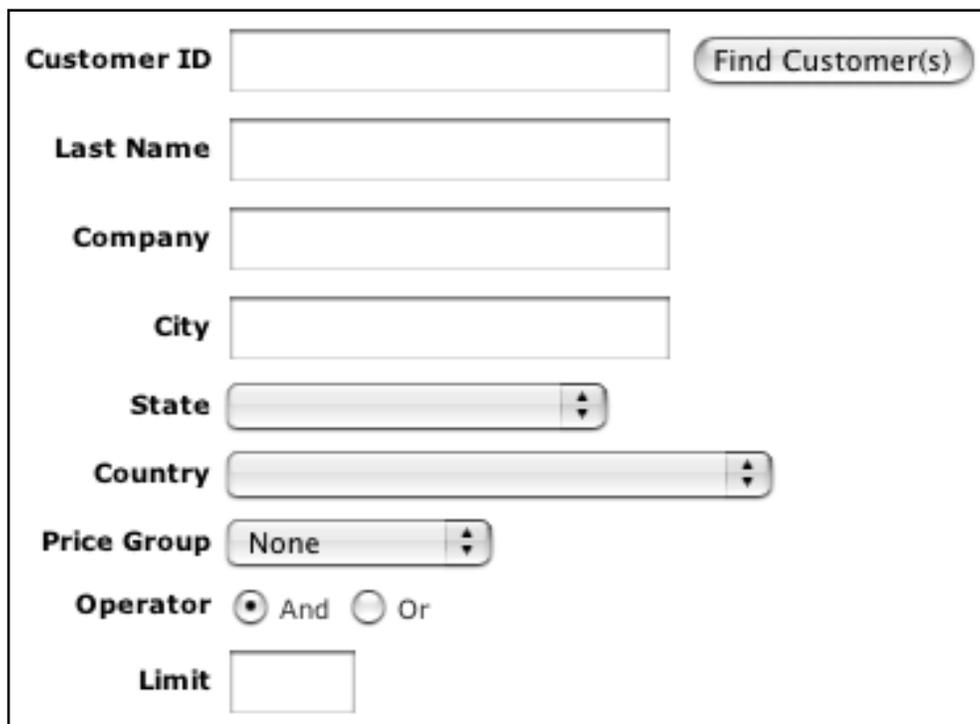
Create customer form

Find Customer(s)

A search form is provided to search for existing customers. If more than one match is found, then a list of all matching customers will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching customer is displayed in an edit customer form. The fields for editing a customer are the same as the fields available when adding a new customer.

The fields that can be searched are listed below along with the operator used in the search:

- **CustomerID** - Equals (eq)
- **LastName** - Contains (cn)
- **Company** - Contains (cn)
- **City** - Equals (eq)
- **State** - Equals (eq)
- **Country** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The screenshot shows a search form with the following fields and controls:

- Customer ID**: Text input field with a "Find Customer(s)" button to its right.
- Last Name**: Text input field.
- Company**: Text input field.
- City**: Text input field.
- State**: Dropdown menu.
- Country**: Dropdown menu.
- Price Group**: Dropdown menu with "None" selected.
- Operator**: Radio buttons for "And" (selected) and "Or".
- Limit**: Text input field.

Customer search form area of the customer administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate customer in the listing and click the **ID** hyperlink for the customer. The customer will be loaded into the edit form. The form fields and process for editing customers is similar to the process for adding a customer detailed in the [Store Administration](#) > [Customer Administration](#) > [Add New Customer](#) section.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.



Customer found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the customer ID to edit the customer record.

<u>Customer ID</u> ▲▼	<u>Last Name'</u> ▲▼	<u>Company</u> ▲▼	<u>City</u> ▲▼	<u>State</u> ▲▼	<u>Country</u> ▲▼
52	Doe	HostedStore Corporation	Bellevue	WA	US
61	Doe	HostedStore Corporation	Bellevue	WA	US
68	Doe	HostedStore Corporation	Bellevue	WA	US

Customer search results list

After all fields have been modified, click the **Update Customer** button at the bottom of the edit form to update the customer record in the table.



Update customer button on the edit customer form

Maintenance

This section of *Store Administration* handles a few basic database export/backup, database cleanup, and application related functions.

Maintenance Menu

Database Export

Table Name

File Path

Format

Database Backup

Table Name

File Path

Purge Abandoned Carts

Age (Days)

Mark Abandoned Orders

Age (Days)

Purge Account Info

Age (Days)

Status

HostedStore Application

Maintenance menu

Database Export

Select a table name, a file path, and a file format to create a file with the contents of the chosen table. This feature can be useful for exporting data for eventual import into a third party reporting or accounting application. However, it is recommended that the database client is used for making export files if a data export is a regular routine.

Database Export

Table Name

File Path

Format

Database export form area of the maintenance menu

Database Backup

Select a table name and a file path to create a backup of the chosen table. This feature is offered as a convenience for making a quick backup of database tables. However, it is strongly recommended that a better backup routine is used for regular reliable backups.

Important: This feature only works when Lasso MySQL is used as the database application.

Database Backup

Table Name

File Path

Database backup form area of the maintenance menu

Purge Abandoned Carts

Select an age (in days) and all shopping carts that were not converted to orders will be removed from the database if their age is greater than the number of days specified. This feature is added as a means to keep database sizes to a minimum. However, careful thought should be given before shopping carts are purged. Removing abandoned shopping carts removes useful data!

Purge Abandoned Carts

Age (Days)

Purge abandoned carts form area of the maintenance menu

Mark Abandoned Orders

Select an age (in days) and all orders that were not completed will be marked as abandoned if their age is greater than the number of days specified.

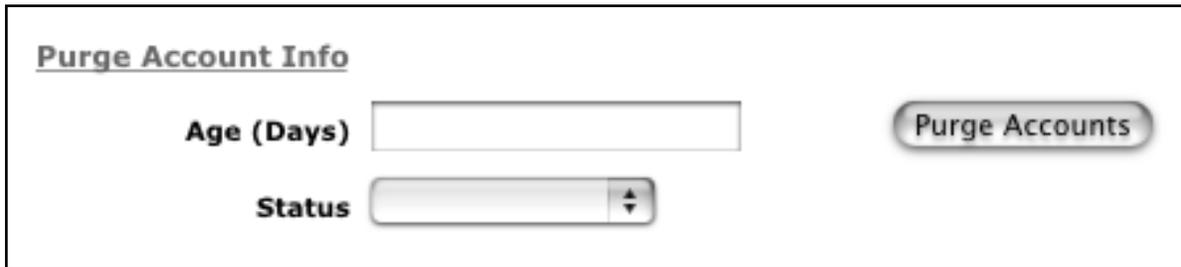
Mark Abandoned Orders

Age (Days)

Mark abandoned orders form area of the maintenance menu

Purge Account Info

Select an age (in days) and order status and all orders that meet the criteria will have the credit card and checking account information purged from the database. Although that information is stored encrypted, this feature is offered for those that have anxiety about keeping that kind of information in a live database.



The screenshot shows a form titled "Purge Account Info". It contains two input fields: "Age (Days)" which is a text box, and "Status" which is a dropdown menu. To the right of these fields is a button labeled "Purge Accounts".

Purge account info form area of the maintenance menu

Mask Account Info

Select an age (in days) and order status and all orders that meet the criteria will have the credit card and checking account information masked in the database in a manner that preserves the first characters and the last four characters. All characters in the middle will be replaced with an X. The masked information is still stored encrypted.

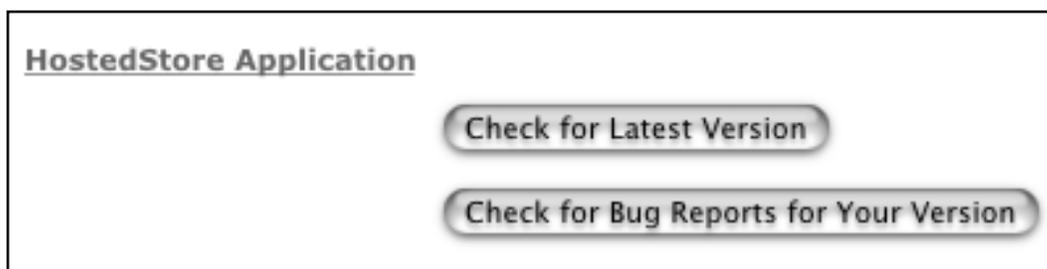


The screenshot shows a form titled "Mask Account Info". It contains two input fields: "Age (Days)" which is a text box, and "Status" which is a dropdown menu. To the right of these fields is a button labeled "Mask Accounts".

Mask account info form area of the maintenance menu

HostedStore Application

Two features check the HostedStore server for information about the currently installed version of HostedStore.



The screenshot shows a form titled "HostedStore Application". It contains two buttons: "Check for Latest Version" and "Check for Bug Reports for Your Version".

HostedStore application form area of the maintenance menu

Check for Latest Version

Compares the installed version to the latest version available. In general, one should upgrade to the latest version (it's free afterall!) to take advantage of any bug fixes or new features.

A Note about Versions

Because all future upgrades are free for purchased HostedStore licenses the standard full release and dot release version numbering has been avoided. All versions of HostedStore have a version number corresponding to the date the version was released in YYYY.MM.DD format. Details about HostedStore version numbering can be found in the [Upgrading](#) section.

Maintenance Menu

Latest version of HostedStore: 2002.07.01
Installed version of HostedStore: 2002.06.17

Download the latest version of HostedStore at <http://www.hostedstore.com/support/download/>.

Results of checking for latest version

Check for Bug Reports for Your Version

Returns a list of bugs reported for the currently installed version of HostedStore. If any of the bugs were fixed, then the version in which the fix was made available will also be reported. Each bug title will be displayed as a hyperlink to the full bug report in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Bugs Reported for Version 2002.06.17

Bug reports for your installed version of HostedStore, 2002.06.17, are listed below.
Click the title hyperlink to open a new browser window with the full bug report and any added comments.

Reported	Status	Released	Title/Description
6/30/2002	Closed	2002.07.01	Prevent caching of cart The modify cart item content page and the cart reload after modifying an item cache.
6/30/2002	Closed	2002.07.01	Radio button attributes not displayed when modifying item The modify item content page displays radio button attributes as select lists.
6/28/2002	Closed	2002.07.01	Attribute radio button option does not hold The display attribute as radio buttons option does not maintain its checked status when a product is loaded into the update product form.
6/19/2002	Closed	2002.07.01	&#34;AND&#34; shows up for City when adding new affiliate Adding a new affiliate in store admin places "AND" in the City field.
6/19/2002	Open		Error when searching for featured products in product admin Error -9951 when searching for featured products in product admin.
6/19/2002	Closed	2002.07.01	Error when generating store statistics for new store Error -9951 (Attempt to divide by zero) occurs when attempting to generate a statistics report for new stores (a store without any activity).
6/18/2002	Closed	2002.07.01	Secure path to store settings do not hold for config page The secure path to store setting resets to "Use Shared Certificate Path" each time the store configuration page is loaded.

Results of checking for bug reports

Modules

If any modules have been installed, then access to the module administration functions may be made available through this menu. Consult the documentation provided by the module developer for installation and configuration instructions.

Order Administration

Three different functions are available for order records: Add New Order, Find Order(s), and Batch Settlement. These three functions can be used in conjunction to create new orders, update the status of orders, and reconcile the orders against an external sales summary.

The screenshot shows the 'Order Administration Menu' with the following fields and buttons:

- Customer ID** **Add New Order**
- Customer ID** **Add New Return**
- Invoice Number** **Find Order(s)**
- Customer ID**
- Last Name**
- Begin Date**
- End Date**
- Status**
- Items**
- Total**
- Affiliate ID**
- Limit**
- Batch Settlement**
- Export for Stone Edge Order Manager**
- File Path** **Export Order(s)**
- Begin at Invoice Number**

Order administration menu

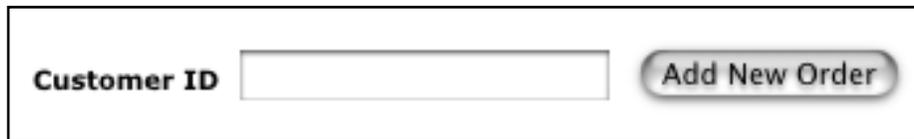
Sections

- [Add New Order](#) - Create a new order received via mail order or in a retail location.
- [Add New Return](#) - Create a new order that adds items back into inventory.
- [Find Order\(s\)](#) - Search for existing orders.
- [Edit an Order](#) - Modify order status.
- [Batch Settlement](#) - Reconcile orders.
- [Export for Stone Edge Order Manager](#) - Create a text file export of orders.

Add New Order

Orders can be entered manually to facilitate the use of HostedStore for mail order sales, phone sales, retail sales, and any other sales process where the customer will not be placing the order via the online store.

Enter the ID for the customer that is placing the order and click the **Add New Order** button. If the order is for a new order, then the customer will need to be created first in the *Customer Administration* area.

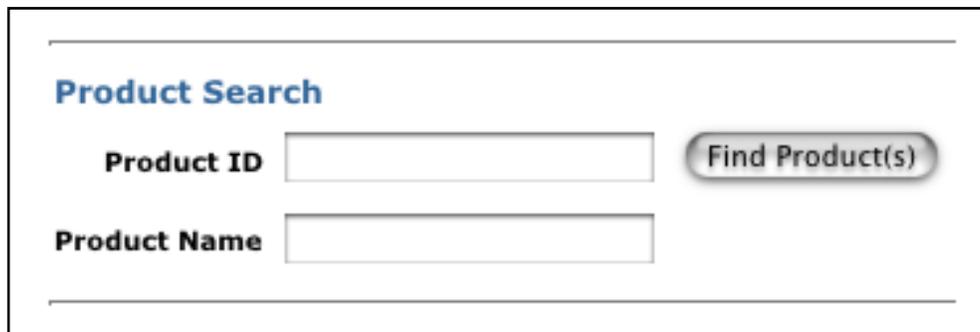


The image shows a rectangular form area with a thin black border. On the left side, the text "Customer ID" is displayed in a bold, black font. To its right is a white rectangular input field. Further to the right is a rounded rectangular button with a grey gradient and the text "Add New Order" in a bold, black font.

Add order form area of the order administration menu

Adding Products to the Order

A simple search form is provided to find the appropriate product. Only the **ProductID** and **ProductName** fields can be searched because it is assumed that the store administrator will be familiar enough with their own product line to find the product with just those two fields. If more search fields are needed, simply open a new browser window and search for the product through the storefront like an online customer would or search for the product using the *Product Administration* search form.



The image shows a rectangular form area with a thin black border. At the top left, the text "Product Search" is displayed in a bold, blue font. Below this, there are two rows of input fields. The first row has the text "Product ID" in bold black font to the left of a white input field. The second row has the text "Product Name" in bold black font to the left of a white input field. To the right of the "Product ID" input field is a rounded rectangular button with a grey gradient and the text "Find Product(s)" in a bold, black font.

Product search form

The products that meet the search criteria will be listed below the search form. Clicking the **Product ID** hyperlink will open a new window with the product detail content page from the storefront. Use this hyperlink when there may be uncertainty about a product, but do not use the product detail content page to enter products into the shopping cart. Clicking the **Product Name** hyperlink will add the item to the shopping cart or will load the configuration form for the product if it has attributes.

Product Search Results

<u>Product ID</u>	<u>Product Name</u>	<u>Price</u>	
17221600001	Holster	\$5.39	Add to Cart
17221075014	Holster	\$5.50	Add to Cart
17221075015	Holster	\$5.39	Add to Cart
17221075008	Holster	\$5.39	Add to Cart
17221075007	Holster	\$5.39	Add to Cart
17221075005	Holster	\$5.39	Add to Cart
17221075003	Holster	\$5.39	Add to Cart
17221075004	Holster	\$5.39	Add to Cart
NTN1987	Holster	\$10.10	Add to Cart
SYN8688	Holster	\$7.63	Add to Cart
SYN8454	Holster	\$7.63	Add to Cart
SYN7119	Holster	\$6.67	Add to Cart
SHN6971	Holster	\$6.75	Add to Cart
MCGA0000501	Holster	\$10.10	Add to Cart
MCGY0000801	Holster	\$10.49	Add to Cart
NTN8657	Holster for Older Model	\$6.75	Add to Cart

Product search results list

The shopping cart is displayed at the bottom of the page. Item quantities can be updated in the shopping cart by changing the quantity and clicking the **Update Cart** button. Setting the quantity of an item to *0* will remove the item from the shopping cart. Click the **Cancel Order** button to cancel the order and return to the *Order Administration* menu. Click the **Checkout** button to enter shipping and payment information.

Shopping Cart

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Case w/Swivel & Cutouts	<input type="text" value="1"/>	\$10.13	\$10.13
Holster	<input type="text" value="1"/>	\$10.10	\$10.10

Shopping cart

Checkout

A summary of the order appears at the top of the page. This view is nearly identical with the shopping cart view a customer sees on the default shopping cart detail content page (*/hostedstore/storefront/html/cart_display.lasso*).

Order Summary					
<u>Product Name</u>		<u>Quantity</u>	<u>Price</u>	<u>Total</u>	
Case w/Swivel & Cutouts		1	1	\$10.13	
Holster		1	1	\$10.10	
Total Without Tax			Total with Tax		
	<u>Quantity</u>	<u>Total</u>		<u>Quantity</u>	<u>Total</u>
SubTotal	2	\$20.23	SubTotal	2	\$20.23
Shipping		\$5.00	Shipping		\$5.00
Total		\$25.23	Tax		\$1.76
			Total		\$26.99

Order summary

The shipping information form is automatically populated with the customer's billing address information. Any of the information can be modified. Shipping and tax charges will be recalculated to reflect any changes caused by a change in shipping information.

Shipping Address

* denotes an empty required field.

First Name

Last Name

Company

Address

City

State/Province

Postal Code

Country

Email Address

Phone

Fax

Shipping information

Comments can be added. Note that this is the same as the comments section for online customers. It is not the same as the administrator comments field available on the edit order page in *Order Administration*.

If a promotional code is to be applied to the order, a text box is available for it. Like online orders, the promotion will be checked to make sure it is valid and can be applied to the order.

Comments

Comments

Order Discount

Promotion/Coupon Code

Comments and promotion code

Several payment methods are available. These may not match the payment methods available to online customers because more methods are necessary to support mail order, retail sales, etc.

Important: Payment processing does not occur in real-time even if the storefront is setup for real-time processing. All payment processing for manually created orders must be performed manually via a credit card swipe terminal, through the Web interface provided by the payment gateway, or via another method.

Select the source of the order and click the **Review Order** button.

Payment Information

Cash

Purchase Order

PO Number

Check

MICR

Transit Routing Number

Check Number

Credit Card

Credit Card Type

Credit Card Number

Expiration Date

Order Information

 * **Order Source**

Payment and order source

Order Review

Verify that the shipping information and order total information is correct at the top of the form. Click the **Make Changes** button if any of the information is incorrect.

Review Order

Shipping Address

John Doe
 HostedStore Corporation
 PO Box 6743
 Bellevue, WA 98008-0743
 US
 johndoe@hostedstore.com
 425.644.5983

Order Total

SubTotal	\$20.23
Shipping	\$5.00
Tax	\$1.76
Discount	(\$0.00)
TOTAL	\$26.99

Payment Information

Method Cash

Review order

Select what the status of the order should be when it is posted and who (if anyone) should receive email notification of the placed order.

Click the **Finalize Order** button to complete the order and return to the *Order Administration* menu. If an invoice for the order is needed (for a retail sale, for example) search for the order and click the **Mailing Invoice** hyperlink on the edit order page. Details about working with placed orders can be found in the [Store Administration](#) > [Order Administration](#) > [Edit an Order](#) section.

Important: There is very little error checking done for order creation in this part of the application in comparison to the order checkout for customers. Be very careful about hitting the browser **Back** button and resubmitting forms because multiple orders may be created.

Order Status

Email receipt to customer.

Email receipt to sales.
(sales@hostedstore.com)

Set status and finalize order

Add New Return

The process of creating a return is the same as the process used to manually create an order. The main difference is that all values are reverse entries so monetary totals are negative and returned items are added back to the available inventory. Like orders, returns must be processed and closed out (shipped) in *Order Administration*.

Enter the ID for the customer that is placing the return and click the **Add New Return** button.

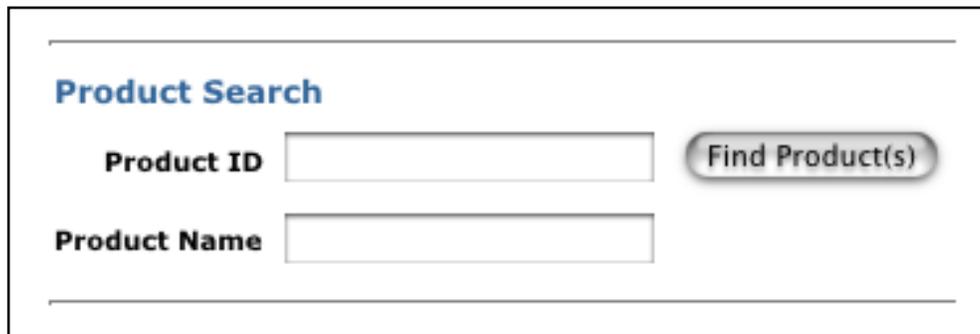


A screenshot of a web form for adding a new return. It features a text input field labeled "Customer ID" and a button labeled "Add New Return".

Add return form area of the order administration menu

Adding Products to the Return

A simple search form is provided to find the appropriate product. Only the **ProductID** and **ProductName** fields can be searched because it is assumed that the store administrator will be familiar enough with their own product line to find the product with just those two fields. If more search fields are needed, simply open a new browser window and search for the product through the storefront like an online customer would or search for the product using the *Product Administration* search form.



A screenshot of a product search form. It has a title "Product Search" in blue. Below the title are two text input fields: "Product ID" and "Product Name". To the right of the "Product ID" field is a button labeled "Find Product(s)".

Product search form

The products that meet the search criteria will be listed below the search form. Clicking the **Product ID** hyperlink will open a new window with the product detail content page from the storefront. Use this hyperlink when there may be uncertainty about a product, but do not use the product detail content page to enter products into the shopping cart. Clicking the **Product Name** hyperlink will add the item to the shopping cart or will load the configuration form for the product if it has attributes.

Product Search Results

<u>Product ID</u>	<u>Product Name</u>	<u>Price</u>
17221600001	Holster	\$5.39 Add to Cart
17221075014	Holster	\$5.50 Add to Cart
17221075015	Holster	\$5.39 Add to Cart
17221075008	Holster	\$5.39 Add to Cart
17221075007	Holster	\$5.39 Add to Cart
17221075005	Holster	\$5.39 Add to Cart
17221075003	Holster	\$5.39 Add to Cart
17221075004	Holster	\$5.39 Add to Cart
NTN1987	Holster	\$10.10 Add to Cart
SYN8688	Holster	\$7.63 Add to Cart
SYN8454	Holster	\$7.63 Add to Cart
SYN7119	Holster	\$6.67 Add to Cart
SHN6971	Holster	\$6.75 Add to Cart
MCGA0000501	Holster	\$10.10 Add to Cart
MCGY0000801	Holster	\$10.49 Add to Cart
NTN8657	Holster for Older Model	\$6.75 Add to Cart

Product search results list

The shopping cart is displayed at the bottom of the page. Item quantities can be updated in the cart by changing the quantity and clicking the **Update Cart** button. Setting the quantity of an item to 0 will remove the item from the cart. Click the **Cancel Return** button to cancel the return and to return to the *Order Administration* menu. Click the **Checkout** button to enter shipping and payment information.

Shopping Cart

<u>Product Name</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
Case w/Swivel & Cutouts	<input type="text" value="1"/>	(\$10.13)	(\$10.13)
Holster	<input type="text" value="1"/>	(\$10.10)	(\$10.10)

Shopping cart

Checkout

A summary of the return appears at the top of the page. This view is nearly identical with the shopping cart view a customer sees on the default shopping cart detail content page (*/hostedstore/storefront/html/cart_display.lasso*) except that extended prices are negative monetary values.

Return Summary					
<u>Product Name</u>		<u>Quantity</u>	<u>Price</u>	<u>Total</u>	
Case w/Swivel & Cutouts		1	(\$10.13)	(\$10.13)	
Holster		1	(\$10.10)	(\$10.10)	
Total Without Tax			Total with Tax		
	<u>Quantity</u>	<u>Total</u>		<u>Quantity</u>	<u>Total</u>
SubTotal	2	(\$20.23)	SubTotal	2	(\$20.23)
Shipping		(\$5.00)	Shipping		(\$5.00)
Total		(\$25.23)	Tax		(\$1.76)
			Total		(\$16.99)

Return summary

The shipping information form is automatically populated with the customer's billing address information. Any of the information can be modified. Shipping and tax charges will be recalculated to reflect any changes caused by a change in shipping information.

Shipping Address

* denotes an empty required field.

First Name	<input type="text" value="John"/>
Last Name	<input type="text" value="Doe"/>
Company	<input type="text" value="HostedStore Corporation"/>
Address	<input type="text" value="PO Box 6743"/> <input type="text"/>
City	<input type="text" value="Bellevue"/>
State/Province	<input type="text" value="Washington"/>
Postal Code	<input type="text" value="98008-0743"/>
Country	<input type="text" value="US"/>
Email Address	<input type="text" value="johndoe@hostedstore.com"/>
Phone	<input type="text" value="425.644.5983"/>
Fax	<input type="text"/>

Shipping information

Comments can be added. Note that this is the same as the comments section for online customers. It is not the same as the administrator comments field available on the edit order page in *Order Administration*. Because the customer will also see the comments, this field is a good place to comment on the original invoice number that is associated with the return.

If a promotional code is to be applied to the order, a text box is available for it. Like online orders, the promotion will be checked to make sure it is valid and can be applied to the order.

Comments

Comments

Order Discount

Promotion/Coupon Code

Comments and promotion code

Several payment methods are available. These may not match the payment methods available to online customers because more methods are necessary to support mail order, retail sales, etc.

Important: Payment processing does not occur in real-time even if the storefront is setup for real-time processing. All payment processing for manually created returns must be performed manually via a credit card swipe terminal, through the Web interface provided by the payment gateway, or via another method.

Select the source of the order and click the **Review Return** button.

Payment Information

Cash

Purchase Order

PO Number

Check

MICR

Transit Routing Number

Check Number

Credit Card

Credit Card Type

Credit Card Number

Expiration Date

Return Information

*** Order Source**

Payment and order source

Return Review

Verify that the shipping information and return total information is correct at the top of the form. Click the **Make Changes** button if any of the information is incorrect.

Review Return

Shipping Address

John Doe
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US
johndoe@hostedstore.com
425.644.5983

Order Total

SubTotal	(\$20.23)
Shipping	(\$5.00)
Tax	(\$1.76)
Discount	(\$0.00)
TOTAL	(\$26.99)

Payment Information

Method Cash

Review return

Select what the status of the order should be when it is posted and who (if anyone) should receive email notification of the placed order.

Click the **Finalize Return** button to complete the return and to return to the *Order Administration* menu. If an invoice for the return is needed (for a retail return, for example) search for the return and click the **Mailing Invoice** hyperlink on the edit order page. More details on working with placed returns can be found in the [Store Administration](#) > [Order Administration](#) > [Edit an Order](#) section.

Important: There is very little error checking done for return creation in this part of the application in comparison to the order checkout for customers. Be very careful about hitting the browser **Back** button and resubmitting forms because multiple returns may be created.

Order Status

Email receipt to customer.

Email receipt to sales.
(sales@hostedstore.com)

Set status and finalize return

Find Order(s)

A search form is provided to search for existing orders. If more than one match is found, then a list of all matching orders will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching order is displayed in an edit order form. The fields and process for editing orders are detailed in the [Store Administration](#) > [Order Administration](#) > [Edit an Order](#) section.

The fields that can be searched are listed below along with the operator used in the search:

- **OrderID** - Equals (eq)
- **CustomerID** - Equals (eq)
- **LastName** - Contains (cn)
- **DateCreated** - Greater than or equal to (gte)
- **DateCreated** - Less than or equal to (lte)
- **Status** - Equals (eq)
- **ItemCount** - Equals (eq)
- **Total** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.

The screenshot shows a search form with the following fields and controls:

- Invoice Number**: Text input field with a **Find Order(s)** button to its right.
- Customer ID**: Text input field.
- Last Name**: Text input field.
- Begin Date**: Three date pickers for Month, Day, and Year.
- End Date**: Three date pickers for Month, Day, and Year.
- Status**: Dropdown menu.
- Items**: Text input field.
- Total**: Text input field.
- Affiliate ID**: Text input field.
- Limit**: Text input field.

Order search form area of the order administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate order in the listing and click the **Invoice** hyperlink for the order. The order will be loaded into the edit form. The form fields and process for editing orders is detailed in the [Store Administration](#) > [Order Administration](#) > [Edit an Order](#) section.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.

<<Previous 1-25 of 2351 Next>>

Order found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the invoice number to edit.

<u>Invoice'</u> ▲▼	<u>Customer</u> ▲▼	<u>Last Name</u> ▲▼	<u>Date/Time Created</u> ▲▼	<u>Status</u> ▲▼	<u>Items</u> ▲▼	<u>Total</u> ▲▼
333	15	Doe	10/26/2002, 8:14 AM	Approved	11	\$552.35
334	15	Doe	10/26/2002, 8:43 AM	Approved	1	\$44.99
336	15	Doe	10/26/2002, 3:04 PM	Approved	1	\$44.99
337	15	Doe	10/26/2002, 4:24 PM	Approved	1	\$44.99
368	15	Doe	5/28/2003, 3:25 AM	Approved	1	(\$884.99)

Order search results list

Edit an Order

A lot of thought and planning went into this section to add features that would be useful while keeping the interface intuitive. Hopefully that time was well spent and the features will exceed the needs and expectations of store administrators. One of the more exciting features (at least to us) is the ability to split multiple items into individual items and apply a status to each item in an order to facilitate line item shipping, back ordering of individual products, canceling of individual products, and more. Each section of the edit order page is described below.

This content page is split into two sections. The first section, *Edit Order Form*, explains the purpose of each of the edit form components. *Processing Orders*, the second section, explains how to use the edit order form to process orders and how the functionality of each form component changes throughout the order approval process.

Edit Order Form

Each component of the edit order form is explained below. Note that what is displayed in each component and the actions available will change as the order progresses through the order processing timeline. That in mind, each form component is discussed in a general manner. Details about how each form component may change are detailed in the *Processing Orders* section below.

Order Summary

The top of the edit order page displays a summary of the global order information including order date, invoice number, order source, order status, a list of dates that summarizes the status progress of the order, payment information, billing address, shipping address, and customer comments (if any). The list of dates is displayed in reverse chronological order (usually--there are some scenarios in which this may not be true, but it doesn't affect usability because it is just an aesthetic issue) and includes all dates for which an action was applied globally to an order.

<u>Date/Time Created</u> 7/11/2002, 3:15 PM	<u>Billing Address</u> John Doe HostedStore Corporation PO Box 6743 Bellevue, WA 98008-0743 US johndoe@hostedstore.com 425.644.5983
<u>Invoice</u> 257	
<u>Source</u> Retail Store	
<u>Status</u> Approved Approved 7/11/2002, 3:15 PM Posted 7/11/2002, 3:15 PM Created 7/11/2002, 3:15 PM	<u>Shipping Address</u> John Doe HostedStore Corporation PO Box 6743 Bellevue, WA 98008-0743 US johndoe@hostedstore.com 425.644.5983
<u>Payment Information</u> Method Cash	

Ordered Items

The middle of the page displays a list of all the items in the order. The status of each item can be modified individually if necessary. Note that setting the status for a single item will disable the ability to modify the global order status (see the *Global Order Modifications* section below).

- **Status** - This column switches between allowing status modification and strictly displaying status depending upon the status of the item and the global status of the order. The status of the order can be set to one of the following:
 - *Back Ordered* - Used for items for which shipping has been delayed because the item is not in stock.
 - *Shipped* - Used for items that have been shipped.
 - *Canceled* - Used for items that have been canceled from the order. Note that canceling an item from the order does not adjust the subtotal or total of the order. A manual adjustment of the order subtotal will need to be made if the order total is to properly reflect the correct value (see the *Order Totals Adjustments* section below).
- **Split Item** - If more than one of an item was ordered, then a select list will be displayed that contains a list of numbers from one up to one less than the quantity ordered. Selecting a number in the list will split the item into two separate line items with identical properties. The quantities and total price will be appropriately adjusted for each line item.

For example, if five of an item was ordered, then selecting **4** from the list will split the item into two items. One item will have a quantity of **1** and the second item will have a quantity of **4**. This feature allows for a different status to be applied to multiple quantity items. For example, if five of an item is ordered but only four are in stock, then the item can be split so the four items can be shipped and the other item can be back ordered.

- **Vendor/Tracking** - This column switches between allowing shipping vendor and tracking number modification and strictly displaying that information depending upon the status of the item and the global status of the order. If a tracking number is entered in the text box, then a shipping vendor must be selected.

Multiple shipping vendors are available for selection. If one of the vendors is selected, then the tracking number will be displayed as a hyperlink to the shipping tracking page on the vendor's Web site on any of the default content pages where a tracking number is displayed. If a vendor other than those listed is needed, be sure to make a feature request for the vendor to be added. In the meantime, *Other* can be used and the appropriate code modifications can be made to content pages to display the tracking number as a hyperlink if that functionality is wanted.

- **Item** - Displays the product name and selected attributes (if applicable). Clicking the product name hyperlink will open a new browser window with the product detail content page.
- **Qty** - The quantity of the item ordered. If this value is greater than one, then the item can be split into separate items.

The quantity of an item cannot be adjusted. If the quantity of an item needs to be reduced, then the item should be canceled. If the quantity is greater than one and the quantity needs to be reduced to a non-zero value, then the item should be split and the appropriate quantity can be canceled.

- **Price** - The price of the item. Note that this is the configured price of the item if the product has price adjusting attributes.
- **SubTotal** - Total price of the ordered items without tax and shipping costs applied.

Status	Vendor/Tracking	Item	Qty	Price	SubTotal
<input type="text"/>	<input type="text"/>	Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
<input type="text"/>	<input type="text"/>	Holster (NTN1987)	2	\$10.10	\$20.20
Split Item <input type="text"/>	<input type="text"/>				

Ordered items

Administrator Comments and Printable Invoices

The bottom of the page contains an area to add comments about the order. The comments are not viewable by the customer. Additionally, the comments are encrypted in the table so, for example, product serial numbers or information about the payment method used for a purchase order can be stored in this field safely.

Administrator Comments

[Pick List](#) [Packing Invoice](#) [Mailing Invoice](#) [Email Customer](#)

Administrator comments and printable invoices

The *Pick List*, *Packing Invoice*, and *Mailing Invoice* hyperlinks open a new browser window with the appropriate content page. Use these hyperlinks if paperwork will be generated directly from HostedStore. All of these content pages can be modified if necessary. Details about editing the pages can be found in the [Modifying HostedStore > Content Pages > Store Administration](#) section. The *Email Customer* hyperlink will open a new browser window with a form for emailing the customer. A copy of the email is not retained in the table so be sure to BCC a copy if needed.

```
2003-06-09 05:07:49

Invoice: 287

1 - BP2 "All'armi, pensieri"

1 - NTN8011 Battery Belt Clip w/Insert

1 - IBR-1 In-Building Repeater

TOTAL ITEMS: 3
```

Pick list

Invoice: 287

Shipped to: Jane Doe
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
US

- 1 - BP2 "All'armi, pensieri"
- 1 - NTN8011 Battery Belt Clip w/Insert
- 1 - IBR-1 In-Building Repeater

Items: 3

SubTotal: \$372.14
Shipping: \$7.50
Tax: \$32.38
Discount: \$0.00

Total: \$412.02

Thank you for your order. Please contact us with any questions.

HostedStore Development Store Sales
HostedStore Corporation
PO Box 6743
Bellevue, WA 98008-0743
USA
425.644.5983
bkm@speakeasy.org

Packing invoice

HostedStore Corporation

PO Box 6743
 Bellevue, WA 98008-0743
 USA
 425.644.5983
 bkm@speakeasy.org

Invoice/Receipt**Invoice #287**

BILL TO:
 Jane Doe
 HostedStore Corporation
 PO Box 6743
 Bellevue, MA 98008-0743
 US

SHIP TO:
 Jane Doe
 HostedStore Corporation
 PO Box 6743
 Bellevue, WA 98008-0743
 US

<i>Date Ordered</i>	<i>Payment Method</i>	<i>Order Status</i>	<i>Ship Date</i>	<i>Shipping Vendor</i>	<i>Tracking Number</i>
7/23/2002	Check by Mail/Phone	Shipped	7/27/2002		

<i>Quantity</i>	<i>Product ID</i>	<i>Product Description</i>	<i>Price</i>	<i>SubTotal</i>
1	BP2	"All'armi, pensieri"	\$30.00	\$30.00
1	NTN8011	Battery Belt Clip w/Insert	\$12.15	\$12.15
1	IBR-1	In-Building Repeater	\$329.99	\$329.99

SubTotal \$372.14
 Shipping \$7.50
 Tax \$32.38
 Discount \$0.00
TOTAL \$412.02

Thanks for your order!

Mailing invoice

Email Customer

* denotes an empty required field.

From

To

BCC

Send me a copy of the email.

Subject

* **Message**

Archive this email.

Email customer form

Order Totals Adjustments

The right column contains four different text boxes that allow components of the order total to be adjusted. All of the adjustments are applied to the order total. An individual adjustment is provided for each order total component to aid as a reminder of why an adjustment was added. Adding a detailed explanation in the administrator notes area may be helpful to aid in recollecting at a later date why an adjustment was made.

SubTotal	\$30.33
Shipping	\$7.50
Tax	\$2.64
Discount	(\$0.00)
SubTotal Adjustment	<input type="text"/>
Shipping Adjustment	<input type="text"/>
Tax Adjustment	<input type="text"/>
Discount Adjustment	<input type="text"/>
Total	\$40.47

Order totals adjustments

Global Order Modifications

If the status of individual ordered items has not been adjusted, then it is possible to make global order modifications that will apply the status change to all ordered items. The status of the order can be set to one of the following:

- **Pending** - Used for orders that have been placed, but availability of funds has not been confirmed. An order will be assigned a pending status in the following scenarios:
 - The order was purchased with a purchase order.
 - The order was purchased when offline check or credit card processing is used by the store.
 - The order was purchased with the check by phone/mail payment method.
 - The order was manually created in *Order Administration* and assigned a pending status.
- **Approved** - Used for orders that have been placed and the availability of funds has been confirmed. An order will be assigned an approved status in the following scenarios:
 - The order was approved via a check or credit card payment gateway.
 - The order was assigned an approved status in *Order Administration*.
 - The order was manually created in *Order Administration* and assigned an approved status.
- **Declined** - Used for orders that have been placed and the transaction was declined due to insufficient funds or another reason. An order will be assigned a declined status in the following scenarios:
 - The order was declined via a check or credit card payment gateway.
 - The order was assigned a declined status in *Order Administration*.
 - The order was manually created in *Order Administration* and assigned a declined status.
- **Processed** - Used for orders for which payment has been received. Orders can only be assigned a processed status in *Order Administration*. Assigning a processed status will also disable the ability for a customer to cancel the order in *Customer Account Administration* (assuming customers have been given the ability to cancel orders in *Store Configuration*). Details about enabling order cancellation can be found in the [Installation and Configuration > Store Configuration](#) section.
- **Back Ordered** - Used for orders for which shipping has been delayed because ordered items are not in stock. Orders can only be assigned a back ordered status in *Order Administration*.
- **Shipped** - Used for orders for which all items have been shipped. Orders can only be assigned a shipped status in *Order Administration*.

- *Canceled* - Used for orders that have been canceled. An order will be assigned a canceled status in the following scenarios:
 - The order was assigned a canceled status in *Order Administration*.
 - The order was manually created in *Order Administration* and assigned a canceled status.
- Once an order has been assigned a canceled status, then all modifications except for administrator notes are disabled.

An order may also have a status that is not available as an option. This type of status is set internally by HostedStore when some action is taken that would not normally be applied when processing an order. Each is described below:

- *Mixed Status* - Assigned internally when individual items in the orders have a different status. Once all ordered items have been assigned a canceled or shipped status, then the order will no longer have a mixed status.
- *Canceled (Pending)* - Assigned internally when a customer has requested that an order be canceled in *Customer Account Administration*. This status allows the order to be reviewed by the store manager prior to canceling the order.

Multiple shipping vendors are available for selection. If one of the vendors is selected, then the tracking number will be displayed as a hyperlink to the shipping tracking page on the vendor's Web site on any of the default content pages where a tracking number is displayed. If a vendor other than those listed is needed, be sure to make a feature request for the vendor to be added. In the meantime, *Other* can be used and the appropriate code modifications can be made to content pages to display the tracking number as a hyperlink if that functionality is wanted.

Shipping vendors currently available for selection:

- *Airborne* - Airborne
- *FedEx* - Federal Express
- *UPS* - United Parcel Service
- *USPS* - United States Postal Service

Besides *Other*, one other shipping vendor, *Pick Up*, is available for selection. That entry is provided for retail locations where the customer takes immediate delivery or for situations where the customer picks up the order instead of having it delivered.

The image shows a screenshot of a web form titled "Global Order Modifications". The form contains three main sections:

- Change Status**: A dropdown menu with a small arrow icon on the right side.
- Shipping Vendor**: A dropdown menu with a small arrow icon on the right side.
- Tracking Number**: A wide, empty text input field.

 The form is enclosed in a thin black border.

Global order modifications

Processing Orders

Orders can go through many different stages and/or adjustments from the time the order is placed to the time when the order is finally shipped or canceled. As the order progresses through the processing timeline, the components on the edit order form will change to match the current state of the order.

Initial Status

Orders created via the storefront will have an initial status of *Pending* or *Approved*. The status of an order can be easily determined by looking at the status area in the order summary information that is displayed at the top of the form.

<u>Status</u>	<u>Status</u>
Pending	Approved
Posted 7/12/2002, 9:22 AM	Approved 7/12/2002, 9:26 AM
Created 7/12/2002, 9:22 AM	Posted 7/12/2002, 9:22 AM
	Created 7/12/2002, 9:22 AM

Order status summary of two orders: one pending, one approved

If the order has a status of *Pending*, then individual adjustment of ordered items will be disabled until the order has been assigned a different status.

Status	Vendor/Tracking	Item	Qty	Price	SubTotal
		Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
		Holster (NTN1987)	1	\$10.10	\$10.10

Ordered items for an order with a pending status

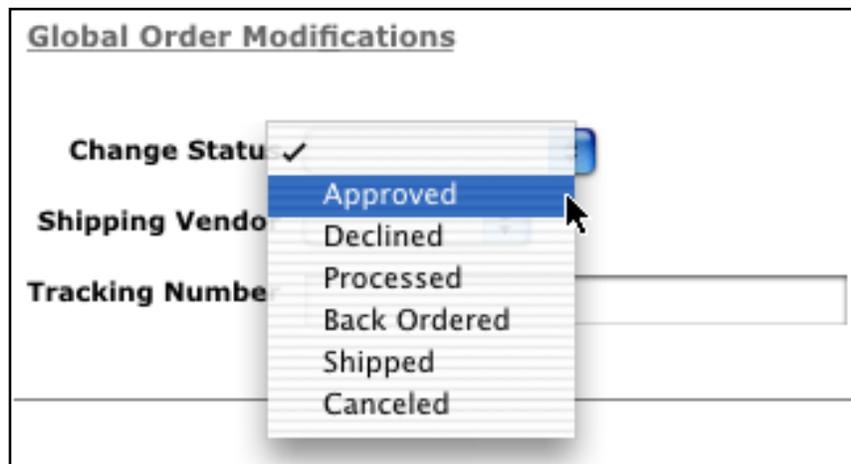
Status	Vendor/Tracking	Item	Qty	Price	SubTotal
<input type="text"/>	<input type="text"/>	Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
<input type="text"/>	<input type="text"/>	Holster (NTN1987)	2	\$10.10	\$20.20

Split Item

Ordered items for an order without a pending status

Orders with a pending status should be assigned a new status using the global order modifications area of the form. The status select list provides several choices, but it is recommended to only use one of the following three so that a logical status progression is maintained. Note that it is not mandatory to just select a status recommended below if a store's business rules are different than those expected by HostedStore. It just makes more sense to take an order from pending to approved to processed to shipped than to take an order from pending to shipped so that is why the recommendation is made.

- *Approved* - Assign if the availability of funds has been confirmed.
- *Declined* - Assign if the transaction was declined due to insufficient funds or another reason.
- *Canceled* - Assign if the order has been canceled.



Changing a pending status

Important: Once an order has been assigned a canceled status, then all future modifications (other than administrator comments) will be disabled. So think twice before assigning a canceled status.

Progressing from the Approved Status

There is a natural progression of an order's status as it moves through stages of the order processing process. Once an item has moved from a status of *Pending* to *Approved*, it should be assigned a status of *Processed* once payment has been received. The status change may be immediate or it may take a couple of days depending upon the situation. Receipt of payment may come in many different forms depending upon the payment method used:

- Preauthorized payment gateway transactions have been charged.
- Payment for a purchase order has been received.
- A mailed in check has been received.
- Payment via PayPal has been confirmed.

Progressing from the Processed Status

Two options are available after an order has been processed: continue to change the order status at the global level or change the status of individual items.

Global Status Changes

If the status of all items in an order will always be the same and/or the same shipping vendor and tracking number will be used to a ship an order, then global order modifications will suffice for completing the order processing procedure.

Each time the global status of an order is changed, a line is added to the status area of the order summary that notes the status and the date/time the status was applied.

Status
Shipped
<i>Shipped 7/12/2002, 7:16 AM</i>
<i>Processed 7/12/2002, 7:15 AM</i>
<i>Approved 7/12/2002, 7:15 AM</i>
<i>Posted 7/12/2002, 7:07 AM</i>
<i>Created 7/12/2002, 7:07 AM</i>

Order status summary of an order with multiple status changes

An order with a processed status can be changed to a back ordered, shipped or canceled status. Once changed to a back ordered status, an order can only be assigned a status of shipped or canceled. If the status of the order is shipped, then a shipping vendor and tracking number can be specified. If a tracking number is specified, then a shipping vendor is required. Once an order has been changed to a shipped or canceled status, then all global modifications except for administrator comments will be disabled.

Order totals adjustments can be made at the same time as status adjustments. Once an order has been changed to a shipped or canceled status, the ability to make order totals adjustments will be disabled.

SubTotal	\$30.33
Shipping	\$7.50
Tax	\$2.64
Discount	(\$0.00)
SubTotal Adjustment	<input type="text"/>
Shipping Adjustment	<input type="text" value="-7.50"/>
Tax Adjustment	<input type="text"/>
Discount Adjustment	<input type="text"/>
Total	\$32.97

Order totals adjustments for an order with a processed status

SubTotal	\$30.33
Shipping	\$7.50
Tax	\$2.64
Discount	(\$0.00)
SubTotal Adjustment	
Shipping Adjustment	(\$7.50)
Tax Adjustment	
Discount Adjustment	
Total	\$32.97

Order totals adjustment for an order with a shipped or canceled status

Individual Item Status Changes

In some cases it may be necessary to assign a separate status to individual items in an order. Each item can be assigned its own status of *Back Ordered*, *Shipped*, or *Canceled*. The date/time for each status change will be recorded for each item along with its status. Once an item has been changed to a shipped or canceled status, then its status can no longer be modified.

Status	Vendor/Tracking	Item	Qty	Price	SubTotal
Back Ordered 7/12/2002, 3:21 PM	<input type="text"/>	Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
Shipped 7/12/2002, 3:21 PM	FedEX ABCDEF123456	Holster (NTN1987)	1	\$10.10	\$10.10
Canceled 7/12/2002, 3:21 PM		Holster (NTN1987)	1	\$10.10	\$10.10

Ordered items, each with a different status

The global status for the order will be changed to *Mixed Status* if any of the items in the order have a different status. The one exception to this rule is if at least one item has a shipped status and all items have been assigned a shipped or canceled status. In that case, the global order status is changed to *Shipped*.

Status
Mixed Status
Approved 7/11/2002, 3:15 PM
Posted 7/11/2002, 3:15 PM
Created 7/11/2002, 3:15 PM

Order status summary of an order with mixed status

If more than one of an item was ordered and a different status needs to be applied to a subset of the quantity ordered, then the item can be split and the appropriate status can be applied to the split items.

Status	Vendor/Tracking	Item	Qty	Price	SubTotal
<input type="text"/>	<input type="text"/>	Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
<input type="text"/>	<input type="text"/>	Holster (NTN1987)	2	\$10.10	\$20.20

Split Item

Ordered items before splitting

Status	Vendor/Tracking	Item	Qty	Price	SubTotal
<input type="text"/>	<input type="text"/>	Case w/Swivel & Cutouts (RLN4927)	1	\$10.13	\$10.13
<input type="text"/>	<input type="text"/>	Holster (NTN1987)	1	\$10.10	\$10.10
<input type="text"/>	<input type="text"/>	Holster (NTN1987)	1	\$10.10	\$10.10

Ordered items after splitting an item

Batch Settlement

This feature is used to batch orders together according to store procedures and policies. For example, orders may be batched against a credit card batch report. Or if the store is used for mail order or retail sales, orders may be batched daily or weekly. Batching is not required for HostedStore to function. This feature is provided as a convenience for store administrators that want a means to reconcile store sales against reports from another process.

Clicking the **Batch Settlement** button will load a form that contains a summary of all unsettled orders regardless of status, i.e., even orders that are still in the cart or have been canceled will be displayed. Two fields are available for each order:

- **Adjustment** corresponds to the **SettlementAdjustment** field in the **Invoice** table. A decimal value can be entered to adjust the order total so it reconciles with the total reported from another process.
- **Batch Number** corresponds to the **SettlementBatch** field in the **Invoice** table. The name of the batch is entered in this field. Note that despite the name of the field, the value does not have to be a number.

Unsettled Orders List							
Invoice	Date Created	Status	Customer	Payment Type	Total	Adjustment	Batch Number
186	4/25/2002 4:16 PM	Approved	Doe	Check by Mail/Phone	\$406.05	<input type="text"/>	<input type="text"/>
257	7/11/2002 3:15 PM	Approved	Doe	Cash	\$26.99	<input type="text"/>	<input type="text"/>

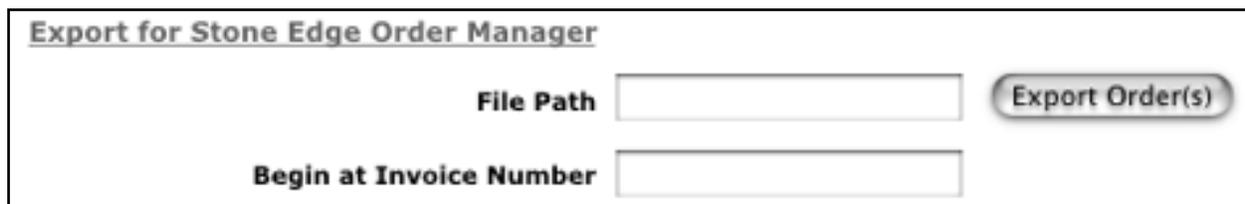
Batch settlement form

Export for Stone Edge Order Manager™

This section of *Order Administration* creates text files that contain order information in a format that is recognized by the Stone Edge Order Manager software. More information about Stone Edge Order Manager can be found at the Stone Edge Web site (www.stoneedge.com).

Creating the Export File

Select a file path from the */hostedstore/ordermanager/* directory, a starting invoice number if wanted, and click the **Export Order(s)** button. A new export file will be created with a file name corresponding to the date and time of the export in the format *YYYYMMSS_HHMMSS.txt*. If a starting invoice number is not specified, then all invoices will be exported.



The screenshot shows a web form titled "Export for Stone Edge Order Manager". It features two text input fields: "File Path" and "Begin at Invoice Number". To the right of the "File Path" field is a button labeled "Export Order(s)".

Export for Order Manager form area of order administration

Export File Structure

This feature can also be used to create a flat file representation of orders provided that the export format is compatible with the requirements of another system. The structure of the export file is provided below. The file is in Comma Separated Format (CSV) format with each entry surrounded by double quotes and separated by commas (except for the `\r\n` line feeds/carriage returns).

- Header
- 1.0
- HostedStore
- The `HS_StoreName` preference variable set in *Store Configuration*.
- Current date in MM/DD/YYYY format.
- Current time in HH:MM:SS format.
- BLANK
- |
- BLANK
- \r\n
- Repeat for each order:
 - **ID** field from the **Invoice** table.
 - Date portion of the **DatePosted** field from the **Invoice** table in MM/DD/YYYY format.
 - Time portion of the **DatePosted** field from the **Invoice** table in HH:MM:SS format.
 - **FirstName** field from the **Customer** table.
 - **LastName** field from the **Customer** table.
 - Concatenated **FirstName** and **LastName** fields from the **Customer** table.
 - **Company** field from the **Customer** table.
 - **Email** field from the **Customer** table.
 - **Address1** field from the **Customer** table.
 - **Address2** field from the **Customer** table.
 - **City** field from the **Customer** table.
 - **State** field from the **Customer** table.
 - **Postal** field from the **Customer** table.
 - **Country** field from the **Customer** table.

- END_ITEM
- \r\n

Connecting Directly to HostedStore

HostedStore supports the ability for Order Manager to connect directly to HostedStore and obtain the order information. Order Manager should be configured to connect to HostedStore at: *PATH TO STORE/HostedStore.LassoApp?-ResponseLassoApp=adminstore/ordermanager.lasso*.

Two parameters must be passed in the URL:

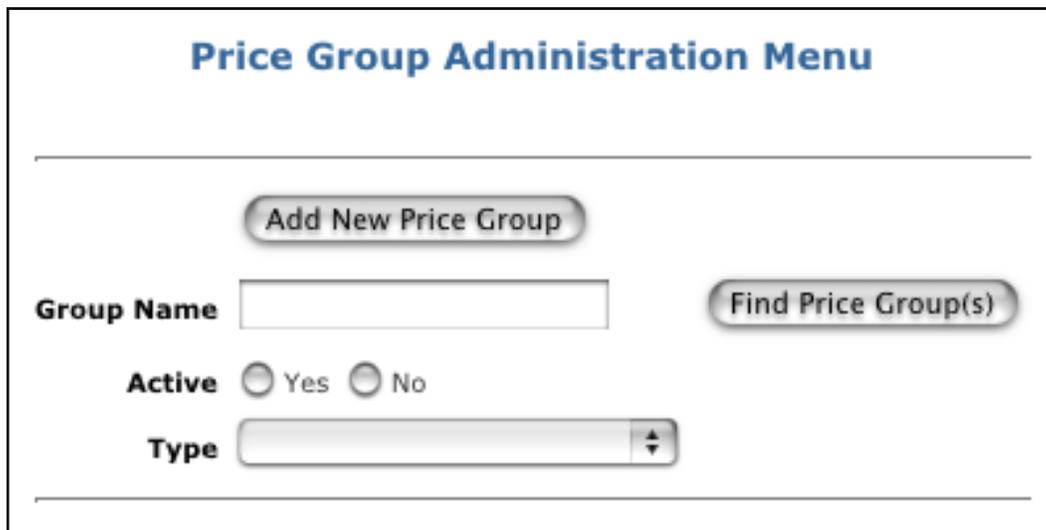
- *om_username* - The username registered with Lasso Administration. This corresponds to the *HS_Username* preference variable set in *Store Configuration*.
- *om_password* - The password registered with Lasso Administration. This corresponds to the *HS_Password* preference variable set in *Store Configuration*.

One optional parameter may be passed in the URL:

- *ID* - The starting invoice number. If a starting invoice number is not specified, then all invoices will be exported.

Price Group Administration

The *Price Group Administration* menu provides for two different operations: Add New Price Group, and Find Price Group(s). For any field requiring a percentage value, the value should be expressed as a decimal without the percent symbol. For example, 15% would be expressed as .15.



The image shows a screenshot of a web application interface titled "Price Group Administration Menu". The interface is enclosed in a rectangular border. At the top, the title "Price Group Administration Menu" is centered in a blue font. Below the title, there is a horizontal line. Underneath the line, there are two buttons: "Add New Price Group" on the left and "Find Price Group(s)" on the right. Below these buttons, there are three input fields: "Group Name" with a text box, "Active" with two radio buttons labeled "Yes" and "No", and "Type" with a dropdown menu. The form is separated from the rest of the page by another horizontal line at the bottom.

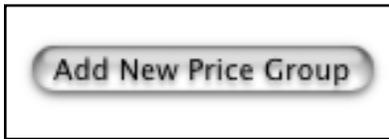
Price group administration menu

Sections

- [Add New Price Group](#) - Create a new price group.
- [Find Price Group\(s\)](#) - Search for and edit existing price groups.

Add New Price Group

Create a new price group by clicking the **Add New Price Group** button.



Add new price group form area of the price group administration menu

Multiple parameters must be set for the price group so that the system can determine how to properly adjust product prices. All of the fields on this form correspond to fields in the **PriceGroup** table only. An explanation of each field and its corresponding field in the **PriceGroup** table are detailed by below:

- **Name** corresponds to the field of the same name in the table (*REQUIRED*). This value should be unique. If a price group name is assigned and it already exists, then an error will be returned.
- **Active** corresponds to the **Active** field in the table (*REQUIRED*). **Active** sets whether or not the pricing dictated by the price group should be used. If set to *Yes*, then the price group will be applied to product pricing if a customer belongs to the price group. If set to *No*, then the price group will not be applied to product pricing even if a customer belongs to the price group.
- **Type** corresponds to the field of the same name in the table (*REQUIRED*). **Type** is used to set what type of product price adjustments should be made for customers belonging to the price group. An explanation of each selection is provided below:
 - *Discount (%)* - Product price is calculated by applying the discount percentage specified in the **Value** field (see below) to the default price (lowest price of all applicable prices: regular price, sale price, and member price).
 - *At Retail* - Product price is always the normal price (**Price** field in the **Product** table). This price group type disables the default price calculation (lowest price of all applicable prices: regular price, sale price, and member price).
 - *Discount from Retail (%)* - Product price is calculated by applying the discount percentage specified in the **Value** field (see below) to the normal price (**Price** field in the table). This price group type disables the default price calculation (lowest price of regular price, sale price, and member price).
 - *At Cost* - Product price is the base cost for the product (**BaseCost** field in the table).
 - *Markup Cost (%)* - Product price is calculated by applying the markup percentage specified in the **Value** field (see below) to the base cost (**BaseCost** field in the table).
- **Value** corresponds to the field of the same name in the table. If the price group type is *Discount (%)*, *Discount from Retail (%)*, or *Markup Cost (%)* then this field is required. The value should be set to the percentage discount or markup expressed as a decimal. For example, a 15% discount would be entered as .15.
- **Tax Exempt** corresponds to the **TaxExempt** field in the table. **Tax Exempt** sets whether or not the customer assigned to the price group will be charged tax (assuming the tax calculation script contains the appropriate code). If set to *Yes*, then tax will be applied to the order total if a customer belongs to the price group. If set to *No*, then tax will be applied to the order total if a customer belongs to the price group (assuming the customer resides in a taxable jurisdiction).

Create Price Group

* denotes an empty required field.

Name

* **Active** Yes No

* **Type**

IMPORTANT: Express percentages as decimals. (ex. 15% is .15)

Percentage

Tax Exempt Yes No

Create Price Group

Add price group form

Find Price Group(s)

A search form is provided to search for existing price groups. If more than one match is found, then a listing of all matching price groups will be provided. Many of the columns can be sorted in ascending or descending order. If only one match is found then the matching price group is displayed in an edit price group form. The fields for editing price groups are the same as the fields available when adding a new price group.

The fields that can be searched are listed below along with the operator used in the search:

- **Name** - Contains (cn)
- **Active** - Equals (eq)
- **Type** - Equals (eq)
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



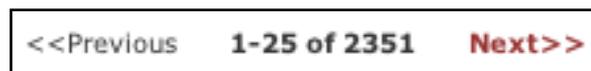
The image shows a search form with three main sections. The first section is labeled 'Group Name' and contains a text input field. To the right of this field is a button labeled 'Find Price Group(s)'. The second section is labeled 'Active' and contains two radio buttons, one for 'Yes' and one for 'No'. The third section is labeled 'Type' and contains a dropdown menu with a small arrow icon on the right side.

Price group search form area of the price group administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate pricegroup in the listing and click the **Name** hyperlink for the price group. The price group will be loaded into the edit form. The edit price group form is nearly identical to the add price group form. See the [Store Administration > Price Group Administration > Add New Price Group](#) section for an explanation of each of the fields on the form.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.



Price group found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Price Group List

Click on the group name to edit the group record.

<u>Name'</u> ▲▼	<u>Active</u> ▲▼	<u>Type</u> ▲▼	<u>Percentage</u>	<u>Customers</u>
Test Group 1	Y	At Retail	0.05500	1
Test Group 2	Y	At Retail	0.00000	0

Price group search results list

Product Administration

Product Administration contains all of the products in the store catalog. Products in the store catalog are not necessarily for sale in the online store. The menu offers the ability to list all products in the catalog, add a new product, or find and edit an existing product.

Important: It is possible to import products directly into the database through the database client or via another method. However, bypassing the browser based product administration interface should only be done with great care and with a thorough understanding of how the product data should be structured. This is especially true with regards to products with attributes. Product data with an incorrect structure will cause HostedStore to function improperly. *Currently, bypassing the browser interface is not recommended nor supported.*

The screenshot shows the 'Product Administration Menu' interface. It is divided into several sections:

- Product Catalog:** Includes a 'Product ID' input field with an 'Add New Product' button, and a search section with 'Product ID', 'Product Name', 'Department', 'Category', and 'SubCategory' input fields, each with a 'list' button. Below these are radio buttons for 'Featured' and 'On Sale' (Yes/No), and date pickers for 'Created' and 'Modified' (Month, Day, Year). A 'Limit' input field is also present.
- Catalog-wide Functions:** Contains buttons for 'Adjust Pricing', 'Adjust Inventory', 'Generate Static Product Pages', and 'Export Products to HostedStores.com'.
- Global Price Adjustment:** Features a 'Products' dropdown, an 'Adjust Prices' button, and a 'Price(s) to Adjust' section with checkboxes for 'MSRP', 'Price', 'Sale', and 'Member'. An 'Adjustment' input field with a percentage dropdown is also included.
- File Upload:** Includes a 'Type of File' dropdown, an 'Upload File(s)' button, and a 'Number of Files' dropdown.

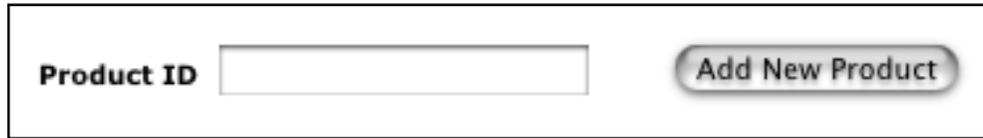
Product administration menu

Sections

- [Add New Product](#) - Create a new product.
- [Find Product\(s\)](#) - Search for and edit existing products.
- [Adjust Pricing](#) - Simultaneously adjust the pricing for all products.
- [Adjust Inventory](#) - Simultaneously adjust the inventory for all products.
- [Generate Static Product Pages](#) - Create static product detail pages for submission to search engines or to reduce database use.
- [Global Price Adjustment](#) - Adjust prices by the same monetary amount or percentage.
- [File Upload](#) - Upload multiple description files and product images.

Add New Product

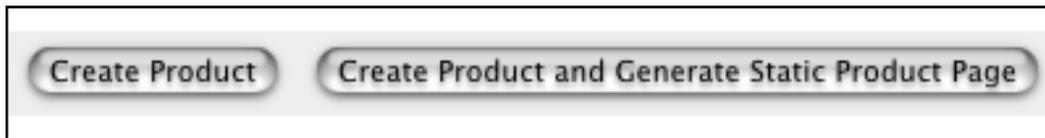
Add a new product by entering a unique **Product ID** and clicking the **Add New Product** button. The system will prompt for a new product ID if a product with the submitted ID already exists.



Add new product form area of the product administration menu

The form for adding a product will display with its chosen ID. Note that due to the size of the form and the number of fields there will be a slight delay as the form loads. The amount of information on the form may be overwhelming at first, but it is unlikely that all of the items on the form will be used for any single product. The exhaustive list of configuration possibilities is provided so that even complex products can be easily entered into the store catalog. Each field of the form is described below in groups corresponding to the section headings on the form.

A **Create Product** and a **Create Product and Generate Static Product Page** button appear multiple times on the form. The **Create Product** button simply adds the product to the table. The **Create Product and Generate Static Product Page** button adds the product to the table and creates a static product detail page. Details about static pages can be found in the [Store Administration > Product Administration > Generate Static Product Pages](#) section. Clicking any of the buttons will submit the data for the entire form--there is no need to click the button for each section of the form.



Create product buttons on the add product form

This form uses fairly complex error checking to make sure that products are properly configured. Because the logic is conditional on the values in different fields, it would not have been meaningful to place the standard * indicator next to required fields. Almost all of the fields can be left blank if the product is not made available for sale. Once marked for sale, several fields are required.

In general, use common sense when configuring the product. For example, a product that is made available for sale must have a price, product name, description, etc. Another example of conditional logic is that if a product is marked as being on sale, then it must have a sale price. Don't worry--the system will not allow an improperly configured product to be submitted. The error messages are very informative as to what criteria were not met. Hopefully after a few products are created this will become an intuitive process.

All of the fields on this form correspond to fields in the **Product** table only. An explanation of each field and its corresponding field in the **Product** table are detailed by group below.

Product Summary

The top of the form contains general information about the product. None of this information can be directly modified with fields on the form.

If a thumbnail image for the product has already been uploaded to the server, then it will display at the top. Beneath the image, the **ProductID** field is displayed followed by the **DateCreated** and **DateModified** fields formatted according to date and formatting preferences specified *Store Configuration*.

A **Product Detail Link** textbox provides a hyperlink to the product detail content page. This can be used for hard coded hyperlinks in other areas of the site (or other sites) to link to the product detail content page. Use the same hyperlink as the link for adding the

item to the shopping cart if the product has attributes. Details about product attributes can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) > section.

An **Add to Cart Link** provides a hyperlink to add the product to the shopping cart. This can be used for hard coded hyperlinks in other areas of the site (or other sites). Do not use this as the add to cart hyperlink if the product has attributes. Use the product detail hyperlink instead. If this hyperlink is used for products with attributes, the customer will receive an error message that the product has not been properly configured. So there isn't really any adverse effects with regards to proper functionality if this hyperlink is accidentally used for products with attributes--it just results in a slightly poorer customer experience.



The screenshot shows a form titled "Create Product". Below the title, the "Product ID" is set to "TestProduct". There are two input fields for URLs. The first is labeled "Product Detail Link (also use for Add to Cart link if product has attributes)" and contains the URL "http://hdev.hostedstore.com/HostedStore.LassoApp?~ResponseLassoApp=detail.lasso&ID=03e78a89f98e82b8d5c719bd5d8a47bb". The second is labeled "Add to Cart Link (only use for products without attributes)" and contains the URL "http://hdev.hostedstore.com/HostedStore.LassoApp?~ResponseLassoApp=cart.lasso&add=03e78a89f98e82b899fe431b5b02ad0d".

Product summary area of the add product form

Categorization

This group contains the name of the product and six levels of classification to categorize the product. Note that just because a field has a particular name (**Category**, **SubCategory**, **Department**, **Manufacturer**, etc) does not mean that it must be used strictly for that purpose. Use the fields in a manner that matches the business logic of the store.

- **Product Name** corresponds to the **ProductName** field in the table. Use this field to assign a name to the product. Note that the product name does not need to be unique because the **ProductID** set earlier is used to uniquely identify the product. **Product Name** is a searchable field on the default advanced search form content page (*/hostedstore/storefront/html/search_form.lasso*) and is also searched via the quick search form on all default store templates.
- **Category**, **SubCategory**, **SubCategory 2**, **SubCategory 3** correspond to the **Category**, **SubCategory**, **SubCategory2**, **SubCategory3** fields in the table, respectively. **Category** and **SubCategory** are searchable fields on the default advanced search form content page (*/hostedstore/storefront/html/search_form.lasso*). Both are also searched via the quick search form on all default store templates. It is expected that **Category** and **SubCategory** will provide the needed product heirarchy for most stores. But just in case, additional category related fields are available. Large catalogs may require detailed categorization to facilitate easy navigation, whereas stores with only a handful of products may not even need to categorize products.
- **Department** corresponds to the field of the same name in the table. **Department** can be useful as an additional categorizing field or as an internal classification system for store administration purposes only. Classifying by department can also allow for complex stores where a product is repeated (note that each product will need to have a unique **ProductID**) in different sections (departments) of the online store to determine where the product sells the best.
- **Manufacturer** corresponds to the field of the same name in the table. **Manufacturer** is not required, but it is a searchable field on the default advanced search form content page (*/hostedstore/storefront/html/search_form.lasso*) and is also searched via the quick search form on all default store templates.
- **Search Rank** corresponds to the **SearchRank** field in the table. Specify a two digit number to force the product to the top of product search results listings. When a product search is performed, the result set is sorted on the **SearchRank** field first and then sorted on the sort field specified for the search. Note that the order of search ranking is counter intuitive due to the way that sorting works - a product with a rank of *99* will appear before a product with a rank of *1*.

<u>Categorization</u>	
Product Name	<input type="text"/>
Department	<input type="text"/>
Category	<input type="text"/>
SubCategory	<input type="text"/>
SubCategory 2	<input type="text"/>
SubCategory 3	<input type="text"/>
Manufacturer	<input type="text"/>
Search Rank	<input type="text"/>

Categorization area of the add product form

Complementary Products

Complement 1 through **Complement 6** correspond to the **Complement1** through **Complement6** fields in the table, respectively. The **ProductID** for the complementary product should be entered into the appropriate field. The order of the complements determines the order in which the products are displayed on the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*).

Note that the validity of complementary product IDs is not verified when the product is created. This leniency is granted to eliminate endless error loops that would result from two products that list each other as complements. The existence of each complementary product is verified before the products are displayed on the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*). If a product does not exist, then it will not be available for display.

Multiple product IDs can be specified in one complementary field provided that each product ID is separated by two adjacent pipe characters (| |). The pipe character key is located in the shift position above the backslash (\) key (usually located above the return/enter key) on the keyboard. Placing multiple product IDs in one complementary product field allows products to be grouped. The grouping of products can be utilized on the product detail content page to display each group in a different fashion.

Important: Complementary product fields must be completely sequentially to function properly. For example, if a product has two complementary products, then use the **Complement 1** and **Complement 2** fields only.

Example uses of complementary products are to display up-sale, cross-sale, or needed accessory items to the customer on the product detail content page. Note that the use of this feature may or may not be a good idea for a store. Many customers appreciate the display of related items and especially the display of needed accessory items with a product. Others view such a practice as pushy or overselling and ultimately leave the store without purchasing anything.

<u>Complementary Products</u>	
Complement 1	<input type="text"/>
Complement 2	<input type="text"/>
Complement 3	<input type="text"/>
Complement 4	<input type="text"/>
Complement 5	<input type="text"/>
Complement 6	<input type="text"/>

Complementary products area of the add product form

Pricing

This group contains four different pricing structures for a product as well as additional parameters that dictate how and where the product is available for display. Use the fields in a manner that matches the business logic, enabled features, and promotional policies of the store. Note that, as with the categorization fields, the field name for the prices does not necessarily dictate that the field must be used specifically for that purpose. For example, if wholesale pricing is available to business customers, then the business customers can be given membership status and the member price can be the wholesale price offered to them.

Important: All prices should be specified with a decimal point (even if a different decimal delimiter is used for display) and without any currency symbols. The decimal delimiter and currency symbol used for display purposes are specified with the `HS_CurrencyDecimal` and `HS_CurrencySymbol` preference variables, respectively, in *Store Configuration*.

- **Sell product in store** corresponds to the **WebSell** field in the table. Checking the checkbox makes the product eligible for sale. Other factors like current inventory will affect if the product is actually displayed in the online store. Exactly what determines which products are ultimately displayed for sale is detailed under the *Inventory* heading below.

This simple checkbox may seem to be trivial, but it actually offers a lot of flexibility with regard to the store catalog:

- Products can be preloaded into the catalog for sale at a later date.
- Out of stock products can be temporarily removed from the store until inventory is received. Note that HostedStore has built-in capabilities to temporarily remove, but still display, out of stock product from the catalog (details are provided in the **Minimum on Hand to Sell** field under the *Current Inventory* heading below), but there may be scenarios where manual removal is a superior technique.
- Discontinued products can be kept in the catalog for reference when reviewing past orders.
- Products used solely for mail order or in-store sales can be easily separated from products in the online catalog.
- **Feature this product** corresponds to the **Featured** field in the table. Checking the checkbox makes the product eligible for display on the default store home content page (`/hostedstore/storefront/html/index.lasso`). Other factors like current inventory will affect if the product is actually displayed in the online store. Exactly what determines which products are ultimately displayed for sale is detailed under the *Inventory* heading below.

The product will automatically be displayed on the default store home content page (`/hostedstore/storefront/html/index.lasso`) if the total number of products marked as featured is less than the featured product limit set as the `HS_FeaturedLimit` preference variable in *Store Configuration*. If the number of marked products exceeds the limit, then the product may or may not display as a featured product.

- **MSRP** corresponds to the field of the same name in the table. This field can be used to display a suggested retail price for a product on the default product detail content page (`/hostedstore/storefront/html/product_detail.lasso`). All stores may not have a need for this feature so it can be toggled on or off with the `HS_Display MSRP` preference variable in *Store Configuration*.
- **Price** corresponds to the field of the same name in the table. This field is used for the normal or everyday price for the

product.

- **Sale Price** corresponds to the **SalePrice** field in the table. This field stores the price for the product when it is marked for sale pricing (the **Product is currently on sale** checkbox is checked).
- **Member Price** corresponds to the **MemberPrice** field in the table. This field is used to set the member price for the product if a membership program is used in the store. Note that if a customer is a member they will receive the membership price unless the sale price for a product on sale is less (HostedStore always determines the lowest possible price for a customer when all business rules are applied).

Important: A customer with a membership will only receive the member price if they have logged in prior to placing items in the shopping cart. An explanation should be given to members that they should log into *Customer Account Administration* prior to shopping in the store if they want to receive member pricing or if they want to be able to purchase member only items (details provided under the *Memberships* heading below). Prices are not necessarily recalculated during the checkout process when the customer logs in to identify their account. Details about logging into *Customer Account Administration* can be found in the [The Customer's Perspective > Account Administration](#) section.

- **Product is available to members only** corresponds to the **MemberOnlyItem** field in the table. Check the checkbox if the product can only be purchased by customers with a current membership plan.

Important: If a product is available to members only, only the following restrictions are enforced if the customer is not a current member:

- Member only products will not appear in product search results.
- Member only products cannot be added to the cart.
- Member only products cannot be added to a wish list.

The restriction does not apply if:

- The member only product is purchased from a saved cart.
- The member only product is purchased from a shopping list.
- The member only product is purchased from a wish list.

- **Product is currently on sale** corresponds to the **OnSale** field in the table. Checking the checkbox will switch pricing for a product from the price set in the **Price** field above to the price in the **Sale Price** field above.
- **Volume Price** corresponds to the **VolumePrice** field in the table. This field can be used to specify quantity ranges and a price for that range. Multiple ranges and prices can be entered by separating each entry with a carriage return.

Entries must follow these simple structure rules:

- Entries should be of the format *QuantityLowerBound,QuantityUpperBound,Price*. Note the comma between each value and that there is not a space after the comma. Each entry should be a contiguous string of non-whitespace characters.
- The lower bound must be less than or equal to the upper bound
- The lower bound of an entry must be greater than the upper bound of the prior entry. This does not apply to the first entry.

Example: Quantities of 11 to 50 should have a price of 10.00. Quantities of 51 to 100 should have a price of 9.50. Quantities of 101 to 1000 should have a price of 9.00. The following would be entered into the field:

```
11,50,10.00
51,100,9.50
101,1000,9.00
```

If a quantity range is not specified in the volume pricing, then the price will be determined by the lowest applicable price from the regular price, sale price, and member price. Using the example above, if a customer ordered 1001 units, then the volume price would not apply because the quantity ordered does not fall in the range of any of the volume pricing entries. The upper bound of the last entry should be sufficiently large for products where it is expected that a customer will order large quantities.

Note that the price for an item is always determined by the lowest possible price when all pricing rules have been applied. Using the example above, if a customer orders 75 of the item, then the price should be 9.50 according to the volume pricing rules. However, if the product is currently on sale for 9.25, then the customer will receive a price of 9.25 because it is a lower price than the result from applying the volume pricing rule.

Volume pricing is applied to configured products, not to the product at the product ID level. So if, for example, a customer orders 5 of product A with selection X for attribute 1 and 2 of product X with selection Y for attribute 1, then the customer will not receive volume pricing for 7 items. Volume pricing will be applied to the 5 units and the 2 units of the product separately.

- **Base Cost** corresponds to the **BaseCost** field in the table. This field can be used to calculate net revenues on a product. Obviously this does not determine net profit for an item because other cost factors like labor, overhead, etc probably will not be factored into the base cost.

Base cost is not averaged across inventory. That value must be calculated manually or a more comprehensive accounting package should be used to calculate profitability. For example, if the current inventory of five items was purchased at \$10.00 and another 100 items are received at a cost of \$4.50, then the appropriate base cost must be calculated and entered by the store manager.

Pricing help

Sell product in store (publish in store catalog).

Feature this product.

MSRP

Price

Member Price

Product is available to members only.

Sale Price

Product is currently on sale.

Volume Pricing

Base Cost

Pricing area of the add product form

Inventory

Fair warning: The fields in this group are complex because of their interaction with other product and store settings. Read this section carefully to ensure a properly configured product. Like everything else though it should become intuitive after a few products are created.

- **Current Inventory** corresponds to the **OnHand** field in the table. This field is used for inventory tracking purposes as well as to determine if a product will be available for sale. This quantity will be decremented by the appropriate value each time the product is sold. Note that the inventory is associated with the product ID so it is not necessarily tied strictly to physical products on the shelf. For example, a t-shirt may have small and medium sizes, but both may be tracked together under the same product ID (details can be found under the *Product Attributes* heading below).
- **Build To Inventory** corresponds to the **BuildTo** field in the table. This field can serve as a reminder of when additional

inventory should be ordered from the supplier. This value is not at all critical to store operations--just a simple added feature. Build to inventory is a quantity threshold at which to be reminded to order more inventory. The store will generate an inventory order report based upon the difference between the quantity of product on hand and the build to quantity. For example, if the current inventory is 3 and the build to quantity is 10, then the inventory order report will suggest purchasing 7 more units of the product from the supplier(s). Details about the suggested inventory order report can be found in the [Store Administration > Reports > Date Independent Reports > Suggested Inventory Order](#) section.

- **Minimum on Hand to Sell** corresponds to the **WebSellQty** field in the table. This field sets the minimum quantity of the product that must be in inventory for the product to be available for sale. If the inventory dips below that value, then the product will no longer be displayed to customers even if the product has been flagged as eligible for sale. This feature can be used to keep some product on hand for replacements, for mail order customers, or for retail customers.

Consider unchecking **WebSell** if the **OnHand** quantity falls below the **WebSellQty** quantity and is expected to stay that way for awhile. Products that are no longer for sale due to low inventory can interfere with navigation numbers on the default product search results content page (*/hostedstore/storefront/html/search_list.lasso*).

Minimum on Hand to Sell can have a negative value. This feature can be used to display out of stock product in the store. Setting the field to a large negative number will display the product as out of stock in the store if **Current Inventory** is less than **Minimum on Hand to Sell** even if **Current Inventory** is zero or a negative value.

Important: If **Sell product in store** is checked, then out of stock product will still be sold in the store unless **Display product if out of stock** (see explanation below) is checked.

- **Enforce inventory at checkout** corresponds to the **EnforceInventory** field in the table. If checked, at checkout the inventory available for sale (**Current Inventory** value less the **Minimum on Hand to Sell** value) will be compared to the quantity of the item ordered by the customer. If the available inventory is not greater than or equal to the quantity ordered, then the customer is returned to the shopping cart detail page and a message is displayed that informs the customer which of the items in the cart exceeds the current inventory. The excess quantity is not automatically removed because it is possible that multiple cart items contain the same product (products with attributes that have different selections) so the customer should make the choice of which item(s) to remove from the cart.

Important: This feature is meant to aid in the reduction of the oversale of limited quantity items, clearance items, one-of-a-kind items, etc. However, it is not guaranteed to completely eliminate overselling items. It is possible in high traffic situations that two or more customers could order the last item(s) simultaneously. All incoming orders should be examined carefully to determine if all items can be fulfilled. For wildly popular items in high traffic stores it may be a good idea to assign a positive value to **Minimum on Hand to Sell** to act as a buffer for possible oversale of the item.

- **Notify when out of stock level is reached** corresponds to the **OutOfStockNotify** field in the table. If checked, when the **Current Inventory** value becomes less than the **Minimum on Hand to Sell** value then an email notification is sent to the email address specified by the **HS_StoreManagerEmail** preference variable in *Store Configuration*.
- **Out of Stock Message** corresponds to the **OutOfStockMessage** field in the table. If this field has a value, then it overrides the default out of stock message set with the **HS_OutOfStockMessage** preference variable in *Store Configuration*. HTML can be used in the message. The out of stock message displays on the default product search results content page (*/hostedstore/storefront/html/search_list.lasso*) and the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*) if the quantity on hand is either zero or less than the minimum quantity needed to sell online.
- **Display product if out of stock** corresponds to the **OutOfStockDisplay** field in the table. If checked, products that are out of stock (the quantity on hand is either zero or less than the minimum quantity needed to sell online) will be displayed to the customer, but will not be available for purchase.
- **Shadow Products** corresponds to the **ShadowProducts** field in the table. A list of product IDs and quantities that should have their inventory adjusted when the product is sold. Product ID and the corresponding quantity should be separated by a comma (,) and product ID and quantity pairs should be separated by two adjacent pipe characters (| |).

The following example illustrates how to configure a product so that the inventory of a product with an ID of *ProductID1* will be decreased by 1 and the inventory of a product with an ID of *ProductID2* will be decreased by 2 when the main product is purchased:

```
ProductID1, 1 | ProductID2, 2
```

This feature can be used when the top level product for sale is comprised of other products from the store catalog which may or

may not be individually sold. For example, a pre-built computer may be sold as a package, but it is comprised of multiple shadow products like the case, power supply, memory, processor, hard drive(s), etc.

The screenshot shows the 'Inventory' section of a form. At the top, there are two links: 'Inventory' (underlined) and 'help' (in red). Below these are four input fields: 'Current Inventory', 'Build To Inventory', and 'Minimum on Hand to Sell'. Under 'Minimum on Hand to Sell', there are two checkboxes: 'Enforce inventory at checkout.' and 'Notify when out of stock level is reached.'. Below these is the 'Out of Stock Message' section, which includes a text input field and a checkbox labeled 'Display product if out of stock (but do not sell)'. At the bottom is the 'Shadow Products' section, which contains a large text input field.

Inventory area of the add product form

Taxation

- **Product is taxable** corresponds to the **Taxable** field in the table. Checking the checkbox will make the product subject to taxation. Details on how to specify tax calculations can be found in the [Shipping and Taxation > Taxation](#) section.
- **Class** corresponds to the **TaxClass** field in the table. Specify the tax class to which the product applies so that special tax rules can be applied in the tax calculation script (`/hostedstore/includes/calculations/calculate_tax.lasso`). Details about the tax calculation script can be found in the [Shipping and Taxation > Taxation](#) section.

The screenshot shows the 'Taxation' section of a form. At the top, there are two links: 'Taxation' (underlined) and 'help' (in red). Below these is a checkbox labeled 'Product is taxable.'. Below the checkbox is the 'Class' section, which contains a large text input field.

Taxation area of the add product form

Shipping

- **Shipping Type** and **Shipping Value** correspond to the **ShippingType** and **ShippingValue** fields in the table, respectively. **Shipping Type** determines what method will be used to calculate shipping costs and **Shipping Value** provides one of the inputs used to calculate the shipping cost. Details on how to specify shipping calculations are provided in the [Shipping and Taxation > Shipping](#) section. If no shipping is to be charged for an item, then leave the field blank. Otherwise, three different calculation methods are currently supported:
 - *Fixed* charges a shipping cost of the monetary value specified in the **Shipping Value** field.
 - *Quantity* is an arbitrary type with the number of items the product represents specified in the **Shipping Value** field. For example, a bookstore may have different size books. A small paperback book may represent a quantity of one that costs \$.75 to ship, whereas a thick hard cover book may represent a quantity of four that costs \$3.00 to ship. The small paperback book would be set up with a **Shipping Type** of *Quantity* and a **Shipping Value** of *1*. The thick hard cover book would be set up with a **Shipping Type** of *Quantity* and a **Shipping Value** of *4*. The `HS_ShippingQuantityRate` preference variable would be set to *.75* in *Store Configuration*.
 - *Weight* is the actual weight of the product in any units (pounds, kilos, etc). The only stipulation is that the same weight unit of measure must be used for all weight-based products for the shipping calculation script to work properly. For example, a chair that weighs 75 pounds would be set up with a **Shipping Type** of *Weight* and a **Shipping Value** of *75*. The shipping calculation script can then be used to specify different shipping rates for different weight ranges. Details about creating a shipping calculation script are provided in the [Shipping and Taxation > Shipping](#) section.

Important: Shipping costs are fixed values regardless of any chosen attributes (details about configurable products can be found under the *Product Attributes* heading below). If the shipping costs do vary in reality (added weight with certain options, for example), then a couple of options are available. One method involves setting the shipping value equal to the average shipping cost for all possible product configurations. A second method is to create separate products for configurations that have significantly different shipping costs. The two methods described above are not an exhaustive list of solutions. There may be additional solutions to this problem.

The image shows a screenshot of a form section titled 'Shipping'. At the top right of this section are two links: 'Shipping' (underlined) and 'help' (in red). Below these links are two input fields. The first is labeled 'Shipping Type' and is a dropdown menu with a small arrow icon on the right. The second is labeled 'Shipping Value' and is a standard text input box.

Shipping area of the add product form

Quantity Limits

- **Per Order Min** corresponds to the **PerOrderMin** field in the table. If a value is entered, then the customer must order at least that quantity of the product.
- **Per Order Max** corresponds to the **PerOrderMax** field in the table. If a value is entered, then the customer may order at most that quantity of the product.
- **Per Customer Max** corresponds to the **PerCustomerMax** field in the table. If a value is entered, then the customer may order at most that quantity of the product cumulative across all orders. Note that this restriction can be easily circumvented by simply creating a new account.

<u>Quantity Limits</u>	
Per Order Min	<input type="text"/>
Per Order Max	<input type="text"/>
Per Customer Limit	<input type="text"/>

Quantity limits area of the add product form

Dependencies/Restrictions

- **Class** corresponds to the **RestrictClass** field in the table. Entering a value assigns the product to a product group that can be used to restrict how other products are purchased.
- **Restrict To** corresponds to the **RestrictTo** field in the table. Enter the name of a product class. The product cannot be added to the shopping cart unless a product from the restrict class has been added to the shopping cart first. If the customer later removes the required product from the shopping cart, then the customer will not be able to checkout unless a product from the required product class is added to the shopping cart or unless the restricted product is removed from the shopping cart.
- **Price Group(s)** corresponds to the **PriceGroup** field in the table. All active price groups will be listed in the select list. Select the price groups to which the product should be restricted. If one or more price groups are selected, then the product will only show up in the search results list for customers who belong to one of the price group(s). More information about price groups can be found in the [Store Administration](#) > [Price Group Administration](#) section.

<u>Dependencies / Restrictions</u>		help
Class	<input type="text"/>	list
Restrict To	<input type="text"/>	list
Price Group(s)	<input type="text" value="Test Group 1"/> <input type="text" value="Test Group 2"/>	

Dependencies/Restrictions area of the add product form

Memberships

- **Product is a membership** corresponds to the **IsMembership** field in the table. If the product itself is a membership plan for the store, then this checkbox should be checked. Customers with a current membership plan will be offered the member price for products and will have access to products that are available to members only (if applicable).
- **Days in Plan** corresponds to the **MemberPlanDays** field in the table. Membership plans expire and need to be renewed eventually. If the product is a membership, then select the appropriate number of days in the membership plan.

Memberships area of the add product form

Gift Certificates

This feature has not been implemented yet.

- **Product is a gift certificate** corresponds to the **IsGiftCertificate** field in the table. Check the checkbox if the item is a gift certificate.
- **May Apply To** corresponds to the **GiftCertificateUse** field in the table. Gift certificates may be applied to one or more of the order *SubTotal*, *Shipping*, or *Tax*.
- **Face Value** corresponds to the **GiftCertificateValue** field in the table. The value entered will be the monetary value of the gift certificate.

Electronic Asset Delivery (EAD)

Warning: This feature is pseudo-implemented. EAD works, but not quite to our satisfaction due to some browser compatibility issues. If it functions in a manner that is to the satisfaction of the store manager, then by all means use it.

- **Product requires EAD** corresponds to the **UseEAD** field in the table. Check the checkbox if the item will be distributed via electronic download.
- **File Name** corresponds to the **EADFileNames** field in the table. Type in the name of the file to be downloaded. EAD files are placed in the */hostedstore/EADFiles/* directory. If the file is in a subdirectory then be sure to include the path in the file name. For example, a file named *EAD.pdf* would be entered as *EAD.pdf* if the file is located at the root of the */hostedstore/EADFiles/* directory and entered as *downloads/EAD.pdf* if the file is located in a directory named *downloads/* in the */hostedstore/EADFiles/* directory.

At this time only one file name can be entered in this field. If more than one file must be distributed, then place all of the files in a directory, compress the directory (using a compression utility to stuff, tar, zip, etc), and then enter the name of the compressed file in the field. Note that it may be necessary to register the file extension for the compressed file in Lasso Administration on the **Format File Settings** pane of the **Global Settings > Settings** tab. If making modifications to these settings is unfamiliar, consult the Lasso documentation or seek the aid of the server administrator.

- **Content Type** corresponds to the **EADContentType** field in the table. Enter in the MIME settings for the file so that the customer's Web browser knows how to handle the file. If unknown, MIME settings for many document types can be found in one of the settings panes of the browser preferences. The location of settings varies between browsers so that will not be detailed here.

A list of MIME types common to downloaded files:

- *application/mac-binhex40* - BinHex (.hqx) compressed files.
- *application/mac-binhex40* - BinHexed StuffIt Archive (.sithqx) compressed files.
- *application/macbinary* - MacBinary (.bin) file.
- *application/msword* - Microsoft Word (.doc) files.
- *application/pdf* - PDF (.pdf) files.
- *application/vnd.ms-excel* - Microsoft Excel (.csv, .xl, .xls) files.
- *application/x-stuffit* - StuffIt Archive (.sit) compressed files.
- *application/x-tar* - Unix Tape Archive (.tar) compressed files.
- *application/zip* - Zip (.zip) compressed files.

- o *audio/mpeg3* - MP3 (.mp3) audio files.
 - o *image/gif* - GIF (.gif) image files.
 - o *image/jpeg* - JPEG (.jpg, .jpeg) image files.
 - o *video/ms-asf* - Microsoft Windows Media (.asf) files.
 - o *video/quicktime* - Quicktime (.mov) video files.
 - o *video/x-msvideo* - AVI (.avi) media files.
- **Download Limit** corresponds to the **EADDownloadLimit** field in the table. Sets the maximum number of times the file can be downloaded. Leave empty if unlimited downloads are allowed.
 - **Days Limit** corresponds to the **EADTimeLimit** field in the table. Sets the maximum number of days the customer has to download the file. Leave empty if there is not an expiration date.

EAD area of the add product form

Supplier Information

- **Supplier ID** corresponds to the **SupplierID** field in the table. This value should correspond to the ID for the supplier if one has been defined.
- **Email order notification** corresponds to the **NotifySupplier** field in the table. If checked, the supplier will be emailed notification when an order is ready for them to ship.

Content Page

- **File Name** corresponds to the **ContentPage** field in the table. If this field contains a value, then the file name specified will be used as the product detail content page instead of the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*). Note that the existence of this file will not be verified when the product is added to the database so products may be entered before the display code has been written. However, the existence of the file will be verified at runtime. If the file does not exist, then the default product detail content page will be used instead.

The file name should be specified from the root directory of */hostedstore/storefront/html/*. For example, if multiple product detail content pages are used for a store, all content pages are in a directory named *product_templates* in the */hostedstore/storefront/html/* directory, and the name of the content page to be used is *template_a.lasso*, then the value for **File Name** should be *product_templates/template_a.lasso*.

- **Page Title** corresponds to the **PageTitle** field in the table. If this field contains a value, then its value will be substituted between the `<title></title>` tags in the document head of the store template file for the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*). Note that this value will override the global value usually substituted that is determined by the *HS_PageTitle* preference variable set in *Store Configuration*.
- **Meta Keywords** corresponds to the **MetaKeywords** field in the table. If this field contains a value, then its value will be substituted into the *keywords* meta tag of the store template file for the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*). Note that this value will override the global value usually substituted that is determined by the *HS_MetaKeywords* preference variable set in *Store Configuration*.

- **Meta Description** corresponds to the **MetaDescription** field in the table. If this field contains a value, then its value will be substituted into the **description** meta tag of the store template file for the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*). Note that this value will override the global value usually substituted that is determined by the **HS_MetaDescription** preference variable set in *Store Configuration*.



The image shows a form titled "Content Page" with four input fields. The fields are labeled "File Name", "Page Title", "Meta Keywords", and "Meta Description". Each field is represented by a horizontal text box with a vertical scrollbar on the right side, indicating they are text input areas.

Content page area of the add product form

Product Descriptions

- **Brief description contains HTML** corresponds to the **BriefHasHTML** field in the table. Check the checkbox if the brief description for the product contains HTML tags.
Important: It is HIGHLY recommended to verify any HTML in a browser before using it in product brief descriptions. Invalid HTML could result in pages with abnormal properties.
- **Brief Description** corresponds to the **BriefDescription** field in the table. A product brief description is displayed with featured and favorite products on the default store home content page (*/hostedstore/storefront/html/index.lasso*), the default search results content page (*/hostedstore/storefront/html/search_list.lasso*), and the default email a friend content page (*/hostedstore/storefront/html/email_friend.lasso*).
- **Description contains HTML** corresponds to the **DescHasHTML** field in the table. Check the checkbox if the description for the product contains HTML tags.
Important: It is HIGHLY recommended to verify any HTML in a browser before using it in product descriptions. Invalid HTML could result in pages with abnormal properties.
- **Full Description** corresponds to the **Description** field in the table. A product description is displayed on the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*) and the default product modification content page for configurable products (*/hostedstore/storefront/html/cart_modifyitem.lasso*).

Product Descriptions

Important: It is **HIGHLY** recommended that you verify any HTML in a browser before using it in product descriptions. Invalid HTML could result in pages with abnormal properties.

Brief description contains HTML.

Brief Description

Description contains HTML.

Full Description

Product descriptions area of the add product form

Product Attributes

Products that may have multiple configurations can be set up as a single product with multiple attributes. For example, a t-shirt may have the same print design, but is offered in five different colors and three different sizes. Rather than create 15 individual t-shirt products (five colors times three sizes), a single t-shirt product can be created with color and size attributes. Product attributes can be used for very simple scenarios like the t-shirt example just presented or for very complex scenarios that combine several products, add or subtract from the base price, and adjust the inventory of other products.

Product attributes are a very powerful feature. A total of thirty attributes are available for any one product so complex products can be offered for customization. Product attributes are not limited to use with attributes that are specific to the product itself. Upsell items or enhancements are a perfect use for the attributes feature. For example, gift wrapping, warranties, and service plans can all be offered using product attributes.

Just as an explanation as to why certain design decisions are made, during research it was discovered that computers frequently require the most number of attributes during configuration. Twenty-five attributes was the maximum number found anywhere so thirty were built into HostedStore "just in case." If more attributes are needed send in a request and more can be added.

The fields listed below are repeated for all thirty attributes. Whenever **X** is used in one of the field names or explanations it generically refers to the appropriate attribute number: one through thirty.

- **Display as radio button** corresponds to the **AttributeXDescription** field in the table. Checking the checkbox will display the attribute value selections as radio button selections on the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*) and the default product modification content page for configurable products (*/hostedstore/storefront/html/cart_modifyitem.lasso*).
- **Name** corresponds to the **AttributeXName** field in the table. **Name** is the attribute name that is to be displayed to the customer. This field essentially provides a means to communicate the purpose of the attribute. For the t-shirt example above, the **Name** might be *Color* for the first attribute and *Size* for the second.
- **Description File** corresponds to the **AttributeXDescription** field in the table. If the attribute requires additional explanation, then this field can be used to associate a supplemental file with the attribute. For the t-shirt example above, it may be helpful to have an image associated with the color attribute that displays all five colors so the customer can visualize each choice.

Description File can have values of *gif*, *jpg*, *png*, *html*, or *lasso*. The value chosen determines the document file format and file name that is associated with the attribute. The description files must follow a standard naming convention: ProductID_AttributeNumber.Extension. For example, if the product ID is Sample01, the attribute number is 2, and the file is a .jpg image, then the file name would be Sample01_2.jpg. All description files must be placed in the */hostedstore/images/products/descriptions/* directory to work properly with the default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*) and the default product modification content page for configurable products (*/hostedstore/storefront/html/cart_modifyitem.lasso*).

- **Values** corresponds to the **AttributeXValues** field in the table. All of the possible selections for the attribute are listed in this field separated by two adjacent pipe characters (| |). The pipe character key is located in the shift position above the backslash (\) key (usually located above the return/enter key) on the keyboard. Double pipes were chosen as a delimiter because it is highly unlikely that double pipes would be used in an attribute value--in fact, do not do so because it will break the functionality of HostedStore!

Continuing with the t-shirt example, the customer will see a select list called Size with Medium, Large, and Extra Large as the choices when the attribute with a **Name** of *Size* has the following entered into its **Values** field:

Medium | | *Large* | | *Extra Large* (note that there are no spaces before and after the two pipe symbols).

Important: Each selection must be unique.

- **Prices** corresponds to the **AttributeXPrices** field in the table. The **Prices** field is used when different attribute value choices affect the price of the product. If the price is not affected by attribute values, then leave the field blank. As with attribute values, all price variations are separated by | |. Price increases are entered as a numeric value, price decreases are entered as a numeric value preceded by a minus sign (-), and no price change is left blank (note that blank does not mean a space character). For instance, for the t-shirt example if medium and large t-shirts are \$10.00 and extra-large t-shirts are \$12.00, then *10.00* would be entered in the **Price** field (the base price for the product) and the following would be entered into the **Prices** field for the size attribute: | | | | *2.00*

Note that the first choice (area to the left of the first | | sequence) is blank because the price does not vary for a medium sized t-shirt, the second choice (area to the right of the first | | sequence and to the left of the second | | sequence) is blank because the price does not vary for a large sized t-shirt, and the third choice contains 2.00 because the price for extra-large t-shirts is \$2.00 more than the base price.

- **Part Numbers** and **Part Quantities** correspond to the **AttributeXPartNumbers** and **AttributeXPartQuantities** fields in the table, respectively. These two fields are used in unison for products with attributes that involve other products. For example, a custom built computer may have attributes like CPU, hard drive, and keyboard, each of which could involve a different product. Decrementing the main product inventory will not properly track the change in inventory for the other products. The **Part Numbers** and **Part Quantities** fields are used for attribute related inventory management.

Important: **Part Numbers** and **Part Quantities** only affect inventory. If adding attributes affects the weight of the product then attention should be given to the proper pricing of the attribute to recoup additional costs of handling, shipping, etc. For example, if a computer has bundling attributes of monitor, printer, etc that allow the customer to bundle additional products to the base product then the overall shipping cost will likely be affected but would not be accounted for in the shipping cost for the base product.

Part Numbers is used for the product IDs of the products associated with the choices in the **Values** field separated by double pipe symbols. As with prices, each of the product IDs must map to the attribute choices in in the **Values** field. If one of the attribute choices does not affect an inventory item, then it should be left blank (note that blank does not mean a space character).

Part Quantities is used for the inventory adjustments to make for the products associated with the attribute choices. Probably guessed it by now, but each value must be separated by double pipe symbols and each must map to the attribute choices in in the **Values** field. If one of the attribute choices does not affect an inventory item, then it should be left blank (note that blank does not mean a space character).

Here is a simple example that combines all of the attribute fields. Suppose a customizable computer is offered that comes without a keyboard, but can have a standard keyboard (part number *KB123*) added for \$10. A picture of the keyboard is available as a JPEG image. This is what all of the attribute fields would look like:

Name*Keyboard*
Description File*jpg*

Values*No Keyboard | Standard Keyboard*
Prices | *10.00*
Part Numbers | *KB123*
Part Quantities | *1*

Important: If an attribute is created for a product, then a selection is required by the customer. To create an optional attribute within this framework, add an attribute value that does not affect pricing or inventory. An example of this was provided in the example above. The computer did not need to be purchased with a keyboard, but because a keyboard attribute was created a value of *No Keyboard* was added to satisfy the required attribute.

Multiple product IDs and quantities can be specified for an attribute value by separating the product IDs and quantities with a comma (,). The number of comma delimited product IDs and quantities must match for each value.

Here is a simple example that combines all of the attribute fields and uses multiple product IDs and quantities for an attribute value. Suppose a customizable computer is offered that comes without a printer, but can have a printer added for \$100. The printer is actually comprised of two parts: 1 part number *Printer1* for the printer itself and 1 part number *Cable1* for the cable necessary to attach the printer to the computer. A picture of the printer is available as a JPEG image. This is what all of the attribute fields would look like:

Name*Printer*
Description File*jpg*
Values*No | Yes*
Prices | *100.00*
Part Numbers | *Printer1, Cable1*
Part Quantities | *1, 1*

• Text Box Attributes

Using a keyword, two special case attributes can be defined that allow the customer to enter information into a text box. One keyword makes a required textbox (this is no surprise as all other attributes are required). The second keyword provides a textbox that is not required.

What is the purpose of these text boxes? Here are a couple of examples:

- Suppose monogrammed towels are sold in a store. The store staff would need to know what the customer wanted monogrammed on the towel. A required text box attribute would be ideal for this.
- Suppose one of the product attributes allows a customer to purchase gift wrapping service for the product. As part of the gift wrapping service, the store staff will place a message on the card for the customer if wanted. All customers may not want a message so an optional text box would be ideal for this.

The two types of text box attributes are detailed below:

- **Required Text Boxes** are specified by entering the keyword *Textbox* in the **Prices** field for the attribute. The **Name** field will be used as the label for the textbox. The **Values** field will be used as the initial text for the textbox, but is not required.

This is how the monogrammed towel example might look:

Name*Monogram*
Values*Please use this box to type the monogram exactly as it should appear on the towel.*
Prices*Textbox*

- **Optional Text Boxes** are specified by entering the keyword *OptionalTextbox* in the **Prices** field for the attribute. The **Name** field will be used as the label for the textbox. The **Values** field will be used as the initial text for the textbox, but is not required.

This is how the optional message example might look:

Name*Message for Card*
Values*Please use this box to type the message exactly as it should appear*

on the card.

Prices *Optional* *Textbox*

Important: When using text box attributes special care must be given to writing display code that determines what type of attribute is being displayed. The default product detail content page (*/hostedstore/storefront/html/product_detail.lasso*) and the default product modification content page for configurable products (*/hostedstore/storefront/html/cart_modifyitem.lasso*) both contain code for doing so. Use care when making modifications to those two default content pages.

The screenshot shows a form titled "Product Attributes" with a sub-section "Attribute 1". It includes a checkbox labeled "Display as radio button." Below this are two columns of input fields. The left column has "Name" (a text box), "Values" (a large text area), and "Part Numbers" (a text area). The right column has "Description File" (a dropdown menu), "Prices" (a large text area), and "Part Quantities" (a text area).

Product attributes area of the add product form

Find Product(s)

A search form is provided to search for existing products. If more than one match is found, then a list of all matching products will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching product is displayed in an edit product form. The fields for editing a product are the same as the fields available when adding a new product. Note that due to the size of the form and the number of fields there will be a slight delay as the form loads.

The fields that can be searched are listed below along with the operator used in the search:

- **ProductID** - Equals (eq)
- **ProductName** - Contains (cn)
- **Department** - Contains (cn)
- **Category** - Contains (cn)
- **SubCategory** - Contains (cn)
- **Featured** - Equals (eq)
- **OnSale** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.

The screenshot shows a search form with the following fields and controls:

- Product ID**: A text input field with a "Find Product(s)" button to its right.
- Product Name**: A text input field with a "list" button to its right.
- Department**: A text input field with a "list" button to its right.
- Category**: A text input field with a "list" button to its right.
- SubCategory**: A text input field with a "list" button to its right.
- Featured**: Radio buttons for "Yes" and "No".
- On Sale**: Radio buttons for "Yes" and "No".
- Created**: A date selection interface with dropdowns for Month, Day, and Year.
- Modified**: A date selection interface with dropdowns for Month, Day, and Year.
- Limit**: A text input field.

Product search form area of the product administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate product in the listing and click the **ProductID** hyperlink for the product. The product will be loaded into the edit form.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.



Product found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Three buttons are provided to perform functions on the found set. Each button corresponds to the same catalog-wide function.

Click the **Adjust Pricing** button to simultaneously adjust the pricing of the products in the found set. More details on pricing adjustment can be found in the [Store Administration](#) > [Product Administration](#) > [Adjust Pricing](#) section.

Click the **Adjust Inventory** button to simultaneously adjust the inventory of the products in the found set. More details on inventory adjustment can be found in the [Store Administration](#) > [Product Administration](#) > [Adjust Inventory](#) section.

Click the **Generate Static Product Pages** button to create static HTML pages for the products in the found set. More details on static pages can be found in the [Store Administration](#) > [Product Administration](#) > [Generate Static Product Pages](#) section.

Click on the product name to edit.

Adjust Pricing Adjust Inventory Generate Static Product Pages

Product ID▲▼	Product Name▲▼	Department▲▼	Category▲▼	SubCategory▲▼	Featured▲▼	On Sale▲▼
AMP3WATT	3 Watt Booster Amplifier		Cellular Accessories	Best Sellers	Y	Y
2000R35HM02	Aeron Chair	Furniture	Chairs		Y	Y
2000R35HM04	Aeron Side Chair	Furniture	Chairs		Y	Y
PentiumIV	Custom Built Pentium IV Computer	Electronics	Computers	Systems - Custom-Built	Y	
IBR-1	In-Building Repeater		Cellular Accessories	Best Sellers	Y	Y
onecent	One Cent Test Item	Test	Test		Y	
SGHN0000801	Permanent Hands Free Kit		Cellular Accessories	Permanent install hands-free kits	Y	

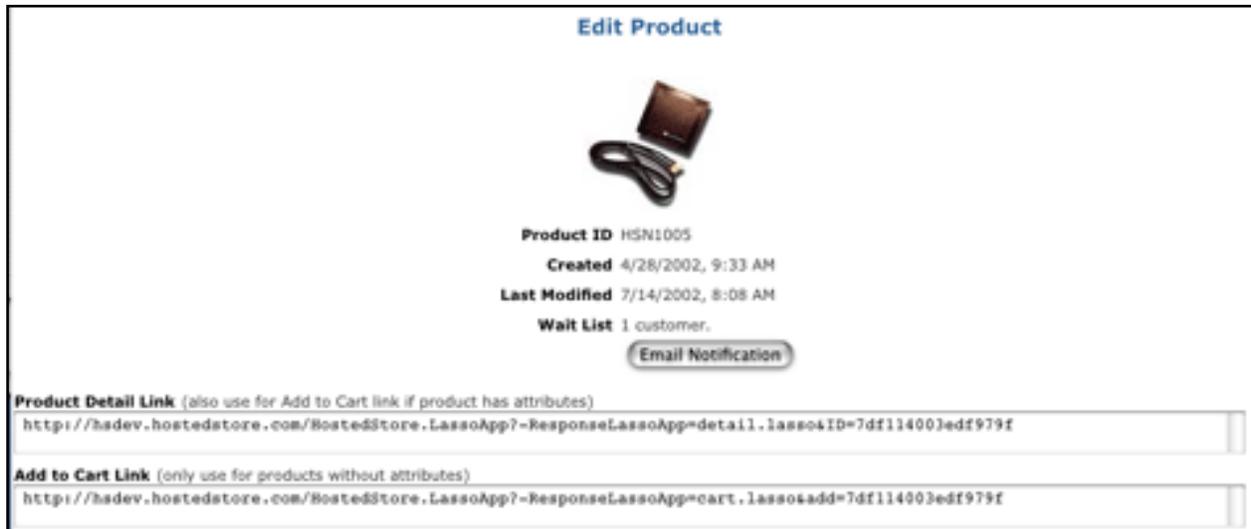
Product search results list

Edit Product

The edit product form is nearly identical to the add product form. An explanation of the fields on the edit product form can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.

If wait lists are used, the product summary information at the top of the page will display the number of customers currently in the wait list. Clicking the **Email Notification** button will email notification to the customers that the product is in stock (details about the notification email can be found in the [Modifying HostedStore](#) > [Email Messages](#) > [Store Administration](#) > [Product In Stock Notification](#) section). The wait list will be purged and the edit product form will be returned.

Important: Clicking the **Email Notification** button only processes the wait list code and then reloads the current product information from the database. If any changes are made to the product information be sure to update the product information before processing the wait list or the new information will be lost.



The screenshot shows the 'Edit Product' form. At the top, there is a title 'Edit Product' and an image of a product. Below the image, the following information is displayed: 'Product ID HSN1005', 'Created 4/28/2002, 9:33 AM', 'Last Modified 7/14/2002, 8:08 AM', and 'Wait List 1 customer.'. There is a button labeled 'Email Notification'. Below this, there are two text input fields: 'Product Detail Link (also use for Add to Cart link if product has attributes)' with the URL 'http://hdev.hostedstore.com/HostedStore.LassoApp?~ResponseLassoApp=detail.lasso&ID=7df114003edf979f' and 'Add to Cart Link (only use for products without attributes)' with the URL 'http://hdev.hostedstore.com/HostedStore.LassoApp?~ResponseLassoApp=cart.lasso&add=7df114003edf979f'.

Wait list in product summary area of the edit product form

An **Update Product** button and an **Update Product and Generate Static Product Page** button appear multiple times on the form. The **Update Product** button simply updates the product record in the table. The **Update Product and Generate Static Product Page** button updates the product record in the table and creates a static product detail page. If a static product detail page already exists for the product, then it will be overwritten with the new product information. Details about static pages can be found in the [Store Administration](#) > [Product Administration](#) > [Generate Static Product Pages](#) section. Clicking any of the buttons will submit the data for the entire form--there is no need to click the button for each section of the form.



The screenshot shows two buttons side-by-side: 'Update Product' and 'Update Product and Generate Static Product Page'.

Update product buttons on the edit product form

Using Products as Templates

To use the currently displayed product as a template for a new product, make the appropriate changes to the product information, enter a new unique product ID in the **Product ID** field at the bottom of the form, and click the **Save as New Product** button or the **Save as New Product and Generate Static Product Page** button. The **Save as New Product** button simply adds the product to the table. The **Save as New Product and Generate Static Product Page** button adds the product to the table and creates a static product detail page. Details about static pages can be found in the [Store Administration](#) > [Product Administration](#) > [Generate Static Product Pages](#) section. The product will be created if no errors were encountered. Otherwise the form will be reloaded with error messages displayed at the top of the page.



The screenshot shows the 'Product ID' field with a text input box, followed by two buttons: 'Save as New Product' and 'Save as New Product and Generate Static Product Page'.

Save as new product buttons on the edit product form

Adjust Pricing

Click the **Adjust Pricing** button to load a form containing every product and all of its pricing. Each price can be adjusted as appropriate.



Adjust pricing button on the product administration menu

Important: All prices should be specified with a decimal point (even if a different decimal delimiter is used for display) and without any currency symbols. The decimal delimiter and currency symbol used for display purposes are specified with the `HS_CurrencyDecimal` and `HS_CurrencySymbol` preference variables, respectively, in *Store Configuration*.

This feature is convenient if a catalog price change is necessary because each product does not need to be loaded individually. Note that for large product catalogs the time to load and submit this form may be long so be a little patient. The explanation for each of the form fields and its corresponding field in the **Product** table are detailed below. If the purpose for any of the fields is unfamiliar, details about each field can be found in the [Store Administration](#) > [Product Administration](#) > [Add New Product](#) section.

- **Sell** corresponds to the **WebSell** field in the table.
- **Featured** corresponds to the field of the same name in the table.
- **On Sale** corresponds to the **OnSale** field in the table.
- **MSRP** corresponds to the field of the same name in the table.
- **Price** corresponds to the field of the same name in the table.
- **Sale Price** corresponds to the **SalePrice** field in the table.
- **Member Price** corresponds to the **MemberPrice** field in the table.

Adjust Product Pricing

IMPORTANT: Do not include currency symbols in pricing.
Be sure to check for errors prior to submitting this form. If there are errors, all modifications will be lost.

Product ID ▲ ▼	Product Name ▲ ▼	Sell	Featured	On Sale	MSRP	Price	Sale Price	Member Price
NTN8397	Hard Leather Case w/Swivel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	55.00	44.55	<input type="text"/>	<input type="text"/>
RLN4608	Hard Leather Case w/Swivel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	42.00	33.75	<input type="text"/>	<input type="text"/>
RLN4917	Hard Lthr Case w/Swivel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	45.00	31.05	<input type="text"/>	<input type="text"/>
RLN4926	Case w/Swivel & Window	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16.50	10.13	<input type="text"/>	<input type="text"/>
RLN4927	Case w/Swivel & Cutouts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16.50	10.13	<input type="text"/>	<input type="text"/>
RLN4994	Leather Flip Case w/Swivel Clip	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12.00	9.11	<input type="text"/>	<input type="text"/>
RLN5050	Sharkskin Neoprene Case w/Swivel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15.00	9.11	<input type="text"/>	<input type="text"/>
RLN5334	Flip Style Case w/Swivel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	9.99	7.76	<input type="text"/>	<input type="text"/>

Adjust product pricing form

Adjust Inventory

Click the **Adjust Inventory** button to load a form containing every product and all of its inventory information.



Adjust inventory button on the product administration menu

Each product's **OnHand** field in the table can be adjusted as appropriate. This feature is convenient for entering new inventory when it is received from suppliers.

Adjust Inventory

Click on the Product ID to edit the product.

Current inventory can be modified for all products by adjusting the "On Hand" amount and clicking the "Update Inventory" button at the bottom.

Product ID ▲▼	Product Name ▲▼	Sold ▲▼	On Hand ▲▼	Build To ▲▼
NTN8397	Hard Leather Case w/Swivel		<input type="text" value="10000"/>	
RLN4608	Hard Leather Case w/Swivel		<input type="text" value="10000"/>	
RLN4917	Hard Lthr Case w/Swivel		<input type="text" value="10000"/>	
RLN4926	Case w/Swivel & Window		<input type="text" value="10000"/>	
RLN4927	Case w/Swivel & Cutouts	3	<input type="text" value="9998"/>	
RLN4994	Leather Flip Case w/Swivel Clip		<input type="text" value="10000"/>	
RLN5050	Sharkskin Neoprene Case w/Swivel		<input type="text" value="10000"/>	
RLN5334	Flip Style Case w/Swivel		<input type="text" value="10000"/>	

Adjust product inventory form

Generate Static Product Pages

Click the **Generate Static Product Pages** button to create an HTML page for each product that has a **WebSell** field value of γ . The HTML pages are created in the `/hostedstore/products/` directory. The **ProductID** is used for the file name with an extension of `html`. For example, a product with an ID of `DemoProduct` will be created as an HTML page named `DemoProduct.html` in the `/hostedstore/products/` directory.



Generate static product pages button on the product administration menu

If there are a lot of products in the table, then this process could last several minutes or hours. Once the button has been clicked, Lasso will begin creating the pages. Feel free to open another browser window and continue work in other areas of *Store Administration* rather than waiting for the success page to be returned. Our test database of over 2300 products took about an hour to generate all of the pages.

Important: File permissions must be set properly for the pages to be generated. If using Mac OS X system operating software, the `/hostedstore/products/` directory should be owned by `lasso` with `Write` and `Read` permissions granted.

This feature is useful for creating pages for search engines that have trouble following dynamically created pages. Static pages can also be used to eliminate complex hyperlinks when linking to a specific product detail page. There are other approaches to this problem that are beyond the scope of this documentation. Articles explaining other methods may be available in the [support area](#) of the HostedStore Web site.

Global Price Adjustment

This option will adjust the prices of all the chosen products by the same monetary amount or percentage. Note that this is different than the *Adjust Pricing* option described in the [Store Administration](#) > [Product Administration](#) > [Adjust Pricing](#) section because the same adjustment is applied to all of the selected prices rather than allowing for individual price adjustments to be made. This feature is provided for situations where, for example, all store products on sale are discounted by an additional 10% for a week or if prices needs to be increased by 5% for all member only items.

To globally adjust prices select the proper values for each of the fields on the form and click the **Adjust Prices** button. Each price can be adjusted as appropriate.

A selection is required for each field. The purpose of each form field and its corresponding field in the **Product** table are detailed below.

- **Products** designates to which set of products the price adjustment is to apply.
 - *All* - All products.
 - *Published* - Products currently eligible for sale in the store. This corresponds to all products with a **WebSell** field value of *Y*.
 - *Featured* - Products flagged as featured items. This corresponds to all products with a **Featured** field value of *Y*.
 - *On Sale* - Products flagged as on sale. This corresponds to all products with an **OnSale** field value of *Y*.
 - *Member* - Products that are available for purchase by customers with a current membership. This corresponds to all products with a **MemberOnlyItem** field value of *Y*.
- **Price(s) to Adjust** designates the price(s) to which the price adjustment is to apply.
 - *MSRP* corresponds to the field of the same name in the table.
 - *Price* corresponds to the field of the same name in the table.
 - *Sale* corresponds to the **SalePrice** field in the table.
 - *Member* corresponds to the **MemberPrice** field in the table.
- **Adjustment** sets the actual adjustment value.
 - All prices should be specified with a decimal point (even if a different decimal delimiter is used for display) and without any currency symbols. The value will be added to the designated price. For example, to decrease the price by 5.75 monetary units, the appropriate value to enter is *-5.75*.
 - All percentage values should be expressed as a decimal value. Prices are multiplied by the percentage to determine the new price. Do not use a negative value to decrease a price by a percentage amount. The appropriate decimal value must be used instead. For example, to decrease prices by 20%, the appropriate value to enter is *.8* ($1.00 - .20$), not *-.20*.
- The select list is used to specify if the adjustment value is to be interpreted as *Monetary Units* or a *Percentage*.

Global Price Adjustment

Products [Dropdown Menu] **Adjust Prices**

Price(s) to Adjust

- MSRP
- Price
- Sale
- Member

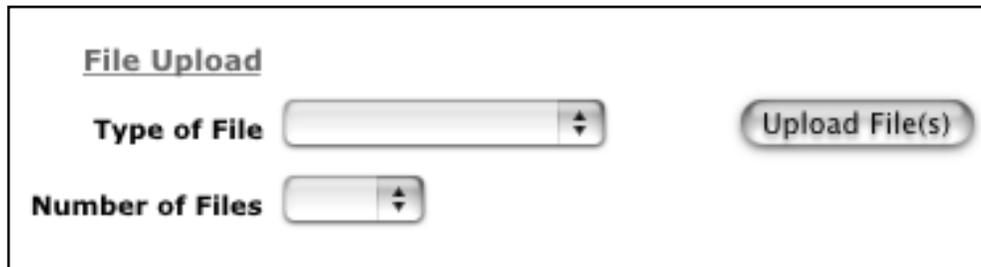
Adjustment [Text Input] [Dropdown Menu]

Global price adjustment form

File Upload

This feature is convenient for quickly uploading product images and/or description files for new products. If many products are entered at once (for example, during initial store setup), it is recommended to use another file transfer mechanism more suited for large quantities of files. The steps for uploading files are detailed below:

1. Select which type of file is being uploaded from the **Type of File** select list. The directory on the server for each selection is detailed below:
 - o *Description File* uploads to the `/hostedstore/images/products/descriptions/` directory.
 - o *Thumbnail Image* uploads to the `/hostedstore/images/products/thumbnail/` directory.
 - o *Detail Image* uploads to the `/hostedstore/images/products/detail/` directory.
 - o *Zoom Image* uploads to the `/hostedstore/images/products/zoom/` directory.
2. Select the appropriate number of files from the **Number of Files** select list.
3. Click the **Upload File(s)** button. A new form will be loaded with a number of file selection boxes corresponding to the value chosen from the **Number of Files** select list above.



The screenshot shows a form titled "File Upload". It contains two dropdown menus: "Type of File" and "Number of Files". To the right of the "Type of File" dropdown is a button labeled "Upload File(s)".

Upload files form area of the product administration menu

To select and upload the files:

1. Using the **Browse...** button for each file selection box, find and select the file from the hard drive.
2. Click the **Upload** button. If all goes well the main *Product Administration* menu will appear with a message that each file was uploaded successfully. Otherwise, if an error is encountered then it will be reported.



The screenshot shows a form titled "Upload Images". It contains a text box with the instruction "Upload files to the `/hostedstore/images/products/thumbnail/` directory." Below this are three rows, each with a file selection box labeled "File 1", "File 2", and "File 3" respectively, and a "Browse..." button to the right of each box. At the bottom of the form is an "Upload" button.

Upload files form for three thumbnail images

Promotion Administration

The *Promotion Administration* menu provides for two different operations: Add New Promotion, and Find Promotion(s). For any field requiring a monetary value, the value should be expressed as a decimal value without currency symbols. Also, be sure to use a decimal point (.) as the currency delimiter even if a different currency delimiter, a comma (,) for instance, is used for display purposes. For any field requiring a percentage value, the value should be expressed as a decimal without the percent symbol. For example, 15% would be expressed as .15.

Promotion Administration Menu

Promotion ID

Promotion ID

Promotion Name

Active Yes No

Begin Date

End Date

Type

Operator And Or

Limit

Promotion administration menu

Sections

- [Add New Promotion](#) - Create a new promotion.
- [Find Promotion\(s\)](#) - Search for and edit existing promotions.
- [Example Uses for Promotions](#) - Some suggestions for how to use promotions.

Add New Promotion

Create a new promotion by selecting a unique promotion code, entering it in the **Promotion Code** field, and clicking the **Add New Promotion** button. The system will verify that the promotion code selected is unique. If not, a message will prompt for the selection of a new promotion code.



Promotion code area of the add promotion form

Add Promotion Form

The top of the add promotion form displays the chosen promotion code. All of the fields on this form correspond to fields in the **Promotion** table only. An explanation of each field and its corresponding field in the **Promotion** table are detailed by group below.



Promotion code area of the add promotion form

After all fields have been completed, click the **Create Promotion** button at the bottom of the form to add the promotion to the database.



Create promotion button on the add promotion form

Description

Two parameters are provided to note information about the promotion for internal use.

- **Promotion Name** corresponds to the **PromotionName** field in the table. **Promotion Name** is used to provide additional information about what the promotion is. This information is not used by the promotion system internally, it is meant solely for administration purposes as a reminder of the purpose of the promotion.
- **Description** corresponds to the field of the same name in the table. **Description** is used to provide additional information about the purpose of the promotion. This information is not used by the promotion system internally, it is meant solely for administration purposes as a reminder of the purpose of the promotion.

The image shows a screenshot of a web form for adding a promotion. It features two main input areas: a single-line text box for 'Promotion Name' and a larger multi-line text area for 'Description'. The labels are positioned to the left of their respective input fields.

Description area of the add promotion form

Parameters

Multiple parameters must be set for the promotion so that the system can determine when a promotion should be applied to an order.

- **Activate Promotion** corresponds to the **Active** field in the table. **Activate Promotion** sets whether or not the promotion is ready for use (*REQUIRED*). If set to *Yes*, then the promotion can be applied to an order if all other parameters and restrictions are met.
- **Begin Date** and **End Date** correspond to the **Begin Date** and **End Date** fields in the table, respectively. **Begin Date** and **End Date** are used to set the date time frame to which the promotion can be applied (*REQUIRED*). Promotions begin the first hour of the beginning date and end the first hour of the ending date. If the promotion is meant to last indefinitely, then set the **End Date** to a date far into the future.
- **Apply To** corresponds to the **ApplyTo** field in the table. **Apply To** specifies to what order value the promotion is applied (*REQUIRED*). Selecting *Product* will apply the promotion to a specific product (specified in the *Promotion Restrictions* area), *Shipping* applies the promotion to the shipping fee, and *SubTotal* applies the promotion to the subtotal of the order.
- **Type** corresponds to the field of the same name in the table. **Type** specifies if the promotional discount will be a *Fixed Value* monetary value or a *Percentage* of the product price, shipping fee, or subtotal (determined by the value set for **Apply To**) (*REQUIRED*).
- **Value** corresponds to the field of the same name in the table. **Value** sets the promotional discount amount (*REQUIRED*). Use either a fixed monetary value or a percentage. Percentages are expressed as a decimal value. For example, 15% is expressed as .15 and 100% is expressed as 1.00.

Parameters

*** Activate Promotion** Yes No

*** Begin Date**

*** End Date**

*** Apply To**

Type

IMPORTANT: Do not include currency symbols in prices.
Express percentages as decimals. (ex. 15% is .15)

*** Value**

Parameters area of the add promotion form

Restrictions

Restrictions can be set so that promotions are only applied when the restricting criteria are met. One or several restrictions can be set for a single promotion. All of the restrictions are optional.

- **Match Customer ID** corresponds to the **MatchCustomerID** field in the table. **Match Customer ID** can be used to allow only one customer to use a promotion code. Enter the appropriate customer ID (**ID** field in the **Customer** table) for the customer.
- **Match Product ID** corresponds to the **MatchProductID** field in the table. **Match Product ID** applies the promotion to one specific product. Enter the appropriate product ID (**ID** field in the **Product** table) for the product.
- **Minimum Items** corresponds to the **MinItems** field in the table. **Minimum Items** specifies the minimum number of items that must be in an order for the promotion to apply. Enter the appropriate integer value.
- **Minimum Shipping** corresponds to the **MinShipping** field in the table. **Minimum Shipping** specifies the minimum shipping fee for an order before the promotion is applied. Enter the appropriate decimal value.
- **Minimum SubTotal** corresponds to the **MinSubTotal** field in the table. **Minimum SubTotal** sets the required minimum subtotal for an order before the promotion is applied. Enter the appropriate decimal value.
- **Per Customer Limit** corresponds to the **PerCustomerLimit** field in the table. **Per Customer Limit** sets the maximum number of times a customer can use the promotion code for an order (an order is counted as an order that has been successfully completed, i.e., its **Status** field is not *In Cart* or *Canceled*). Enter the appropriate integer value.
- **Maximum Discount** corresponds to the **MaximumDiscount** field in the table. **Maximum Discount** sets the maximum monetary value for a promotional discount for an order. For example, if a promotion offers 10% off the order subtotal with a maximum discount of \$50, then a customer that orders \$1000 in merchandise will receive a \$50 discount instead of \$100. Enter the appropriate decimal value.

Restrictions

Match Customer ID

Match Product ID

Minimum Items

Minimum Shipping

Minimum SubTotal

Per Customer Limit

Maximum Discount

Restrictions area of the add promotion form

Find Promotion(s)

A search form is provided to search for existing promotions. If more than one match is found, then a listing of all matching promotions will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching promotion is displayed in an edit promotion form. The fields for editing a promotion are the same as the fields available when adding a new promotion.

The fields that can be searched are listed below along with the operator used in the search:

- **PromotionID** - Equals (eq)
- **PromotionName** - Contains (cn)
- **Active** - Equals (eq)
- **BeginDate** - Greater Than or Equals (gte)
- **EndDate** - Less Than or Equals (lte)
- **Type** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The screenshot shows a search form with the following fields and controls:

- Promotion ID**: A text input field.
- Promotion Name**: A text input field.
- Active**: Radio buttons for Yes and No.
- Begin Date**: Three dropdown menus for Month, Day, and Year.
- End Date**: Three dropdown menus for Month, Day, and Year.
- Type**: A dropdown menu.
- Operator**: Radio buttons for And (selected) and Or.
- Limit**: A text input field.
- Find Promotion(s)**: A button to execute the search.

Promotion search form area of the promotion administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate promotion in the listing and click the **Promotion Code** hyperlink for the promotion. The promotion will be loaded into the edit form.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.

Promotion found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the promotion code to edit.

Promotion Code▲▼	Promotion Name'▲▼	Active▲▼	Begin Date▲▼	End Date▲▼	Type▲▼	Value▲▼
ASC20P	Aeron Side Chair December Promotion	Y	12/1/2001	12/15/2001	Percentage	0.20
testing	Aeron Side Chair December Promotion	Y	12/1/2001	12/15/2002	Percentage	0.20
HSFS090102	Free shipping during summer	Y	6/1/2002	8/31/2002	Percentage	1.00
HSFS122401	Holiday free shipping promotion	Y	12/1/2001	12/31/2001	Percentage	1.00
March20PercentOff	March Discount Promotion	Y	3/1/2002	3/31/2002	Percentage	0.20

Promotion search results list

Edit Promotion

The edit promotion form is nearly identical to the add promotion form. See the [Store Administration](#) > [Promotion Administration](#) > [Add New Promotion](#) section for an explanation of each of the fields on the form.

Using Promotions as Templates

To use the currently displayed promotion as a template for a new promotion, make the appropriate changes to the promotion information, enter a new unique promotion code in the **Promotion Code** field at the bottom of the form, and click the **Save as New Promotion** button. The promotion will be created if no errors were encountered. Otherwise the form will be reloaded with error messages displayed at the top of the page.

The image shows a section of a form with a label 'Promotion Code' followed by a text input field. To the right of the input field is a button labeled 'Save as New Promotion'.

Save as new promotion button on the edit promotion form

Example Uses for Promotions

The promotion system is very flexible and can be used in creative ways to address many needs. Here are some examples of its uses:

- Storewide sales - Suppose that all customers are to receive 10% off their order for one week. Create a promotion with the appropriate values and then provide a note on the Web site to enter the appropriate code on the checkout content page. Or to avoid possible oversight of the special, force the promotion by setting the promotion code on the checkout page so it is automatically entered for the customer.
- Thank you emails - Reward customers that have purchased x times by sending them an email that contains a hyperlink to the store with the **PromotionID** in the URL. Pass the **PromotionID** as the value for the promotion parameter set with the `HS_PromotionIDName` preference variable in *Store Configuration*. If the entry point will be the store home content page, the `HS_PromotionIDName` preference variable is set to `PID`, and the `PromotionID` is `DecemberFreeShipping`, then the hyperlink would look like:
`http://www.yourserver.here/HostedStore.LassoApp?PID=DecemberFreeShipping`, where `www.yourserver.here` is replaced by the fully qualified domain name associated with the HostedStore license.
- Compensation for Complaints/Suggestions - Suppose a customer is to be compensated for a service complaint or rewarded for an outstanding suggestion they provided. Create a promotion with the appropriate monetary/percentage value that is assigned to their **CustomerID** with a **Per Customer Limit** of `1`. Either provide them with just the **PromotionID** or email a hyperlink that sets the promotion code for them automatically (see the suggestion above for details).
- Gift Certificates - Promotions do not provide a true gift certificate system that allows for orders to deplete the gift certificate value until the remaining value reaches zero. However, promotions can certainly be used for one-time use gift certificates. If the customer does not use the entire value, then the remaining value is forfeited.
- Free shipping - Offering free shipping for orders of a minimum monetary value is popular with online stores especially during the holiday season. Such a promotion can be created easily by setting **Apply To** to `Shipping` and setting the appropriate **Minimum SubTotal** or **Minimum Items** value.
- Buy One, Get One Free - Set **Match Product ID** to the **ProductID** of the item and set **Minimum Items** to `2`. If this special is to be limited to one per customer, set **Per Customer Limit** to `1`. A note about the special and the appropriate code to enter can be provided on the store home content page, via email to customers, on the product detail content page, etc. Note that using promotions for this purpose will not guarantee that the customer receives the special--like all promotions, the promotion code (**PromotionID**) must be entered on the checkout content page or set via a URL. In addition, this type of promotion will not be applied multiple times because multiple promotions are not supported at this time. For example, if a customer purchases four items that are "buy one, get one free" then the promotion will only be applied once so they will pay for three and receive one free. If multiple applications of such a discount is necessary, then a special product can be created that depletes two items from inventory instead of one for each item purchased.

Reports

Several reports are provided to help provide information about store activity. More reports can be added upon request. It is also possible to create a module that will generate any needed reports.

Date dependent reports require a beginning and ending date. If a report is needed for just one day, then set the beginning and ending date to that day. Check the **Display print version** checkbox to remove the *Store Administration* navigation from the report. Specify a file name in the **Export To** field to save a tab delimited export of the report to the */hostedstore/adminstore/report/* directory.

Date independent reports use the current state of the store as input. Some of the reports like *Missing Images* are intended to provide maintenance type information, whereas reports like *Search to Sales Conversion* provide an overview of general store statistics.

Most of the reports only provide data when it is relevant. For example, The *Sales by Product* date dependent product sales report only lists products that were sold during the time period. Depending upon the amount of data returned, the reports may take a substantial amount of time to generate. For example, the *View Image Library* date independent report will take less time if the store catalog contains 100 items than if the catalog contain 10,000 items.

Important: The reports that involve sales data are not meant to replace a more comprehensive accounting package. Reports are provided solely as a convenience to gain an overview of store operations, but should not be interpreted as an accurate depiction of the financial aspects of store operations.

Report Menu

Date Dependent Reports

Report

Begin Date

End Date

Display print version.

Export To

Date Independent Reports

Report administration menu

Sections

- [Date Dependent Reports](#) - Reports that require a date range.
- [Date Independent Reports](#) - Reports based on the current date.

Date Dependent Reports

All date dependent reports are generated by choosing the name of the report from the select list, specifying a beginning date and an ending date, and clicking the **View Report** button. Check the **Display print version** checkbox to remove the store navigation menu from the report. Specify a file name in the **Export To** field to save a tab delimited export of the report to the */hostedstore/adminstore/report/* directory.



The screenshot shows a web form titled "Date Dependent Reports". It contains the following elements:

- A "Report" dropdown menu with a "View Report" button to its right.
- "Begin Date" section with three dropdown menus for "Month", "Day", and "Year".
- "End Date" section with three dropdown menus for "Month", "Day", and "Year".
- A checkbox labeled "Display print version." which is currently unchecked.
- An "Export To" text input field.

Date dependent report form area of the report administration menu

Sections

- [Sales Order Reports](#)
- [Gross Income Reports](#)
- [Order Status Reports](#)
- [Product Sales Reports](#)
- [Sales Tax Reports](#)
- [Discount Reports](#)
- [Affiliate Reports](#)
- [Shipping Reports](#)

Sales Orders Reports

Reports that provide a summary of all orders that have a status which indicates the order has already or may eventually be counted as revenue. Specifically, sales orders reports are generated based on orders where:

- The **DatePosted** field for the order in the **Invoice** table has a value in the specified date range.
- The order status field, **Status**, for the order in the **Invoice** table does not have a value of *In Cart*, *Abandoned*, *Declined*, or *Canceled*.
- The **OrderStatus** field for each ordered item in the **Item** table does not have a value of *In Cart*, *Abandoned*, *Declined*, or *Canceled*.

Each report has an **Items** column which sums the **Quantity** field for each of the ordered items with a qualifying **OrderStatus** field value. The **SubTotal** column is the sum of the **SubTotal** field values for each order.

Important: Because these reports use the order **SubTotal** field value from the **Invoice** table, the sales total will not reflect any canceled items or price adjustments. These reports are provided only as a quick summary, whereas the *Gross Income Reports* provide a more accurate view albeit with different search criteria.

Orders by Day

The order subtotal and number of ordered items are summed for each day in the specified date range and displayed chronologically. Only the days that have orders appear in the report. Each day's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining weekly, monthly, and quarterly order volume, relative order volume between days, etc.

Orders by Day					
7/15/2002 through 7/31/2002					
Total Sales: \$26,937.29					
<u>Date</u>	<u>Day</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
7/22/2002	Mon	3	23	\$2,470.35	9.17%
7/23/2002	Tue	1	2	\$3,399.91	12.62%
7/24/2002	Wed	2	20	\$3,670.66	13.63%
7/27/2002	Sat	4	6	\$456.85	1.70%
7/29/2002	Mon	1	1	\$8,357.76	31.03%
7/30/2002	Tue	1	1	\$8,581.76	31.86%
Totals		12	53	\$26,937.29	

Orders by day report

Orders by Hour

The order subtotal and number of ordered items are summed for each one hour time period for each day in the specified date range and displayed chronologically. Only the one hour time periods that have orders appear in the report. Each hour's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining relative order volume between hours in a day or hours from day to day.

Orders by Hour						
7/15/2002 through 7/31/2002						
Total Sales: \$26,937.29						
<u>Date</u>	<u>Day</u>	<u>Time</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
7/22/2002	Mon	9:00 AM - 10:00 AM	2	3	\$1,079.85	4.01%
7/23/2002	Tue	8:00 AM - 9:00 AM	1	2	\$3,399.91	12.62%
7/23/2002	Tue	10:00 AM - 11:00 AM	4	6	\$456.85	1.70%
7/24/2002	Wed	2:00 AM - 3:00 AM	1	10	\$3,490.00	12.96%
7/24/2002	Wed	4:00 PM - 5:00 PM	1	10	\$180.66	0.67%
7/29/2002	Mon	11:00 AM - 12:00 PM	1	20	\$1,390.50	5.16%
7/29/2002	Mon	9:00 PM - 10:00 PM	1	1	\$8,357.76	31.03%
7/30/2002	Tue	11:00 AM - 12:00 PM	1	1	\$8,581.76	31.86%
Totals			12	53	\$26,937.29	

Orders by hour report

Orders by Product

The ordered item total price and number of ordered items are summed for each product ordered during the specified date range and displayed alphabetically by product ID. Only products that have been ordered appear in the report. Each product's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between products during a date range.

Orders by Product						
7/15/2002 through 7/31/2002						
Total Sales: \$26,937.29						
<u>ProductID</u>	<u>Product Name</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>	
98274	Adaptable Dual Pocket Desktop Charger	1	1	\$29.62	0.12%	
AMP3WATT	3 Watt Booster Amplifier	1	1	\$199.95	0.81%	
BC2500	Belt Clip	1	1	\$10.10	0.04%	
BC9100	Belt Clip	1	10	\$40.50	0.16%	
BP2	"All'armi, pensieri"	1	1	\$30.00	0.12%	
BWLassoStudio	Lasso Studio	1	10	\$3,490.00	14.11%	
CMP316MMC	16 MB MMC Chip	1	1	\$35.00	0.14%	
DTH1100	Desktop Holder	1	3	\$48.60	0.20%	
DTH3	Desktop Holder	1	4	\$75.60	0.31%	
IBR-1	In-Building Repeater	3	3	\$1,209.89	4.89%	
LargeCustomPizza	Large Pizza	1	1	\$28.15	0.11%	
NKN6482	PC Data Cable	1	10	\$1,350.00	5.46%	
NTN8011	Battery Belt Clip w/Insert	1	1	\$12.15	0.05%	
PCP9100AM	Amethyst Faceplate	1	2	\$21.46	0.09%	
PentiumIII	Custom Built Intel Celeron/PIII Computer	2	2	\$11,729.52	47.42%	
PentiumIV	Custom Built Pentium IV Computer	1	1	\$8,581.76	34.69%	
SGCD0015801	Dual Port Docking Station	1	1	\$44.99	0.18%	
			Totals	53	\$26,937.29	

Orders by product report

Orders by Department

The ordered item total price and number of ordered items are summed for each product department during the specified date range and displayed alphabetically by product department. Only product departments that have had products ordered appear in the report. Each product department's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product departments during a date range.

Orders by Department				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<hr/>				
<u>Department</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
Electronics	12	51	\$26,879.14	99.78%
Music	1	1	\$30.00	0.11%
Restaurant	1	1	\$28.15	0.10%
		<hr/>	<hr/>	
	Totals	53	\$26,937.29	
<hr/>				

Orders by department report

Orders by Manufacturer

The ordered item total price and number of ordered items are summed for each product manufacturer during the specified date range and displayed alphabetically by product manufacturer. Only product manufacturers that have had products ordered appear in the report. Each product manufacturer's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product manufacturers during a date range.

Orders by Manufacturer				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>Manufacturer</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
AudioVox	4	22	\$259.41	0.96%
BIM	1	1	\$30.00	0.11%
Blue World Communications, Inc.	1	10	\$3,490.00	12.96%
HostedStore Corporation	3	3	\$20,311.28	75.40%
LG	1	1	\$44.99	0.17%
Motorola	1	1	\$29.62	0.11%
Nextel	4	15	\$2,771.99	10.29%
	Totals	53	\$26,937.29	

Orders by manufacturer report

Orders by Customer

The order subtotal and number of ordered items are summed for each customer with orders during the specified date range and displayed alphabetically by customer name. Only the customers that have orders appear in the report. Each customer's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining which customers contribute the most to sales during a time period.

Orders by Customer					
7/15/2002 through 7/31/2002					
Total Sales: \$26,937.29					
<u>ID</u>	<u>Customer</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
15	Doe, Jane	9	50	\$18,094.59	67.17%
67	Doe, Jeremy	1	1	\$8,357.76	31.03%
63	Doe, Jimmy	1	1	\$439.95	1.63%
52	Doe, John	1	1	\$44.99	0.17%
Totals		12	53	\$26,937.29	

Orders by customer report

Orders by New Customers

The order subtotal and number of ordered items are summed for each new customer with only one order during the specified date range and displayed alphabetically by customer name. Only the customers that have placed their first and only order appear in the report. Each customer's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining which sales during a time period that can be attributed to new customers.

Orders by New Customers						
7/15/2002 through 7/31/2002						
Total Sales: \$8,842.70						
<hr/>						
ID	Customer	Orders	Items	SubTotal	Percentage	
67	Doe, Jeremy	1	1	\$8,357.76	94.52%	
63	Doe, Jimmy	1	1	\$439.95	4.98%	
52	Doe, John	1	1	\$44.99	0.51%	
		<hr/>	<hr/>	<hr/>		
Totals		3	3	\$8,842.70		
<hr/>						

Orders by new customers report

Orders by Return Customers

The order subtotal and number of ordered items are summed for each return customer that has placed an order before or placed multiple orders during the specified date range and displayed alphabetically by customer name. Only the customers that have placed a prior order appear in the report. Each customer's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the sales during a time period that can be attributed to repeat customers.

Orders by Return Customers					
7/15/2002 through 7/31/2002					
Total Sales: \$18,094.59					
<hr/>					
<u>ID</u>	<u>Customer</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
15	Doe, Jane	9	50	\$18,094.59	100.00%
Totals		9	50	\$18,094.59	

Orders by return customers report

Orders by Billing Country

The order subtotal and number of ordered items are summed for each country from which a customer has ordered during the specified date range and displayed alphabetically by country ID. Only the countries with customer orders appear in the report. Each country's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic origin of sales during a time period.

Orders by Billing Country				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>Country</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
CA	1	1	\$8,357.76	31.03%
US	11	52	\$18,579.53	68.97%
Totals	12	53	\$26,937.29	

Orders by billing country report

Orders by Billing State/Province

The order subtotal and number of ordered items are summed for each state/province from which a customer has ordered during the specified date range and displayed alphabetically by state/province ID. Only the states/provinces with customer orders appear in the report. Each state's/province's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic origin of sales during a time period.

Orders by Billing State/Province				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>State/Province</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
AL	1	1	\$439.95	1.63%
BC	1	1	\$8,357.76	31.03%
WA	10	51	\$18,139.58	67.34%
Totals	12	53	\$26,937.29	

Orders by billing state/province report

Orders by Billing Country, State/Province

The order subtotal and number of ordered items are summed for each country and itemized by the state/province from which a customer has ordered during the specified date range and displayed alphabetically by country ID and state/province ID. Only the countries and states/provinces with customer orders appear in the report. Each country's and each itemized state's/province's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic origin of sales during a time period.

Orders by Billing Country, State/Province				
7/15/2002 through 7/31/2002				
Total Sales: \$25,607.45				
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	
CA	1	1	\$8,357.76	26.35%
BC	1	1	\$8,357.76	26.35%
US	13	38	\$17,249.69	73.65%
AL	1	1	\$439.95	1.29%
WA	12	37	\$16,809.74	72.35%
Totals	14	39	\$25,607.45	

Orders by billing country, state/province report

Orders by Shipping Country

The order subtotal and number of ordered items are summed for each country to which a customer has shipped an order during the specified date range and displayed alphabetically by country ID. Only the shipping destination countries with customer orders appear in the report. Each country's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic destination of sales during a time period.

Orders by Shipping Country				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>Country</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
CA	1	1	\$8,357.76	31.03%
US	11	52	\$18,579.53	68.97%
Totals	12	53	\$26,937.29	

Orders by shipping country report

Orders by Shipping State/Province

The order subtotal and number of ordered items are summed for each state/province to which a customer has shipped an order during the specified date range and displayed alphabetically by state/province ID. Only the shipping destination states/provinces with customer orders appear in the report. Each state's/province's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic destination of sales during a time period.

Orders by Shipping State/Province				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>State/Province</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
AL	1	1	\$439.95	1.63%
BC	1	1	\$8,357.76	31.03%
WA	10	51	\$18,139.58	67.34%
Totals	12	53	\$26,937.29	

Orders by shipping state/province report

Orders by Shipping Country, State/Province

The order subtotal and number of ordered items are summed for each country and itemized by the state/province to which a customer has shipped an order during the specified date range and displayed alphabetically by state/province ID. Only the shipping destination countries and states/provinces with customer orders appear in the report. Each country's and each state's/province's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the geographic destination of sales during a time period.

Orders by Shipping Country, State/Province				
7/15/2002 through 7/31/2002				
Total Sales: \$25,607.45				
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	
CA	1	1	\$8,357.76	26.35%
BC	1	1	\$8,357.76	26.35%
US	13	38	\$17,249.69	73.65%
AL	1	1	\$439.95	1.29%
WA	12	37	\$16,809.74	72.35%
Totals	14	39	\$25,607.45	

Orders by shipping country, state/province report

Orders by Source

The order subtotal and number of ordered items are summed for each order source in the specified date range and displayed alphabetically by source name. Only the order sources that have orders appear in the report. Each source's summed order subtotal is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the relative order volume between sources.

Orders by Source				
7/15/2000 through 7/31/2002				
Total Sales: \$94,481.02				
<u>Source</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
Email	4	4	(\$1,819.98)	-1.93%
Phone	1	2	(\$3,124.94)	-3.31%
Retail Store	7	12	\$101.12	0.11%
Web	38	126	\$99,324.82	105.13%
Totals	50	144	\$94,481.02	

Orders by source report

Orders by Status

The order subtotal and number of ordered items are summed for each order status in the specified date range and displayed alphabetically by status. Only orders with a specific status appear in the report. The summed order subtotal for each status is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the relative distribution of orders in the order process during a time period.

Orders by Status				
7/15/2002 through 7/31/2002				
Total Sales: \$26,937.29				
<u>Status</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
Pending	1	20	\$1,390.50	5.16%
Processed	1	1	\$8,581.76	31.86%
Shipped	10	32	\$16,965.03	62.98%
Totals	12	53	\$26,937.29	

Orders by status report

Orders by Order

The order subtotal and number of ordered items are summed for each order in the specified date range and displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page. The summed order subtotal for each order is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful for generating an itemized listing of all orders during a time period.

Orders by Order			
7/15/2002 through 7/31/2002			
Total Sales: \$26,937.29			
<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
278	2	\$639.90	2.38%
279	1	\$439.95	1.63%
281	2	\$3,399.91	12.62%
282	1	\$44.99	0.17%
283	1	\$10.10	0.04%
286	1	\$29.62	0.11%
287	3	\$372.14	1.38%
288	10	\$3,490.00	12.96%
289	10	\$180.66	0.67%
290	20	\$1,390.50	5.16%
291	1	\$8,357.76	31.03%
292	1	\$8,581.76	31.86%
12	53	\$26,937.29	

Orders by order report

Orders by Promotion

The order subtotal and number of ordered items are summed for each order that has a promotion code in the specified date range and displayed alphabetically by promotion code. Only orders with a promotion code appear in the report. The summed order subtotal for each promotion code is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the relative distribution and use of promotions during a time period.

Important: Orders for which a promotion code has been entered by the customer are displayed. This does not necessarily mean that the promotion code was successfully applied to the order. For example, the customer may have entered a promotion code that was for a specific product, but that product was never placed in the shopping cart or the product was placed in the shopping cart but removed prior to checkout.

Orders by Promotion				
7/15/2002 through 7/31/2002				
Total Sales: \$84.71				
<u>Promotion Code</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
HSFS090102	3	3	\$84.71	100.00%
Totals	3	3	\$84.71	

Orders by promotion report

Orders by Affiliate

The order subtotal and number of ordered items are summed for each order that has an affiliate ID in the specified date range and displayed alphabetically by affiliate ID. The affiliate ID is hyperlinked to the edit affiliate page in *Affiliate Administration*. Only orders with an affiliate ID appear in the report. The summed order subtotal for each affiliate ID is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the relative distribution and generation of orders due to affiliate efforts during a time period.

Important: Orders for which an affiliate ID has been captured are displayed. This does not necessarily mean that the affiliate will be credited with the order. For example, the affiliate may have an inactive status or the order may never ship.

Orders by Affiliate				
7/15/2002 through 7/31/2002				
Total Sales: \$18,330.02				
<u>Affiliate</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
HostedStore	3	22	\$18,330.02	100.00%
Totals	3	22	\$18,330.02	

Orders by affiliate report

Gross Income Reports

Reports that provide a summary of all orders that have a status which indicates the order has been processed and funds have been collected for the order. Specifically, income reports are generated based on orders where:

- The **DateProcessed** field for the order in the **Invoice** table has a value in the specified date range.
- The order status field, **Status**, for the order in the **Invoice** table has a value of *Processed*, *Back Ordered*, *Shipped*, or *Mixed Status*.
- The **OrderStatus** field for each ordered item in the **Item** table has a value of *Processed*, *Back Ordered*, or *Shipped*.

Each report may have several columns that are derived from different sources. An explanation of columns that may appear is provided below moving from left to right as the columns would appear in the report.

- **Orders** - Simply a total of the number of orders for the row heading (may be day, hour, etc). For some reports this column will not appear because it is not meaningful, i.e., orders are itemized so the value would always be 1.
- **Items** - The sum of the **Quantity** field from the **Item** table for each of the ordered items with a qualifying **OrderStatus** field value.
- **SubTotal** - The sum of the **TotalPrice** field from the **Item** table for each of the ordered items with a qualifying **OrderStatus** field value.
- **Disc** - The promotional discount amount from the **Discount** field of the **Invoice** table.
- **Cost** - The sum of the **Cost** field from the **Item** table for each of the ordered items with a qualifying **OrderStatus** field value.
- **Affiliate** - The seventh value from the double pipe (||) delimited string in the **AffiliatePayout** field of the **Invoice** table.
- **SubTotal (Admin Adjustment)** - The subtotal adjustment amount from the **SubTotalAdjustment** field of the **Invoice** table.
- **Disc (Admin Adjustment)** - The discount adjustment amount from the **DiscountAdjustment** field of the **Invoice** table.
- **Batch (Admin Adjustment)** - The batch settlement adjustment amount from the **SettlementAdjustment** field of the **Invoice** table.
- **Gross Income** - This column heading appears over two columns. The left column is the calculated gross income derived by subtracting the **Disc** and **Cost** column values from the **SubTotal** column value and adding all of the admin adjustments. The second column is the gross income percentage value derived by dividing the calculated gross income value by the **SubTotal** column value and converting to a percentage value.
- The final unlabeled column is the percentage of total gross income contributed by the order(s) in the report row. This value is derived by dividing the calculated gross income value for the orders in the report row by the total gross income value for the date range and converting to a percentage value.

Important: These reports assume that product costing is used, all adjustments have been entered correctly, items canceled from an order have been canceled, canceled orders have actually been canceled, etc. Even if all data is properly entered, these reports are provided and intended solely as an overview of store operations. These reports are not intended to replace true gross income values that would be calculated using a comprehensive accounting application.

Gross Income by Day

Processed orders summed for each day in the specified date range and displayed chronologically. Only the days that have processed orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining weekly, monthly, and quarterly gross income, relative gross income between days, etc.

Gross Income by Day												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
7/22/2002, Mon	2	3	\$1,079.85	\$0.00	\$462.49	\$0.00	\$0.00	\$0.00	\$0.00	\$617.36	(57.17%)	3.16%
7/23/2002, Tue	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(99.93%)	17.41%
7/24/2002, Wed	2	20	\$3,670.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$3,641.75	(99.21%)	18.66%
7/27/2002, Sat	7	12	\$517.51	\$2.50	\$233.99	\$0.00	\$0.00	\$0.00	\$0.00	\$281.02	(54.30%)	1.44%
7/29/2002, Mon	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
7/30/2002, Tue	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(75.00%)	32.98%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by day report

Gross Income by Hour

Processed orders summed for each one hour time period for each day in the specified date range and displayed chronologically. Only the one hour time periods that have processed orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining relative gross income between hours in a day or hours from day to day.

Gross Income by Hour												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
7/22/2002, Mon 4:00 PM - 5:00 PM	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	1.29%
7/22/2002, Mon 7:00 PM - 8:00 PM	1	2	\$639.90	\$0.00	\$274.99	\$0.00	\$0.00	\$0.00	\$0.00	\$364.91	(57.03%)	1.87%
7/23/2002, Tue 8:00 AM - 9:00 AM	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(99.93%)	17.41%
7/24/2002, Wed 2:00 AM - 3:00 AM	1	10	\$3,490.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$3,490.00	(100.00%)	17.88%
7/24/2002, Wed 5:00 PM - 6:00 PM	1	10	\$180.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$151.75	(84.00%)	0.78%
7/27/2002, Sat 12:00 PM - 1:00 PM	7	12	\$517.51	\$2.50	\$233.99	\$0.00	\$0.00	\$0.00	\$0.00	\$281.02	(54.30%)	1.44%
7/29/2002, Mon 9:00 PM - 10:00 PM	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
7/30/2002, Tue 11:00 AM - 12:00 PM	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(75.00%)	32.98%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by hour report

Gross Income by Customer

Processed orders summed for each customer with orders during the specified date range and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only the customers that have orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining which customers contribute the most to gross income during a time period.

Gross Income by Customer											
7/15/2002 through 7/31/2002											
Total Sales: \$25,607.45											
Total Gross Income: \$19,517.18 (76.22%)											
Admin Adjustments											
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Cost</u>	<u>Affiliate</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Batch</u>	<u>Gross Income</u>	
Doe, Jane	9	31	\$16,714.19	\$2.50	\$517.15	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,049.10 (84.05%)	71.98%
Doe, Jeremy	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32 (61.54%)	26.35%
Doe, Jimmy	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45 (57.38%)	1.29%
Doe, John	3	6	\$95.55	\$0.00	\$23.24	\$0.00	\$0.00	\$0.00	\$0.00	\$72.31 (75.68%)	0.37%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%

Gross income by customer report

Gross Income by Customer, Itemized

Processed orders summed for each customer during the specified date range itemized by order and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only the customers with orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining which customers contribute the most to gross income during a time period.

Gross Income by Customer, Itemized												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
Doe, Jane	9	31	\$16,714.19	\$2.50	\$517.15	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,049.10	(84.05%)	71.98%
229	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(0.06%)	0.05%
278	1	2	\$639.90	\$0.00	\$274.99	\$0.00	\$0.00	\$0.00	\$0.00	\$364.91	(2.18%)	1.87%
281	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(20.33%)	17.41%
283	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(0.06%)	0.05%
286	1	1	\$29.62	\$2.50	\$11.25	\$0.00	\$0.00	\$0.00	\$0.00	\$15.87	(0.09%)	0.08%
287	1	3	\$372.14	\$0.00	\$199.50	\$0.00	\$0.00	\$0.00	\$0.00	\$172.64	(1.03%)	0.88%
288	1	10	\$3,490.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$3,490.00	(20.88%)	17.88%
289	1	10	\$180.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$151.75	(0.91%)	0.78%
292	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(38.51%)	32.98%
Doe, Jeremy	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
291	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
Doe, Jimmy	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	1.29%
279	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	1.29%
Doe, John	3	6	\$95.55	\$0.00	\$23.24	\$0.00	\$0.00	\$0.00	\$0.00	\$72.31	(75.68%)	0.37%
257	1	2	\$20.23	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$16.11	(16.86%)	0.08%
262	1	3	\$30.33	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$26.21	(27.43%)	0.13%
282	1	1	\$44.99	\$0.00	\$15.00	\$0.00	\$0.00	\$0.00	\$0.00	\$29.99	(31.39%)	0.15%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by customer, itemized report

Gross Income by New Customers

Processed orders summed for each new customer with only one order during the specified date range and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. Only the customers that have placed their first and only order appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the gross income during a time period that can be attributed to new customers.

Gross Income by New Customers												
7/15/2002 through 7/31/2002												
Total Sales: \$8,797.71												
Total Gross Income: \$5,395.77 (61.33%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
Doe, Jeremy	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	95.32%
Doe, Jimmy	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	4.68%
Totals	2	2	\$8,797.71	\$0.00	\$1,312.50	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,395.77	61.33%	

Gross income by new customers report

Gross Income by New Customers, Itemized

Processed orders summed for each new customer with only one order during the specified date range itemized by order and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only the customers that have placed their first and only order appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the gross income during a time period that can be attributed to new customers.

Gross Income by New Customers, Itemized												
7/15/2002 through 7/31/2002												
Total Sales: \$8,797.71												
Total Gross Income: \$5,395.77 (61.33%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
Doe, Jeremy	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	95.32%
291	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	95.32%
Doe, Jimmy	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	4.68%
279	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	4.68%
Totals	2	2	\$8,797.71	\$0.00	\$1,312.50	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,395.77	61.33%	

Gross income by new customers, itemized report

Gross Income by Return Customers

Processed orders summed for each return customer that has placed an order before or placed multiple orders during the specified date range and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. Only the customers that have placed a prior order appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining the gross income during a time period that can be attributed to repeat customers.

Gross Income by Return Customers										
7/15/2002 through 7/31/2002										
Total Sales: \$16,809.74										
Total Gross Income: \$14,121.41 (84.01%)										
Admin Adjustments										
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income
Doe, Jane	9	31	\$16,714.19	\$2.50	\$517.15	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,049.10 (84.05%) 99.49%
Doe, John	3	6	\$95.55	\$0.00	\$23.24	\$0.00	\$0.00	\$0.00	\$0.00	\$72.31 (75.68%) 0.51%
Totals	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41 84.01%

Gross income by customers report

Gross Income by Return Customers, Itemized

Processed orders summed for each return customer that has placed an order before or placed multiple orders during the specified date range itemized by order and displayed alphabetically by customer name. The customer name is hyperlinked to the edit customer page in *Customer Administration*. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only the customers that have placed a prior order appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining the gross income during a time period that can be attributed to repeat customers.

Gross Income by Return Customers, Itemized												
7/15/2002 through 7/31/2002												
Total Sales: \$16,809.74												
Total Gross Income: \$14,121.41 (84.01%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
Doe, Jane	9	31	\$16,714.19	\$2.50	\$517.15	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,049.10	(84.05%)	99.49%
229	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(0.06%)	0.07%
278	1	2	\$639.90	\$0.00	\$274.99	\$0.00	\$0.00	\$0.00	\$0.00	\$364.91	(2.18%)	2.58%
281	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(20.33%)	24.06%
283	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(0.06%)	0.07%
286	1	1	\$29.62	\$2.50	\$11.25	\$0.00	\$0.00	\$0.00	\$0.00	\$15.87	(0.09%)	0.11%
287	1	3	\$372.14	\$0.00	\$199.50	\$0.00	\$0.00	\$0.00	\$0.00	\$172.64	(1.03%)	1.22%
288	1	10	\$3,490.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$3,490.00	(20.88%)	24.71%
289	1	10	\$180.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$151.75	(0.91%)	1.07%
292	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(38.51%)	45.58%
Doe, John	3	6	\$95.55	\$0.00	\$23.24	\$0.00	\$0.00	\$0.00	\$0.00	\$72.31	(75.68%)	0.51%
257	1	2	\$20.23	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$16.11	(16.86%)	0.11%
262	1	3	\$30.33	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$26.21	(27.43%)	0.19%
282	1	1	\$44.99	\$0.00	\$15.00	\$0.00	\$0.00	\$0.00	\$0.00	\$29.99	(31.39%)	0.21%
Totals	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41	84.01%	

Gross income by return customers, itemized report

Gross Income by Billing Country

Processed orders summed for each country from which a customer has ordered during the specified date range and displayed alphabetically by country ID. Only the countries with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the geographic origin of gross income during a time period.

Gross Income by Billing Country												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
<hr/>												
Admin Adjustments												
<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Cost</u>	<u>Affiliate</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Batch</u>	<u>Gross Income</u>			
CA	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
US	13	38	\$17,249.69	\$2.50	\$727.89	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,373.86	(83.33%)	73.65%
<hr/>												
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	
<hr/>												

Gross income by billing country report

Gross Income by Billing State/Province

Processed orders summed for each state/province from which a customer has ordered during the specified date range and displayed alphabetically by state/province ID. Only the states/provinces with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining the geographic origin of gross income during a time period.

Gross Income by Billing State/Province										
7/15/2002 through 7/31/2002										
Total Sales: \$25,607.45										
Total Gross Income: \$19,517.18 (76.22%)										
Admin Adjustments										
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income
AL	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45 (57.38%) 1.29%
BC	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32 (61.54%) 26.35%
WA	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41 (84.01%) 72.35%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18 76.22%

Gross income by billing state/province report

Gross Income by Billing Country, State/Province

Processed orders summed for each country and itemized by the state/province from which a customer has ordered during the specified date range and displayed alphabetically by country ID and state/province ID. Only the countries and states/provinces with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the geographic origin of gross income during a time period.

Gross Income by Billing Country, State/Province												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
CA	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
BC	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
US	13	38	\$17,249.69	\$2.50	\$727.89	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,373.86	(83.33%)	73.65%
AL	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(1.46%)	1.29%
WA	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41	(81.86%)	72.35%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by billing country, state/province report

Gross Income by Shipping Country

Processed orders summed for each country to which a customer has shipped an order during the specified date range and displayed alphabetically by country ID. Only the shipping destination countries with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining the geographic destination of gross income during a time period.

Gross Income by Shipping Country										
7/15/2002 through 7/31/2002										
Total Sales: \$25,607.45										
Total Gross Income: \$19,517.18 (76.22%)										
Admin Adjustments										
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income
CA	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32 (61.54%) 26.35%
US	13	38	\$17,249.69	\$2.50	\$727.89	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,373.86 (83.33%) 73.65%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18 76.22%

Gross income by shipping country report

Gross Income by Shipping State/Province

Processed orders summed for each state/province to which a customer has shipped an order during the specified date range and displayed alphabetically by state/province ID. Only the shipping destination states/provinces with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the geographic destination of gross income during a time period.

Gross Income by Shipping State/Province											
7/15/2002 through 7/31/2002											
Total Sales: \$25,607.45											
Total Gross Income: \$19,517.18 (76.22%)											
<hr/>											
						Admin Adjustments					
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income	
AL	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45 (57.38%)	1.29%
BC	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32 (61.54%)	26.35%
WA	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41 (84.01%)	72.35%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%

Gross income by shipping state/province report

Gross Income by Shipping Country, State/Province

Processed orders summed for each country and itemized by the state/province to which a customer has shipped an order during the specified date range and displayed alphabetically by country ID and state/province ID. Only the shipping destination countries and states/provinces with customer orders appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the geographic destination of gross income during a time period.

Gross Income by Shipping Country, State/Province												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
CA	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
BC	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
US	13	38	\$17,249.69	\$2.50	\$727.89	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,373.86	(83.33%)	73.65%
AL	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(1.46%)	1.29%
WA	12	37	\$16,809.74	\$2.50	\$540.39	\$2,145.44	\$0.00	\$0.00	\$0.00	\$14,121.41	(81.86%)	72.35%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by shipping country, state/province report

Gross Income by Source

Processed orders summed for each order source to which a customer has shipped an order during the specified date range and displayed alphabetically by source name. Only the order sources that have orders appear in the report. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful in determining the relative gross income between sources.

Gross Income by Source											
7/15/2002 through 7/31/2002											
Total Sales: \$25,607.45											
Total Gross Income: \$19,517.18 (76.22%)											
											Admin Adjustments
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income	
Retail Store	2	5	\$50.56	\$0.00	\$8.24	\$0.00	\$0.00	\$0.00	\$0.00	\$42.32 (83.70%)	0.22%
Web	12	34	\$25,556.89	\$2.50	\$1,844.65	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,474.86 (76.20%)	99.78%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%

Gross income by source report

Gross Income by Order

Processed orders summed for each order during the specified date range and displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful for generating an itemized listing of the gross income from each order during a time period.

Gross Income by Order												
7/15/2002 through 7/31/2002												
Total Sales: \$25,607.45												
Total Gross Income: \$19,517.18 (76.22%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
229	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(100.00%)	0.05%
257	1	2	\$20.23	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$16.11	(79.63%)	0.08%
262	1	3	\$30.33	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$26.21	(86.42%)	0.13%
278	1	2	\$639.90	\$0.00	\$274.99	\$0.00	\$0.00	\$0.00	\$0.00	\$364.91	(57.03%)	1.87%
279	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	1.29%
281	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(99.93%)	17.41%
282	1	1	\$44.99	\$0.00	\$15.00	\$0.00	\$0.00	\$0.00	\$0.00	\$29.99	(66.66%)	0.15%
283	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(100.00%)	0.05%
286	1	1	\$29.62	\$2.50	\$11.25	\$0.00	\$0.00	\$0.00	\$0.00	\$15.87	(53.58%)	0.08%
287	1	3	\$372.14	\$0.00	\$199.50	\$0.00	\$0.00	\$0.00	\$0.00	\$172.64	(46.39%)	0.88%
288	1	10	\$3,490.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$3,490.00	(100.00%)	17.88%
289	1	10	\$180.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$151.75	(84.00%)	0.78%
291	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.35%
292	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(75.00%)	32.98%
Totals	14	39	\$25,607.45	\$2.50	\$1,852.89	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,517.18	76.22%	

Gross income by order report

Gross Income by Unsettled Order

Processed orders summed for each order that has not been batch settled during the specified date range and displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful for generating an itemized listing of the gross income from each order that has not been batched during a time period.

Gross Income by Unsettled Order												
7/15/2002 through 7/31/2002												
Total Sales: \$25,556.89												
Total Gross Income: \$19,474.86 (76.20%)												
Admin Adjustments												
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income		
229	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(100.00%)	0.05%
278	1	2	\$639.90	\$0.00	\$274.99	\$0.00	\$0.00	\$0.00	\$0.00	\$364.91	(57.03%)	1.87%
279	1	1	\$439.95	\$0.00	\$187.50	\$0.00	\$0.00	\$0.00	\$0.00	\$252.45	(57.38%)	1.30%
281	1	2	\$3,399.91	\$0.00	\$2.50	\$0.00	\$0.00	\$0.00	\$0.00	\$3,397.41	(99.93%)	17.45%
282	1	1	\$44.99	\$0.00	\$15.00	\$0.00	\$0.00	\$0.00	\$0.00	\$29.99	(66.66%)	0.15%
283	1	1	\$10.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10.10	(100.00%)	0.05%
286	1	1	\$29.62	\$2.50	\$11.25	\$0.00	\$0.00	\$0.00	\$0.00	\$15.87	(53.58%)	0.08%
287	1	3	\$372.14	\$0.00	\$199.50	\$0.00	\$0.00	\$0.00	\$0.00	\$172.64	(46.39%)	0.89%
288	1	10	\$3,490.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$3,490.00	(100.00%)	17.92%
289	1	10	\$180.66	\$0.00	\$28.91	\$0.00	\$0.00	\$0.00	\$0.00	\$151.75	(84.00%)	0.78%
291	1	1	\$8,357.76	\$0.00	\$1,125.00	\$2,089.44	\$0.00	\$0.00	\$0.00	\$5,143.32	(61.54%)	26.41%
292	1	1	\$8,581.76	\$0.00	\$0.00	\$2,145.44	\$0.00	\$0.00	\$0.00	\$6,436.32	(75.00%)	33.05%
Totals	12	34	\$25,556.89	\$2.50	\$1,844.65	\$4,234.88	\$0.00	\$0.00	\$0.00	\$19,474.86	76.20%	

Gross income by unsettled order report

Gross Income by Settled Order

Processed orders summed for each order that has been batch settled during the specified date range and displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. A full explanation of each column value can be found in the [Store Administration > Reports > Date Dependent Reports > Gross Income Reports](#) section.

This report can be useful for generating an itemized listing of the gross income from each order that has been batched during a time period.

Gross Income by Settled Order												
7/15/2002 through 7/31/2002												
Total Sales: \$50.56												
Total Gross Income: \$42.32 (83.70%)												
Admin Adjustments												
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Cost</u>	<u>Affiliate</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Batch</u>	<u>Gross Income</u>		
257	1	2	\$20.23	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$16.11	(79.63%)	38.07%
262	1	3	\$30.33	\$0.00	\$4.12	\$0.00	\$0.00	\$0.00	\$0.00	\$26.21	(86.42%)	61.93%
Totals	2	5	\$50.56	\$0.00	\$8.24	\$0.00	\$0.00	\$0.00	\$0.00	\$42.32	83.70%	

Gross income by settled order report

Gross Income by Batch

Processed orders summed for each batch during the specified date range and displayed alphabetically by batch number. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful for generating an itemized listing of the gross income from each order batch during a time period.

Gross Income by Batch											
7/15/2002 through 7/31/2002											
Total Sales: \$50.56											
Total Gross Income: \$42.32 (83.70%)											
Admin Adjustments											
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Cost</u>	<u>Affiliate</u>	<u>SubTotal</u>	<u>Disc</u>	<u>Batch</u>	<u>Gross Income</u>	
Batch_1	2	5	\$50.56	\$0.00	\$8.24	\$0.00	\$0.00	\$0.00	\$0.00	\$42.32 (83.70%)	100.00%
Totals	2	5	\$50.56	\$0.00	\$8.24	\$0.00	\$0.00	\$0.00	\$0.00	\$42.32	83.70%

Gross income by batch report

Gross Income by Promotion

Processed orders summed for each order that has a promotion code during the specified date range and displayed alphabetically by promotion code. Only orders with a promotion code appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the relative distribution and use of promotions as a contributing factor to gross income during a time period.

Important: Orders for which a promotion code has been entered by the customer are displayed. This does not necessarily mean that the promotion code was successfully applied to the order. For example, the customer may have entered a promotion code that was for a specific product, but that product was never placed in the shopping cart or the product was placed in the shopping cart but removed prior to checkout.

Gross Income by Promotion											
7/15/2002 through 7/31/2002											
Total Sales: \$84.71											
Total Gross Income: \$55.96 (66.06%)											
Admin Adjustments											
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income	
HSFS090102	3	3	\$84.71	\$2.50	\$26.25	\$0.00	\$0.00	\$0.00	\$0.00	\$55.96 (66.06%)	100.00%
Totals	3	3	\$84.71	\$2.50	\$26.25	\$0.00	\$0.00	\$0.00	\$0.00	\$55.96	66.06%

Gross income by promotion report

Gross Income by Affiliate

Processed orders summed for each order that has an affiliate ID during the specified date range and displayed alphabetically by affiliate ID. Only orders with an affiliate ID appear in the report. A full explanation of each column value can be found in the [Store Administration](#) > [Reports](#) > [Date Dependent Reports](#) > [Gross Income Reports](#) section.

This report can be useful in determining the relative distribution and generation of orders due to affiliate efforts as a contributing factor to gross income during a time period.

Important: Orders for which an affiliate ID has been captured are displayed. This does not necessarily mean that the affiliate will be credited with the order. For example, the affiliate may have an inactive status or the order may never ship.

Gross Income by Affiliate										
7/15/2002 through 7/31/2002										
Total Sales: \$16,939.52										
Total Gross Income: \$11,579.64 (68.36%)										
Admin Adjustments										
	Orders	Items	SubTotal	Disc	Cost	Affiliate	SubTotal	Disc	Batch	Gross Income
HostedStore	2	2	\$16,939.52	\$0.00	\$1,125.00	\$4,234.88	\$0.00	\$0.00	\$0.00	\$11,579.64 (68.36%) 100.00%
Totals	2	2	\$16,939.52	\$0.00	\$1,125.00	\$4,234.88	\$0.00	\$0.00	\$0.00	\$11,579.64 68.36%

Gross income by affiliate report

Order Status Reports

Unlike the *Sales Order Reports* and *Gross Income Reports*, none of the order status reports have generic restrictions imposed as to the status of orders and ordered items. This set of reports is generated based off of the **DateCreated** field from the **Invoice** table falling between the specified date range as well as any other restrictions imposed by the specific report.

Sections

- [All Orders by Status](#)
- [Declined Orders, Itemized](#)
- [Canceled Orders, Itemized](#)
- [Abandoned Orders, Itemized](#)

All Orders by Status

The order subtotal and number of ordered items are summed for each order status in the specified date range and displayed alphabetically by status. Only orders with a specific status appear in the report. The summed order subtotal for each status is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the relative distribution of all orders in the order process.

All Orders by Status				
7/15/2002 through 7/31/2002				
Total Sales: \$27,087.28				
<u>Status</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
In Cart	1	2	\$149.99	0.55%
Pending	1	20	\$1,390.50	5.13%
Processed	1	1	\$8,581.76	31.68%
Shipped	10	32	\$16,965.03	62.63%
Totals	13	55	\$27,087.28	

All orders by status report

Declined Orders, Itemized

The order subtotal and number of ordered items for each order placed in the specified date range that has a status of *Declined* displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only orders with a *Declined* status appear in the report. The summed order subtotal for each order is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the number and value of orders that have been declined.

Declined Orders, Itemized			
7/1/2002 through 7/31/2002			
Total Sales: \$10.10			
<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
264	1	\$10.10	100.00%
1	1	\$10.10	

Declined orders, itemized report

Canceled Orders, Itemized

The order subtotal and number of ordered items for each order placed in the specified date range that has a status of *Cancelled* displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only orders with a *Cancelled* status appear in the report. The summed order subtotal for each order is also displayed as a percentage of the total order subtotal for the date range.

Important: This report only shows orders where the entire order has been canceled. It does not reflect orders that may have had some, but not all, ordered items canceled.

This report can be useful in determining the number and value of orders that have been canceled.

Cancelled Orders, Itemized			
7/1/2000 through 7/31/2002			
Total Sales: (\$8.11)			
<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
202	1	\$10.10	-124.54%
217	1	\$1.01	-12.45%
223	1	\$1.01	-12.45%
260	2	(\$20.23)	249.45%
4	5	(\$8.11)	

Cancelled orders, itemized report

Abandoned Orders, Itemized

The order subtotal and number of ordered items for each order created in the specified date range that has a status of *Abandoned* displayed numerically by invoice number. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only orders with an *Abandoned* status appear in the report. The summed order subtotal for each order is also displayed as a percentage of the total order subtotal for the date range.

This report can be useful in determining the number and value of orders that have been abandoned.

Abandoned Orders, Itemized			
7/1/2000 through 7/31/2002			
Total Sales: \$1.01			
<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
218	5	\$1.01	100.00%
1	5	\$1.01	

Abandoned orders, itemized report

Product Sales Reports

The results of a product sales report are nearly identical to the corresponding *Orders Sales Report* except that the ordered item must have reached at least the order processed stage. Specifically, the ordered item must have a **DateProcessed** field value that falls in the specified date range and an **OrderStatus** field value of *Processed*, *Back Ordered*, or *Shipped*. The restrictions are just like those used for *Gross Income Reports*.

Sections

- [Sales by Product](#)
- [Sales by Category](#)
- [Sales by Category, SubCategory](#)
- [Sales by Department](#)
- [Sales by Department, Product](#)
- [Sales by Manufacturer](#)
- [Sales by Manufacturer, Product](#)

Sales by Product

The ordered item total price and number of ordered items are summed for each product ordered during the specified date range and displayed alphabetically by product ID. Only products for orders that have been processed appear in the report. Each product's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between products during a date range.

Product Sales by Product					
7/15/2002 through 7/31/2002					
Total Sales: \$25,607.45					
<u>ProductID</u>	<u>Product Name</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
98274	Adaptable Dual Pocket Desktop Charger	1	1	\$29.62	0.12%
AMP3WATT	3 Watt Booster Amplifier	1	1	\$199.95	0.78%
BC2500	Belt Clip	1	1	\$10.10	0.04%
BP2	"All'armi, pensieri"	1	1	\$30.00	0.12%
BWLassoStudio	Lasso Studio	1	10	\$3,490.00	13.63%
CMP316MMC	16 MB MMC Chip	1	1	\$35.00	0.14%
DTH1100	Desktop Holder	1	3	\$48.60	0.19%
DTH3	Desktop Holder	1	4	\$75.60	0.30%
IBR-1	In-Building Repeater	3	3	\$1,209.89	4.72%
LargeCustomPizza	Large Pizza	1	1	\$28.15	0.11%
NTN1987	Holster	2	3	\$30.30	0.12%
NTN8011	Battery Belt Clip w/Insert	1	1	\$12.15	0.05%
PCP9100AM	Amethyst Faceplate	1	2	\$21.46	0.08%
PentiumIII	Custom Built Intel Celeron/PIII Computer	2	2	\$11,729.52	45.81%
PentiumIV	Custom Built Pentium IV Computer	1	1	\$8,581.76	33.51%
RLN4927	Case w/Swivel & Cutouts	2	2	\$20.26	0.08%
SGCD0015801	Dual Port Docking Station	1	1	\$44.99	0.18%
SH710	Simple Holder	1	1	\$10.10	0.04%
			Totals	39	\$25,607.45

Sales by product report

Sales by Category

The ordered item total price and number of ordered items are summed for each product category for orders processed during the specified date range and displayed alphabetically by product category. Only product categories that have had products ordered and processed appear in the report. Each product category's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product categories during a date range.

Product Sales by Category				
7/15/2002 through 7/31/2002				
Total Sales: \$25,607.45				
<u>Category</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
Cellular Accessories	10	24	\$1,748.02	6.83%
Computers	3	3	\$20,311.28	79.32%
Pizza	1	1	\$28.15	0.11%
Sheet Music	1	1	\$30.00	0.12%
Software	1	10	\$3,490.00	13.63%
	Totals	39	\$25,607.45	

Sales by category report

Sales by Department, Product

The ordered item total price and number of ordered items are summed for each product department and itemized by product ID for orders processed during the specified date range and displayed alphabetically by product department. Only product departments that have had products ordered and processed appear in the report. Each product department's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product departments during a date range.

Product Sales by Department, Product				
7/15/2002 through 7/31/2002				
Total Sales: \$13,756.67				
	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
Electronics	3	7	\$13,698.52	99.58%
BWLassoStudio	1	5	\$1,745.00	12.68%
PentiumIII	1	1	\$3,371.76	24.51%
PentiumIV	1	1	\$8,581.76	62.38%
Music	1	1	\$30.00	0.22%
BP2	1	1	\$30.00	0.22%
Restaurant	1	1	\$28.15	0.20%
LargeCustomPizza	1	1	\$28.15	0.20%
	Totals	9	\$13,756.67	

Sales by department, product report

Sales by Manufacturer

The ordered item total price and number of ordered items are summed for each product manufacturer for orders processed during the specified date range and displayed alphabetically by product manufacturer. Only product manufacturers that have had products ordered and processed appear in the report. Each product manufacturer's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product manufacturers during a date range.

Product Sales by Manufacturer				
7/15/2002 through 7/31/2002				
Total Sales: \$25,607.45				
<u>Manufacturer</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
AudioVox	4	13	\$229.01	0.89%
BIM	1	1	\$30.00	0.12%
Blue World Communications, Inc.	1	10	\$3,490.00	13.63%
HostedStore Corporation	3	3	\$20,311.28	79.32%
LG	1	1	\$44.99	0.18%
Motorola	1	1	\$29.62	0.12%
Nextel	5	10	\$1,472.55	5.75%
	Totals	39	\$25,607.45	

Sales by manufacturer report

Sales by Manufacturer, Product

The ordered item total price and number of ordered items are summed for each product manufacturer and itemized by product ID for orders processed during the specified date range and displayed alphabetically by product manufacturer. Only product manufacturers that have had products ordered and processed appear in the report. Each product manufacturer's summed total price is also displayed as a percentage of the summed total price for all ordered products during the date range.

This report can be useful in determining relative order volume between product manufacturers during a date range.

Product Sales by Manufacturer, Product				
7/15/2002 through 7/31/2002				
Total Sales: \$14,676.27				
	Orders	Items	SubTotal	Percentage
AudioVox	3	12	\$200.86	1.37%
BC2500	1	1	\$10.10	0.07%
CMP316MMC	1	1	\$35.00	0.24%
DTH1100	1	3	\$48.60	0.33%
DTH3	1	4	\$75.60	0.52%
PCP9100AM	1	2	\$21.46	0.15%
SH710	1	1	\$10.10	0.07%
BIM	1	1	\$30.00	0.20%
BP2	1	1	\$30.00	0.20%
Blue World Communications, Inc.	1	5	\$1,745.00	11.89%
BWLassoStudio	1	5	\$1,745.00	11.89%
HostedStore Corporation	2	2	\$11,953.52	81.45%
PentiumIII	1	1	\$3,371.76	22.97%
PentiumIV	1	1	\$8,581.76	58.47%
LG	1	1	\$44.99	0.31%
SGCD0015801	1	1	\$44.99	0.31%
Motorola	1	1	\$29.62	0.20%
98274	1	1	\$29.62	0.20%
Nextel	4	5	\$672.28	4.58%
AMP3WATT	1	1	\$199.95	1.36%
IBR-1	1	1	\$439.95	3.00%
NTN1987	1	1	\$10.10	0.07%
NTN8011	1	1	\$12.15	0.08%
RLN4927	1	1	\$10.13	0.07%
Totals	27		\$14,676.27	

Sales by manufacturer, product report

Sales Tax Reports

Sales tax reports summarize the tax collected from the customer, any tax adjustments made, and a final tax collected figure for orders placed during the specified date range.

Sections

- [Sales Tax Collected by State/Province](#)

Sales Tax Collected by State/Province

Tax collected from the customer, any tax adjustments made, and a final tax collected figure itemized by state/province for orders placed during the specified date range.

Sales Tax Collected by State/Province						
7/15/2002 through 7/31/2002						
Total Sales Taxed: \$16,809.74						
Total Tax Collected: \$1,462.45						
<u>State/Province</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Tax</u>	<u>Adjustment</u>	<u>Collected</u>
WA	12	37	\$16,809.74	\$1,462.45	\$0.00	\$1,462.45
Totals	12	37	\$16,809.74	\$1,462.45	\$0.00	\$1,462.45

Sales tax by collected state/province report

Discount Reports

Discount reports summarize adjustments made to an order's subtotal, shipping cost, or tax for all orders that have a status which indicates the order has been processed and funds have been collected for the order. Specifically, discount reports are generated based on orders where:

- The **DateProcessed** field for the order in the **Invoice** table has a value in the specified date range.
- The order status field, **Status**, for the order in the **Invoice** table has a value of *Processed*, *Back Ordered*, *Shipped*, or *Mixed Status*.
- The **OrderStatus** field for each ordered item in the **Item** table has a value of *Processed*, *Back Ordered*, or *Shipped*.

Promotions

The number of orders, number of items purchased, subtotal, sum of discounts given, discount expressed as a percentage of percentage of sales, and the subtotal expressed as a percent of total period sales for the specified date range.

Promotions								
7/15/2002 through 7/31/2002								
Total Sales: \$84.71								
Total Discounts: \$2.50								
							Percent of Total	
<u>Promotion</u>	<u>Orders</u>	<u>Items</u>	<u>SubTotal</u>	<u>Discount</u>	<u>Percentage</u>	<u>SubTotal</u>	<u>Discount</u>	
HSFS090102	3	3	\$84.71	\$2.50	2.95%	100.00%	100.00%	
Totals	3	3	\$84.71	\$2.50	2.95%			

Promotions report

Admin Adjustments, Itemized

Details order subtotal, discount, subtotal adjustment, discount adjustment, shipping adjustment, tax adjustment, and batch adjustment for all orders that have discounts or adjustments during the time period.

Admin Adjustments, Itemized									
7/1/2000 through 7/31/2002									
Total Sales with Adjustments: \$229.50									
Total Adjustments: (\$29.50)									
<hr/>									
Admin Adjustments									
<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Discount</u>	<u>SubTotal</u>	<u>Discount</u>	<u>Shipping</u>	<u>Tax</u>	<u>Batch</u>	<u>Total</u>
195	6	\$229.50	\$0.00	(\$29.50)	\$0.00	\$0.00	\$0.00	\$0.00	(\$29.50)
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
1	6	\$229.50	\$0.00	(\$29.50)	\$0.00	\$0.00	\$0.00	\$0.00	(\$29.50)
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>

Admin adjustments, itemized report

Affiliate Reports

Reports that summarize the orders attributed to affiliates, earned affiliate compensation, and payouts issued.

All affiliate reports are generated for only those orders that have had a payout posted.

Sections

- [Payout History by Affiliate, Itemized](#)
- [Sales by Affiliate](#)
- [Sales by Affiliate, Itemized](#)

Payout History by Affiliate, Itemized

The order subtotal, date the affiliate was credited with the order, affiliate plan, affiliate payout, date of payout, check number, and amount of payout are summed for each order that has an affiliate ID in the specified date range, itemized by invoice number, and displayed alphabetically by affiliate ID. Only orders with an affiliate ID appear in the report.

This report can be useful in determining when and how affiliates were compensated during a time period.

Payout History by Affiliate, Itemized								
7/1/2000 through 7/31/2002								
<u>Affiliate</u>	<u>Invoice</u>	<u>SubTotal</u>	<u>Date Credited</u>	<u>Affiliate Plan</u>	<u>Payout</u>	<u>Date Paid</u>	<u>Check</u>	<u>Amount</u>
HostedStore	63	\$5,224.93	3/17/2002	Percentage (25.00%)	\$1,306.23	5/26/2002	1345	\$1,306.23
	71	\$33,524.20	3/17/2002	Percentage (25.00%)	\$8,391.05	7/29/2002	1500	\$8,391.05
	146	\$19.95	3/31/2002	Percentage (25.00%)	\$4.99	5/20/2002	1234	\$54.99
	195	\$200.00	4/28/2002	Percentage (25.00%)	\$50.00	5/20/2002	1234	\$54.99
SampleAffiliate	187	\$336.12	4/25/2002	Fixed (\$5.00)	\$5.00	5/26/2002	7101	\$5.00

Payout history by affiliate, itemized report

Sales by Affiliate

The number of ordered items, order subtotal, and affiliate payout are summed for each order that has an affiliate ID in the specified date range and displayed alphabetically by affiliate ID. The affiliate ID is hyperlinked to the edit affiliate page in *Affiliate Administration*. Only orders with an affiliate ID appear in the report. The summed order subtotal for each affiliate ID is also displayed as a percentage of the total order subtotal along with the payout expressed as a percentage of the total payout for the date range.

This report can be useful in determining the relative distribution and generation of orders due to affiliate efforts during a time period.

Sales by Affiliate								
7/1/2000 through 7/31/2002								
Total Affiliate Sales: \$85,331.42								
Total Affiliate Payout: \$21,253.82								
Affiliate	Plan	Orders	Items	SubTotal	Payout	Percent	Percent of Total	
							SubTotal	Payout
HostedStore	Percentage (25.00%)	3	18	\$25,115.87	\$6,278.97	25.00%	29.43%	29.54%
SampleAffiliate	Fixed (\$5.00)	1	6	\$336.12	\$5.00	1.49%	0.39%	0.02%
HostedStore	Percentage (25.00%)	10	48	\$59,879.43	\$14,969.86	25.00%	70.17%	70.43%
Totals		14	72	\$85,331.42	\$21,253.82	24.91%		

Sales by affiliate report

Sales by Affiliate, Itemized

The number of ordered items, order subtotal, and affiliate payout are summed for each order that has an affiliate ID in the specified date range, itemized by invoice number, and displayed alphabetically by affiliate ID. The affiliate ID is hyperlinked to the edit affiliate page in *Affiliate Administration*. The invoice number is hyperlinked to the edit order page in *Order Administration*. Only orders with an affiliate ID appear in the report. The summed order subtotal for each affiliate ID is also displayed as a percentage of the total order subtotal along with the payout expressed as a percentage of the total payout for the date range.

This report can be useful in determining the relative distribution and generation of orders due to affiliate efforts during a time period.

Sales by Affiliate, Itemized								
7/1/2000 through 7/31/2002								
Total Affiliate Sales: \$83,940.92								
Total Affiliate Payout: \$20,906.20								
Affiliate	Plan	Invoice	Items	SubTotal	Payout	Percent	Percent of Total	
							SubTotal	Payout
HostedStore	Percentage (25.00%)	12	46	\$83,604.80	\$20,901.20	25.00%	99.60%	99.98%
		63	5	\$5,224.93	\$1,306.23	25.00%	1.56%	24.99%
		71	16	\$33,524.20	\$9,687.28	28.90%	11.54%	160.36%
		146	1	\$19.95	\$9,692.27	48582.81%	11.55%	0.10%
		185	10	\$16,239.66	\$13,752.18	84.68%	16.38%	77.68%
		195	6	\$229.50	\$13,809.56	6017.24%	16.45%	1.10%
		196	2	\$8,646.71	\$15,971.24	184.71%	19.03%	41.36%
		200	1	\$5.40	\$15,972.59	295788.66%	19.03%	0.03%
		201	1	\$909.99	\$16,200.08	1780.25%	19.30%	4.35%
		211	1	\$909.99	\$16,427.58	1805.25%	19.57%	4.35%
		216	1	\$954.95	\$16,666.32	1745.26%	19.85%	4.57%
		291	1	\$8,357.76	\$18,755.76	224.41%	22.34%	39.98%
		292	1	\$8,581.76	\$20,901.20	243.55%	24.90%	41.05%
SampleAffiliate	Fixed (\$5.00)	1	6	\$336.12	\$5.00	1.49%	0.40%	0.02%
		187	6	\$336.12	\$5.00	1.49%	0.01%	1.61%
Totals		13	52	\$83,940.92	\$20,906.20	24.91%		

Sales by affiliate, itemized report

Shipping Reports

Reports that summarize orders with items that are designated for drop shipping by a supplier.

Sections

- [Drop Ship Orders by Supplier, Invoice](#)
- [Drop Ships Pending by Supplier, Invoice](#)

Drop Ship Orders by Supplier, Invoice

The invoice number, item count, subtotal, and subtotal expressed as a percentage of sales for the specified date range, itemized by invoice number, and displayed alphabetically by supplier name. The invoice number is hyperlinked to the edit order page in *Order Administration*. The supplier name is hyperlinked to the edit supplier page in *Supplier Administration*. Only orders with items that are to be drop shipped appear in the report.

This report can be useful in determining which orders have items that are fulfilled directly by a supplier.

Drop Ship Orders by Supplier, Invoice				
7/15/2002 through 7/31/2002				
Total Sales: \$25,061.79				
<u>Supplier</u>	<u>Invoice</u>	<u>Items</u>	<u>SubTotal</u>	<u>Percentage</u>
HostedStore Corporation	7	19	\$24,937.59	99.50%
	278	2	\$639.90	2.55%
	279	1	\$439.95	1.76%
	281	1	\$3,371.76	13.45%
	288	10	\$3,490.00	13.93%
	289	3	\$56.46	0.23%
	291	1	\$8,357.76	33.35%
	292	1	\$8,581.76	34.24%
Test Company	1	7	\$124.20	0.50%
	289	7	\$124.20	0.50%

Drop ship orders by supplier, invoice report

Drop Ships Pending by Supplier, Invoice

The invoice number, item count, subtotal, and subtotal expressed as a percentage of sales for the specified date range, itemized by invoice number, and displayed alphabetically by supplier name. The invoice number is hyperlinked to the edit order page in *Order Administration*. The supplier name is hyperlinked to the edit supplier page in *Supplier Administration*. Only orders with items that are to be drop shipped but do not have a *Shipped* status appear in the report.

This report can be useful in determining which orders have items that are fulfilled directly by a supplier, but have not shipped yet.

Drop Ships Pending by Supplier, Invoice						
7/15/2002 through 7/31/2002						
Total Sales: \$8,665.62						
Supplier	Invoice	Items	Product	Status	SubTotal	
HostedStore Corporation	2	6			\$8,665.62	100.00%
<i>Processed: 7/30/2002</i>	292	1			\$8,581.76	99.03%
		1	Custom Built Pentium IV Computer	Processed	\$8,581.76	\$0.00
<i>Processed: 7/31/2002</i>	293	5			\$83.86	0.97%
		2	Rapid Travel Charger	Processed	\$20.92	\$7.50
		3	Extra Capacity NIMH AA Spare Battery	Processed	\$62.94	\$18.75

Drop ships pending by supplier, invoice report

Date Independent Reports

All of the reports listed below are generated by clicking the appropriate button. All of the reports are based off of the current date.



Date independent report form area of the report administration menu

Sections

- [Image Library](#)
- [Missing Images](#)
- [Suggested Inventory Order](#)
- [Search to Sales Conversion](#)
- [Affiliate Referrals Conversion](#)
- [General Store Statistics](#)

Image Library

Provides a detailed listing of products entered in the store catalog, products that are available for sale, and which images have been uploaded to the Web server. If a product is selling in the store, then a red warning message will be shown if one or more of the thumbnail, detail, or zoom images are missing. Messages will also appear for products that have images missing but are not currently selling in the store and for products that have images but have not been entered into the **Product** table. Clicking any of the **ProductID** hyperlinks will open a new browser window that displays the images available for that product.

Use this report as a reference to determine which products still need images and which products still need to be entered into the **Product** table. Red warning messages are given for problems that may affect store aesthetics so those issues should be addressed immediately.

Important: This report can take awhile to render so consider running it only when needed if the product catalog contains many products. For reference, the length of time to run the report during testing was 53 seconds to process 2,347 products and about 45 additional seconds to render the page (this will vary with browsers).

Image Library						
Click the Product ID to view the image(s).						
Product ID	Product Name	Sell	Thumbnail	Detail	Zoom	Warnings
00006363	SQL test		*	*	*	Image(s) missing.
	SQL test		*	*	*	Image(s) missing.
	20 Studies	Y	*	*	*	Image(s) missing. Product is selling in the store.
00007380	Lime Jello Marshmallow Cottage Cheese Surprise (Medium Range)	Y	*	*	*	Image(s) missing. Product is selling in the store.
00008273	Cabaret Songs, Volumes 1&2 (Medium Range)	Y	*	*	*	Image(s) missing. Product is selling in the store.
00120566	Evita (Vocal Selections)	Y	*	*	*	Image(s) missing. Product is selling in the store.

A small portion of the image library report

Missing Images

Provides a listing of products that are in the **Product** table, but are missing images. This report is a variation of the **Image Library** report. The same information is reported, but only for products that are missing images.

If the store has been configured to use zoom images, then a product will be reported as missing an image if the thumbnail, detail, or zoom image is missing. Otherwise, a product will be reported as missing an image if the thumbnail or detail image is missing.

Important: This report can take awhile to render if a lot of products are missing images.

Missing Images						
Click the Product ID to view the image(s).						
Product ID	Product Name	Sell	Thumbnail	Detail	Zoom Warnings	
	SQL test		*	*	*	Image(s) missing.
	SQL test		*	*	*	Image(s) missing.
00006363	20 Studies	Y	*	*	*	Image(s) missing. Product is selling in the store.
00007380	Lime Jello Marshmallow Cottage Cheese Surprise (Medium Range)	Y	*	*	*	Image(s) missing. Product is selling in the store.
00008273	Cabaret Songs, Volumes 1&2 (Medium Range)	Y	*	*	*	Image(s) missing. Product is selling in the store.
00120566	Evita (Vocal Selections)	Y	*	*	*	Image(s) missing. Product is selling in the store.

A small portion of the missing images report

Suggested Inventory Order

Provides the current inventory, current build to quantity, and the suggested order quantity (difference between the build to and current inventory) for products that have a current inventory less than the build to amount.

This report is simply a guideline that may or may not be useful depending upon how inventory is ordered for the store. The historical product sales are also detailed in the report in case that will affect the ordering decision.

Suggested Inventory Order					
7/14/2002, 10:22 AM					
<u>Product ID</u>	<u>Product Name</u>	<u>Sold</u>	<u>On Hand</u>	<u>Build To</u>	<u>Order</u>
17221850001	Belt Clip Holster, Plastic		10	100	90
IBR-1	In-Building Repeater	1	7	25	18

Suggested inventory order report

Search to Sales Conversion

Provides the number of times each product has shown up in search results, the number of times the product detail content page has been viewed, the number of times the product has been added to a shopping cart, and the quantity of the product that has been sold. Percentages that show the conversion rate between each of the columns are also provided.

Use this report to provide insight into which products are more successful in making it through the entire search to view to shopping cart to checkout process. This may provide useful information to determine how to modify products that have had limited success in making it through the entire process.

Note that the **Sold** column contains the quantity sold and the **Carts** column contains the number of times the product has been added to a shopping cart. The percentage between the two will give some insight into the quantity purchased per shopping cart. For example, a **Carts** column value of **2** and a **Sold** column value of **4** would indicate that on average two units of the product are sold for each time it is added to a shopping cart.

Search to Sales Conversion									
7/14/2002, 10:25 AM									
ProductID ▲▼	Product Name ▲▼	Searches ▲▼	Views ▲▼	Carts ▲▼	Sold ▲▼	Views to Cart	Cart to Sold	Views to Sold	
SNN5717	Slim Li-Ion Battery	15	5	2	2	40.00%	100.00%	40.00%	
BP2	"Al'armi, pensieri"	26	3	2	1	66.67%	50.00%	33.33%	
1272	"Parto! Ma tu ben mio" from La Clemenza di Tito	26	2						
RLN5450	"Refresh" Flip Style Carry Case	21	1						
NNTN4176	"Refresh" Holster for New Model Inward facing	21	1						
NTN9861	"Refresh" Holster for New Model Outward Facing	21	3	1		33.33%			
Floppy-1	1.44MB Floppy Drive	35	7	1	6	14.29%	600.00%	85.71%	

Search to sales conversion report

Affiliate Referrals Conversion

Provides the number of times a hyperlink with an affiliate ID has been used to enter the store. The report further breaks down the number of orders that result, percentages, etc to help determine the success of individual affiliates and the affiliate program in general.

Affiliate Referrals Conversion							
Total Sales: \$40,953.66							
<u>Affiliate</u>	<u>Referrals</u>	<u>Orders</u>	<u>Percentage</u>	<u>SubTotal</u>	<u>Percentage</u>	<u>Per Link</u>	<u>Per Order</u>
<i>Uncategorized</i>	0	2		\$664.01	1.62%		
HostedStore	25	5	20.00%	\$39,953.53	97.56%	\$1,598.14	\$7,990.71
SampleAffiliate	0	1		\$336.12	0.82%		

Affiliate referrals conversion report

General Store Statistics

Provides a summary of orders, customers, affiliates, and products. Orders are summarized by the number of orders, number of ordered items, discounts, and the number of customers and affiliates. Customers are summarized by number of orders, address books, wish lists, and other criteria. Affiliates are summarized by their current status and number of orders. Products are summarized by their status and type. The cumulative total of search, view, shopping cart, and sold data (same statistics as the **Search to Sale Conversion** report) for products are also shown.

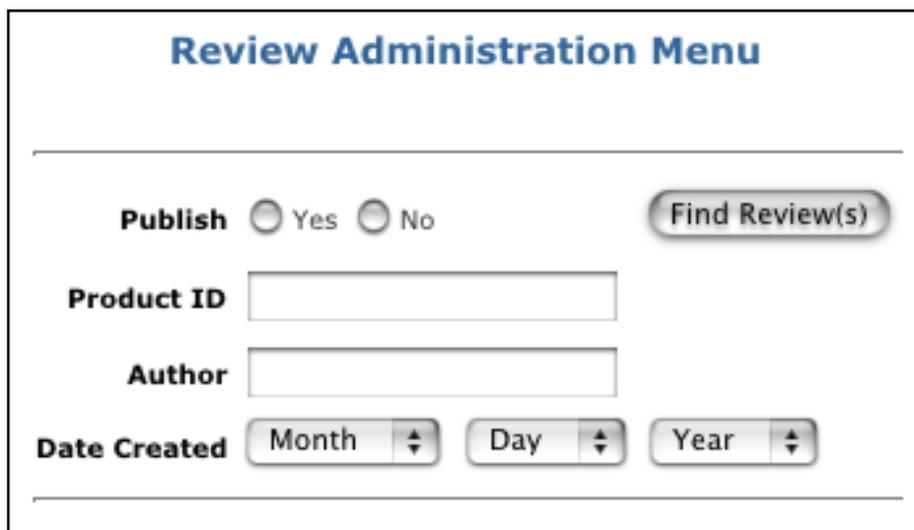
This report can be used in many different ways. One use is to determine how individual features in the store are being utilized. For example, the number of address books and wish lists provides insight into whether or not those features are being used by the existing customer base. If not, then maybe the customers need to be made aware of the feature and the advantage of using it.

General Store Statistics				
7/14/2002, 10:23 AM				
Order Summary				
	Quantity	Percentage	SubTotal	Average
Orders	44		\$73,210.50	\$1,663.88
Ordered Items	202		\$73,210.50	\$362.43
Items per Order				4.59
Promotion Discounts	0		\$0.00	\$0.00
SubTotal Adjustments	1	2.27%	\$-29.50	\$-29.50
Shipping Adjustments	0		\$0.00	\$0.00
Tax Adjustments	0		\$0.00	\$0.00
Discount Adjustments	0		\$0.00	\$0.00
Customers with Orders	16		\$73,210.50	\$4,642.30
Affiliates with Orders	2	100.00%	\$66,091.41	\$33,045.70
Customer Summary				
	Quantity	Percentage	SubTotal	Average
Customers with Orders	16		\$73,210.50	\$4,642.30
Orders per Customer				8.80
Ordered Items per Customer				40.40
Customers with Address Books	5	17.24%		
Address Book Entries	5			1.00
Customers with Wishlists	4	13.79%		
Total Wishlist Items	12			3.25
Purchased Wishlist Items	5	39.46%		
Non-Purchased Wishlist Items	8	61.54%		
Affiliate Summary				
	Quantity	Percentage	SubTotal	Average
Total Affiliates	2			
Active Affiliates	2	100.00%		
Inactive Affiliates	0			
Suspended Affiliates	0			
Affiliates with Orders	2	100.00%		
Orders per Affiliate				5.00
Affiliate Orders	10		\$66,091.41	\$33,045.70
Affiliate Ordered Items	156		\$66,091.41	\$423.66
Product Summary				
	Quantity	Percentage		
Products in Catalog	2,319			
Selling	2,275	96.85%		
Featured	6	0.26%		
On Sale	6	0.34%		
Memberships	2	0.09%		
Member Only Items	1	0.04%		
Searches	12,004			
Views	5,596			
Cart Placements	1,006			
Sold	1,002,750			

General store statistics report

Review Administration

The Review Administration menu provides for one operation: Find Review(s).



The screenshot shows a web form titled "Review Administration Menu". It contains the following elements:

- Publish**: Two radio buttons labeled "Yes" and "No".
- Find Review(s)**: A button with a drop shadow.
- Product ID**: A text input field.
- Author**: A text input field.
- Date Created**: Three dropdown menus labeled "Month", "Day", and "Year".

Review administration menu

Sections

- [Find Review\(s\)](#) - Search for existing reviews.
- [Edit Review](#) - Explanation of review edit form.

Find Review(s)

A search form is provided to search for existing reviews. If more than one match is found, then a listing of all matching reviews will be provided with the author name appearing as a hyperlink to the edit review form. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching review is displayed in an edit review form.

The fields that can be searched are listed below along with the operator used in the search:

- **Publish** - Equals (eq)
- **ProductID** - Equals (eq)
- **Author** - Contains (cn)
- **DateCreated** - Begins With (bw)
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The screenshot shows a search form with the following elements:

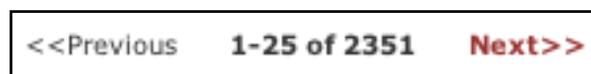
- Publish**: Two radio buttons labeled "Yes" and "No".
- Product ID**: A text input field.
- Author**: A text input field.
- Date Created**: Three dropdown menus labeled "Month", "Day", and "Year".
- Find Review(s)**: A button.

Review search form area of the review administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate review in the listing and click the **Author** hyperlink for the review. The review will be loaded into the edit form.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.



Review found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the author name to edit.

<u>Publish</u> ▲▼	<u>Product ID'</u> ▲▼	<u>Author</u> ▲▼	<u>Date/Time Posted</u> ▲▼
Y	1505883Z01	Anonymous	7/15/2002, 10:42 AM
Y	2000R35100	Anonymous	5/2/2002, 9:45 AM
Y	2000R35416	Anonymous	1/10/2002, 5:28 PM
Y	2000R35416	Anonymous	1/10/2002, 8:01 PM
Y	2000R35500	Anonymous	11/11/2001, 00:49 AM
Y	2000R35901	Anonymous	11/23/2001, 3:35 AM
Y	2000R35HM02	Anonymous	10/30/2001, 3:29 AM
Y	2000R35HM02	Anonymous	3/31/2002, 2:41 PM
Y	2000R35HM82	Anonymous	11/23/2001, 1:34 AM
Y	SGCD0015801	Anonymous	11/10/2002, 9:32 PM

Review search results list

Edit Review

All of the fields on this form correspond to fields in the **Review** table only. An explanation of each field and its corresponding field in the **Review** table are detailed below:

- **Publish** corresponds to the field of the same name in the table (*Required*). Set to *Yes* if the review should be viewable by customers. Set to *No* to temporarily or permanently prevent customers from viewing the review.

By default, when a review is submitted by a customer it will have a **Publish** value of *Yes*. If the `HS_AutoPublishReviews` preference variable is set to a value of *N* in *Store Configuration*, then reviews will be created with a **Publish** value of *No* and will not be viewable until manually flagged for publishing in *Review Administration*.

- **Review Text** corresponds to the **ReviewText** field in the table (*Required*). This field holds the actual review text that is viewed by customers in the store.

Important: If reviews are subject to review and editing before or after being submitted, it may be necessary to post an appropriate disclaimer on the Web site.

Edit Review

* denotes an empty required field.

Publish Yes No

Review Text

This is the most comfortable chair I have ever sat in--definitely worth the price!

Oh, and the service of HostedStore is next to none. I called them several times asking about options, availability, and shipping costs. Each time I was greeted kindly and my questions were answered thoroughly. I'm sure they are sick of me by now, but too bad because they have a customer for life!

Update Review

Edit review form

Store Status

The store status page provides a summary overview of store activity at any given time.

Store Status The store is Open .																																																																																																	
Cart Summary 10 active carts. 11 items with a total value of \$8,423.98.																																																																																																	
Order Queue	<table border="1"><thead><tr><th>Invoice</th><th>Status</th><th>Items</th><th>SubTotal</th><th>Date/Time Created</th><th>Customer ID</th></tr></thead><tbody><tr><td>279</td><td>Pending</td><td>1</td><td>\$439.95</td><td>7/22/2002, 9:37 AM</td><td>63 (New Customer)</td></tr><tr><td>278</td><td>Pending</td><td>2</td><td>\$639.90</td><td>7/22/2002, 9:33 AM</td><td>15 (show orders)</td></tr><tr><td>277</td><td>Pending</td><td>4</td><td>\$1,598.39</td><td>7/22/2002, 9:01 AM</td><td>15 (show orders)</td></tr><tr><td>276</td><td>Pending</td><td>2</td><td>\$335.39</td><td>7/16/2002, 2:34 PM</td><td>15 (show orders)</td></tr><tr><td>275</td><td>Pending</td><td>2</td><td>\$32.40</td><td>7/16/2002, 7:23 AM</td><td>15 (show orders)</td></tr><tr><td>274</td><td>Pending</td><td>1</td><td>\$329.99</td><td>7/15/2002, 4:57 PM</td><td>15 (show orders)</td></tr><tr><td>271</td><td>Pending</td><td>1</td><td>\$329.99</td><td>7/15/2002, 4:17 PM</td><td>15 (show orders)</td></tr><tr><td>268</td><td>Pending</td><td>1</td><td>\$199.95</td><td>7/14/2002, 2:42 PM</td><td>15 (show orders)</td></tr><tr><td>265</td><td>Approved</td><td>1</td><td>\$10.10</td><td>7/13/2002, 7:26 AM</td><td>15 (show orders)</td></tr><tr><td>263</td><td>Back Ordered</td><td>1</td><td>\$10.10</td><td>7/12/2002, 9:22 AM</td><td>52 (show orders)</td></tr><tr><td>262</td><td>Approved</td><td>3</td><td>\$30.33</td><td>7/12/2002, 8:35 AM</td><td>52 (show orders)</td></tr><tr><td>258</td><td>Approved</td><td>0</td><td>\$0.00</td><td>7/11/2002, 9:09 PM</td><td>52 (show orders)</td></tr><tr><td>257</td><td>Mixed Status</td><td>2</td><td>\$20.23</td><td>7/11/2002, 3:15 PM</td><td>52 (show orders)</td></tr><tr><td>232</td><td>Back Ordered</td><td>1</td><td>\$23.25</td><td>6/7/2002, 11:22 AM</td><td>15 (show orders)</td></tr><tr><td>229</td><td>Back Ordered</td><td>1</td><td>\$10.10</td><td>6/6/2002, 12:37 PM</td><td>15 (show orders)</td></tr></tbody></table>	Invoice	Status	Items	SubTotal	Date/Time Created	Customer ID	279	Pending	1	\$439.95	7/22/2002, 9:37 AM	63 (New Customer)	278	Pending	2	\$639.90	7/22/2002, 9:33 AM	15 (show orders)	277	Pending	4	\$1,598.39	7/22/2002, 9:01 AM	15 (show orders)	276	Pending	2	\$335.39	7/16/2002, 2:34 PM	15 (show orders)	275	Pending	2	\$32.40	7/16/2002, 7:23 AM	15 (show orders)	274	Pending	1	\$329.99	7/15/2002, 4:57 PM	15 (show orders)	271	Pending	1	\$329.99	7/15/2002, 4:17 PM	15 (show orders)	268	Pending	1	\$199.95	7/14/2002, 2:42 PM	15 (show orders)	265	Approved	1	\$10.10	7/13/2002, 7:26 AM	15 (show orders)	263	Back Ordered	1	\$10.10	7/12/2002, 9:22 AM	52 (show orders)	262	Approved	3	\$30.33	7/12/2002, 8:35 AM	52 (show orders)	258	Approved	0	\$0.00	7/11/2002, 9:09 PM	52 (show orders)	257	Mixed Status	2	\$20.23	7/11/2002, 3:15 PM	52 (show orders)	232	Back Ordered	1	\$23.25	6/7/2002, 11:22 AM	15 (show orders)	229	Back Ordered	1	\$10.10	6/6/2002, 12:37 PM	15 (show orders)
Invoice	Status	Items	SubTotal	Date/Time Created	Customer ID																																																																																												
279	Pending	1	\$439.95	7/22/2002, 9:37 AM	63 (New Customer)																																																																																												
278	Pending	2	\$639.90	7/22/2002, 9:33 AM	15 (show orders)																																																																																												
277	Pending	4	\$1,598.39	7/22/2002, 9:01 AM	15 (show orders)																																																																																												
276	Pending	2	\$335.39	7/16/2002, 2:34 PM	15 (show orders)																																																																																												
275	Pending	2	\$32.40	7/16/2002, 7:23 AM	15 (show orders)																																																																																												
274	Pending	1	\$329.99	7/15/2002, 4:57 PM	15 (show orders)																																																																																												
271	Pending	1	\$329.99	7/15/2002, 4:17 PM	15 (show orders)																																																																																												
268	Pending	1	\$199.95	7/14/2002, 2:42 PM	15 (show orders)																																																																																												
265	Approved	1	\$10.10	7/13/2002, 7:26 AM	15 (show orders)																																																																																												
263	Back Ordered	1	\$10.10	7/12/2002, 9:22 AM	52 (show orders)																																																																																												
262	Approved	3	\$30.33	7/12/2002, 8:35 AM	52 (show orders)																																																																																												
258	Approved	0	\$0.00	7/11/2002, 9:09 PM	52 (show orders)																																																																																												
257	Mixed Status	2	\$20.23	7/11/2002, 3:15 PM	52 (show orders)																																																																																												
232	Back Ordered	1	\$23.25	6/7/2002, 11:22 AM	15 (show orders)																																																																																												
229	Back Ordered	1	\$10.10	6/6/2002, 12:37 PM	15 (show orders)																																																																																												
New Customers	<table border="1"><thead><tr><th>CustomerID</th><th>Name</th><th>Date/Time Created</th></tr></thead><tbody><tr><td>63</td><td>Doe, Jimmy</td><td>7/22/2002, 9:37 AM</td></tr></tbody></table>	CustomerID	Name	Date/Time Created	63	Doe, Jimmy	7/22/2002, 9:37 AM																																																																																										
CustomerID	Name	Date/Time Created																																																																																															
63	Doe, Jimmy	7/22/2002, 9:37 AM																																																																																															
New Orders	<table border="1"><thead><tr><th>Invoice</th><th>Status</th><th>Items</th><th>SubTotal</th><th>Date/Time Created</th></tr></thead><tbody><tr><td>277</td><td>Pending</td><td>4</td><td>\$1,598.39</td><td>7/22/2002, 9:01 AM</td></tr><tr><td>278</td><td>Pending</td><td>2</td><td>\$639.90</td><td>7/22/2002, 9:33 AM</td></tr><tr><td>279</td><td>Pending</td><td>1</td><td>\$439.95</td><td>7/22/2002, 9:37 AM</td></tr></tbody></table>	Invoice	Status	Items	SubTotal	Date/Time Created	277	Pending	4	\$1,598.39	7/22/2002, 9:01 AM	278	Pending	2	\$639.90	7/22/2002, 9:33 AM	279	Pending	1	\$439.95	7/22/2002, 9:37 AM																																																																												
Invoice	Status	Items	SubTotal	Date/Time Created																																																																																													
277	Pending	4	\$1,598.39	7/22/2002, 9:01 AM																																																																																													
278	Pending	2	\$639.90	7/22/2002, 9:33 AM																																																																																													
279	Pending	1	\$439.95	7/22/2002, 9:37 AM																																																																																													

Store status summary

The first section reports whether or not the store is open or closed. Details about setting the store status can be found in the [Installation and Configuration](#) > [Store Configuration](#) section.

Store Status The store is Open .
--

Store status summary

The number of active shopping carts, the number of items in the carts, and the total value of the items. Note that the numbers do not necessarily reflect the number of current shoppers in the store. Some of the shopping carts may be very old depending upon if old shopping carts have been purged recently. Details about purging shopping carts can be found in the [Store Administration](#) > [Maintenance](#) section.

Cart Summary 27 active carts.
61 items with a total value of \$42,756.37.

Cart summary

All orders that have an open status (not shipped, canceled, or declined) will be showed in the order queue. If a customer is new, then an appropriate note will be displayed next to their customer ID. Otherwise, a hyperlink to all of the customer's past orders will be provided. Invoice numbers and customer IDs are hyperlinked to the edit order form in *Order Administration* and the edit customer form in *Customer Administration*, respectively, so that administration can be done quickly without having to perform a search. This process can also be made easier by opening the hyperlinks in new browser windows so that a return to the store status page is not necessary.

Order Queue	<u>Invoice</u>	<u>Status</u>	<u>Items</u>	<u>SubTotal</u>	<u>Date/Time Created</u>	<u>Customer ID</u>
	372	Pending	1	\$22.49	6/9/2003, 6:17 AM	72 (show orders)
	370	Pending	4	\$3,099.96	5/29/2003, 5:25 AM	15 (show orders)
	369	Pending	2	\$32.59	5/28/2003, 9:31 PM	70 (show orders)

Order queue summary

All new customers for the day are detailed in a fourth section. Customer IDs are hyperlinked to the edit customer form in *Customer Administration* so that administration can be done quickly without having to perform a search. This process can also be made easier by opening the hyperlinks in new browser windows so that a return to the store status page is not necessary.

New Customers	<u>CustomerID</u>	<u>Name</u>	<u>Date/Time Created</u>
	72	Burns, Frank	6/9/2003, 6:16 AM

New customer summary

All new orders for the day are detailed in the last section. Invoice numbers are hyperlinked to the edit order form in *Order Administration* so that administration can be done quickly without having to perform a search. This process can also be made easier by opening the hyperlinks in new browser windows so that a return to the store status page is not necessary.

New Orders	<u>Invoice</u>	<u>Status</u>	<u>Items</u>	<u>SubTotal</u>	<u>Date/Time Created</u>
	372	Pending	1	\$22.49	6/9/2003, 6:17 AM

New order summary

Supplier Administration

Suppliers can be created simply to maintain contact information for product vendors. However, the main intent of this feature is to assign permissions and preferences for suppliers that will drop ship products. Suppliers with the proper permissions will be able to log into their own *Supplier Account Administration* area, view orders in their shipping queue, change the status of ordered items, and assign shipping vendors and tracking numbers. More information can be found in the [The Supplier's Perspective](#) section.

The *Supplier Account Administration* menu provides for three different operations: *Email Supplier(s)*, *Add New Supplier*, and *Find Supplier(s)*.



The image shows a screenshot of the 'Supplier Administration Menu'. At the top, the title 'Supplier Administration Menu' is centered. Below the title, there are three buttons: 'Email Supplier(s)', 'Add New Supplier', and 'Find Supplier(s)'. The 'Find Supplier(s)' button is positioned to the right of a search form. The search form consists of several input fields: 'Supplier ID', 'Last Name', 'Company', 'City', 'State', and 'Country'. The 'State' and 'Country' fields are dropdown menus. Below the search fields, there is an 'Operator' section with two radio buttons: 'And' (which is selected) and 'Or'.

Supplier administration menu

Sections

- [Email Supplier\(s\)](#) - Send email to individual suppliers or all suppliers.
- [Add New Supplier](#) - Create a new supplier.
- [Find Supplier\(s\)](#) - Search for and edit existing suppliers.

Email Supplier(s)

Used to email all suppliers or an individual supplier. This function allows for email-based communication with suppliers without having to use an email client application.

Important: None of the correspondence sent through this interface is saved in the table so if a copy of the email is wanted be sure to check the box to send a copy to the sending email address or BCC an email address that is used for archiving email communication with suppliers.

Click the **Email Supplier(s)** button to load a new page with a form for creating and sending the email.



Email suppliers form area of the supplier administration menu

The fields of the email suppliers form are described below:

- **From** - Automatically loaded with the sales email address set as the `HS_SalesEmail` preference variable in *Store Configuration*. The field is a text box so the email address can be changed if necessary.
- **To** - A select list is loaded with one option plus a single entry for each supplier.
 - *All suppliers* will send the email to all suppliers in the **Supplier** table.
 - The last set of entries is an alphabetized list of suppliers using fields from the **Supplier** table in the following format: **Company (ID)** Selecting one of the entries will send the email to that supplier only. Note that if the store has hundreds of suppliers this form will probably not load because the select list will be too large. So don't do it!
- **BCC** - Enter an email address to send a blind carbon copy of the email to the specified recipient.
- **Send me a copy of the email** - Check the checkbox to send a blind carbon copy to the email address in the **To** field.
- **Subject** - Subject line of the email.
- **Message** - The body of the email. Note that only text email is supported in this interface so any HTML in the body will render as text.

* denotes an empty required field.

* **From**

* **To**

BCC

Send me a copy of the email.

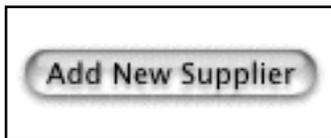
* **Subject**

* **Message**

Email suppliers form

Add New Supplier

Create a new supplier by clicking the **Add New Supplier** button. A new page with a form for adding supplier information will be loaded in the browser.

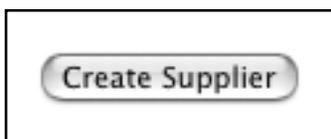


Add supplier form area of the supplier administration menu

Add Supplier Form

All of the fields on this form correspond to fields in the **Supplier** table only. In addition to basic contact information, multiple parameters are used to set preferences for suppliers that will drop ship orders. An explanation of each field and its corresponding field in the **Supplier** table are detailed by group below.

After all fields have been completed, click the **Create Supplier** button at the bottom of the form to add the supplier to the database.



Create supplier button on the add supplier form

Contact Information

- **Active** corresponds to the **Status** field in the table (*REQUIRED*). Designates whether or not the supplier is currently a vendor for store products. Set to **Y** if the supplier is supplying products directly or via drop shipping.
- **First Name** corresponds to the **FirstName** field in the table (*REQUIRED*).
- **Last Name** corresponds to the **LastName** field in the table (*REQUIRED*).
- **Company** corresponds to the field of the same name in the table (*REQUIRED*).
- **Address** and the second unlabeled address field correspond to the **Address1** and **Address2** fields in the table, respectively (*REQUIRED*).
- **City** corresponds to the field of the same name in the table (*REQUIRED*).
- **State** corresponds to the field of the same name in the table (*REQUIRED*).
- **Postal Code** corresponds to the **Postal** field in the table (*REQUIRED*).
- **Country** corresponds to the field of the same name in the table (*REQUIRED*).
- **Email** corresponds to the field of the same name in the table (*REQUIRED*). The username for the supplier when logging into *Supplier Account Administration*.
- **Phone** corresponds to the field of the same name in the table (*REQUIRED*).
- **Fax** corresponds to the field of the same name in the table.
- **Web Site** corresponds to the **URL** field in the table.
- **Password** corresponds to the field of the same name in the table (*REQUIRED*). The password for the supplier when logging into *Supplier Account Administration* if the supplier will be drop shipping products.
- **Allow Login** corresponds to the **AllowLogin** field in the table (*REQUIRED*). Designates whether or not the supplier will have access to the order shipping interface. Set to **Y** if the supplier will be drop shipping some or all of the products which they supply to the store. Details about the order shipping interface can be found in the [The Supplier's Perspective](#) section.

* Active	<input type="radio"/> Yes <input type="radio"/> No
* First Name	<input type="text"/>
* Last Name	<input type="text"/>
* Company	<input type="text"/>
* Address	<input type="text"/>
	<input type="text"/>
* City	<input type="text"/>
* State	<input type="text"/>
* Postal Code	<input type="text"/>
* Country	<input type="text"/>
* Email	<input type="text"/>
* Phone	<input type="text"/>
Fax	<input type="text"/>
Web Site	<input type="text"/>
* Password	<input type="text"/>
* Allow Login	<input type="radio"/> Yes <input type="radio"/> No

Contact information area of the add supplier form

Order Notification

Preferences for email notification messages.

- **Email Address** corresponds to the **NotifyEmail** field in the table (*REQUIRED* if **Allow Login** has a value of *Y*). The supplier email address to which emails will be sent whenever an order has been processed and is ready to be shipped .
- **Email Template** corresponds to the **NotifyEmailTemplate** field in the table. The email body format file that will be used for the order notification sent to the supplier. The name of the file should be relative to the `/hostedstore/adminstore/email/` directory. So, for example, if a directory named `supplier_emails` is created in the `/hostedstore/adminstore/email/` directory to store multiple templates and the template name for the supplier is `notify_supplier.lasso`, then `supplier_emails/notify_supplier.lasso` should be entered in the field. If a template is not specified, then the default `email_supplier.lasso` file will be used.

<u>Order Notification</u>	
Email Address	<input type="text"/>
Email Template	<input type="text"/>

Order notification area of the add supplier form

Supplier Administration

Multiple preferences can be set that allow for each supplier to have a customized *Supplier Account Administration* interface.

- **Admin Template** corresponds to the **SupplierAdminTemplate** field in the table. The name of the template file to use for *Supplier Account Administration*. The name of the file should be relative to the `/hostedstore/adminsupplier/templates/` directory. So, for example, if a template named `HostedStore` is to be used, then `HostedStore` should be entered in the field. If a template is not specified, then the *default* template will be used.
- **Customer Email Template** corresponds to the **CustomerEmailTemplate** field in the table. The email body format file that will be used for the order shipping notification sent by the supplier to customers. The name of the file should be relative to the `/hostedstore/adminsupplier/email/` directory. So, for example, if a directory named `HostedStore` is created in the `/hostedstore/adminsupplier/email/` directory to store the template for a supplier and the template name for the supplier is `email_customerstatus.lasso`, then `HostedStore/email_customerstatus.lasso` should be entered in the field. If a template is not specified, then the default `email_customerstatus.lasso` file will be used.
- **Store Email Template** corresponds to the **StoreEmailTemplate** field in the table. The email body format file that will be used for the order shipping notification sent by the supplier to the store. The name of the file should be relative to the `/hostedstore/adminsupplier/email/` directory. So, for example, if a directory named `HostedStore` is created in the `/hostedstore/adminsupplier/email/` directory to store the template for a supplier and the template name for the supplier is `email_storestatus.lasso`, then `HostedStore/email_storestatus.lasso` should be entered in the field. If a template is not specified, then the default `email_storestatus.lasso` file will be used.
- **Store Email Address** corresponds to the **StoreEmailAddress** field in the table. The store email address to which emails will be sent whenever an order status has been changed by the supplier. If an email address is not specified, then an email will not be sent to the store.

<u>Supplier Administration</u>	
Admin Template	<input type="text"/>
Customer Email Template	<input type="text"/>
Store Email Template	<input type="text"/>
Store Email Address	<input type="text"/>

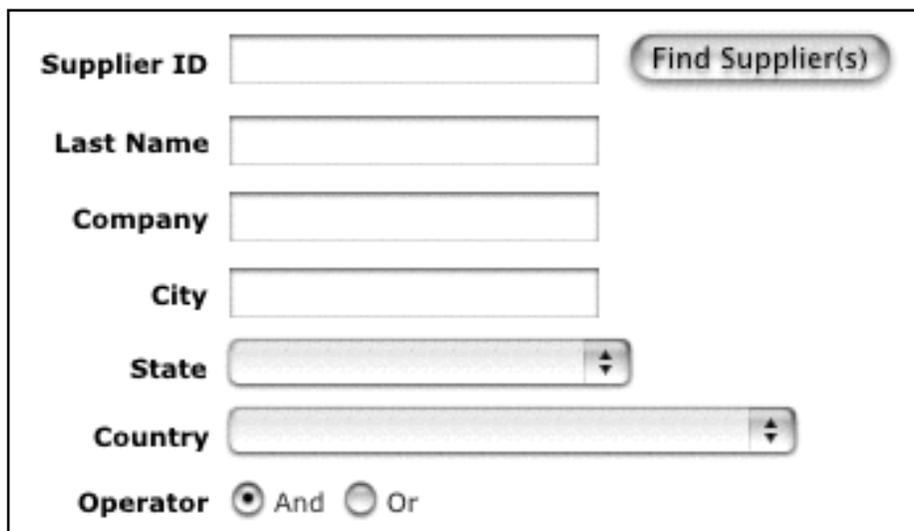
Supplier administration area of the add supplier form

Find Supplier(s)

A search form is provided to search for existing suppliers. If more than one match is found, then a list of all matching suppliers will be provided. Each of the columns can be sorted in ascending or descending order. If only one match is found then the matching supplier is displayed in an edit supplier form. The fields for editing a supplier are the same as the fields available when adding a new supplier.

The fields that can be searched are listed below along with the operator used in the search:

- **ID** - Equals (eq)
- **LastName** - Contains (cn)
- **Company** - Contains (cn)
- **City** - Equals (eq)
- **State** - Equals (eq)
- **Country** - Equals (eq)
- **Operator** - Allows for the search logical operator (-OpLogical) to be set to an AND or OR type search. An AND search must meet all of the specified search criteria. An OR search may meet any of the specified search criteria.
- **Limit** - The maximum number of records (-MaxRecords) to return. If the number of records found exceeds the value set for the limit, then Previous/Next navigation will be displayed on the search results list so the found set can be navigated. If a value is not provided, then all matching records will be returned.



The image shows a search form for suppliers. It includes input fields for Supplier ID, Last Name, Company, City, State, and Country. The State and Country fields are dropdown menus. There is a radio button for the Operator, with 'And' selected and 'Or' unselected. A 'Find Supplier(s)' button is located to the right of the Supplier ID field.

Supplier search form area of the supplier administration menu

Search Results List

If more than one result is returned from the search, then find the appropriate supplier in the listing and click the **Supplier ID** hyperlink for the supplier. The supplier will be loaded into the edit form. The form fields and process for editing suppliers is similar to the process for adding a supplier detailed in the [Store Administration](#) > [Supplier Administration](#) > [Add New Supplier](#) section.

If a value was set for **Limit** on the search form and the number of matching records exceeds the limit value, then Previous/Next navigation will appear at the top and bottom of the search results list. A summary of the records displayed in relation to the found set is displayed between the Previous and Next links.

Supplier found set navigation

The sort column and sort order for the search results list can be modified by clicking the up arrow next to the appropriate column for an ascending sort order or by clicking the down arrow next to the appropriate column for a descending order. Alternatively, the column name itself can be clicked to change the sort order for the column. If a column is sorted in ascending order, a single quote mark (') will be displayed to the right of the column name. If a column is sorted in descending order, a double quote mark (") will be displayed to the right of the column name.

Click on the supplier ID to edit the supplier record.

<u>Supplier ID</u> ▲▼	<u>Last Name'</u> ▲▼	<u>Company</u> ▲▼	<u>City</u> ▲▼	<u>State</u> ▲▼	<u>Country</u> ▲▼
1	Doe	HostedStore Corporation	Bellevue	WA	US
2	Hetfield	Test Company	Los Angeles	CA	US

Supplier search results list

Shipping and Taxation

Shipping charges and tax charges are calculated via two included files. The code in each file may need to be modified to meet the policies of the store. An explanation of the default code in each file is provided below. If Lasso programming is unfamiliar, it is highly recommended to seek the aid of a Lasso developer or ask questions on the HostedStore support list.

Sections

- [Shipping](#) - Script used to calculate shipping costs.
- [Taxation](#) - Script used to calculate tax cost.

Shipping

/hostedstore/includes/calculations/calculate_shipping.lasso

This file is called for each product in the shopping cart every time that shipping costs need to be calculated. Shipping is calculated prior to displaying each of the following content pages:

- Default cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- Default order review content page (*/hostedstore/storefront/html/order_display.lasso*).
- Order review page in store administration.

Public Variables Inputs

Twelve different public variables are available for calculating shipping costs. If additional information is needed, then the appropriate code can be added to the file. The public variables are summarized below:

- **HS_ShippingQuantityRate** - Set as a preference variable in *Store Configuration*. This variable contains the shipping cost for products using the arbitrary quantity based shipping.
- **HSP_ItemShippingType** - The type of shipping calculation to use for the product.
- **HSP_ItemShippingValue** - The value to be used in the shipping calculation for the product.
- **HSP_ItemQuantity** - The quantity of the product in the shopping cart.
- **HSP_ShippingCity** - The city to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingState** - The state to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingCountry** - The country to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingPostal** - The postal code to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingCity** - The city in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingState** - The state in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingCountry** - The country in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingPostal** - The postal code in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).

Public Variables Outputs

One variable must be passed back from the shipping calculation script so that the total shipping cost for the order can be properly calculated. The final value for this variable after all script calls will be used as the total cost of shipping so the variable should be used to accumulate the cost of shipping for all products.

- **HSP_CartShippingCost** - This variable contains the shipping cost for the products.

Default Script

HostedStore is installed with a default shipping calculation script. This script is provided as an example of how to write a script that meets the needs and policies of the store. Review the script and make the modifications necessary. Familiarity with LDML is required to make modifications. If the process of modification is overwhelming, seek out the aid of a Lasso developer or ask questions on the HostedStore support list.

The entire default shipping calculation script is provided below as a reference in case modifications are made and there is a need to return to the default script:

```
<?LassoScript
If:$HSP_ItemShippingType=='Fixed';
    $HSP_CartShippingCost+=($HSP_ItemQuantity
        *$HSP_ItemShippingValue);
Else:$HSP_ItemShippingType=='Quantity';
    $HSP_CartShippingCost+=($HSP_ItemQuantity
        *$HSP_ItemShippingValue
        *$HS_ShippingQuantityRate);
Else:$HSP_ItemShippingType=='Weight';
    If:$HSP_ItemShippingValue<=25.00;
        $HSP_CartShippingCost+=($HSP_ItemQuantity
            *15.00);
    Else:$HSP_ItemShippingValue<=50.00;
        $HSP_CartShippingCost+=($HSP_ItemQuantity
            *20.00);
    Else:$HSP_ItemShippingValue<=100.00;
        $HSP_CartShippingCost+=($HSP_ItemQuantity
            *25.00);
    Else;
        $HSP_CartShippingCost+=($HSP_ItemQuantity
            *50.00);
/If;
/If;
/?>
```

Script Explanation

Products may have one of three different methods for calculating shipping: *Fixed*, *Quantity*, or *Weight*. The script is split into three parts that calculates shipping for each case. Because the script is called once for each product in the shopping cart, the total shipping cost will accumulate in the `HSP_CartShippingCost` variable.

If:\$HSP_ItemShippingType=='Fixed'; handles the first case where the shipping type is *Fixed*. This is the simplest of the shipping types because the shipping cost has already been set for each product by the `HSP_ItemShippingValue` preference variable in *Store Configuration* and the quantity in the shopping cart is known and held in the `HSP_ItemQuantity` variable. Multiply the two values and add to the `HSP_CartShippingCost` variable with **`$HSP_CartShippingCost+=($HSP_ItemQuantity*$HSP_ItemShippingValue);`**

Else:\$HSP_ItemShippingType=='Quantity'; handles the second case where the shipping type is *Quantity* based. Quantity based shipping assigns a fixed shipping cost, set by the `HS_ShippingQuantityRate` preference variable in *Store Configuration*, to products with a quantity value of one. Quantity based shipping is a variation of fixed shipping so the approach to calculation of the shipping cost is similar. Multiplying this value times the quantity value of the item in the shopping cart, `HSP_ItemShippingValue`, and the actual quantity of the item in the cart, `HSP_ItemQuantity`, yields the shipping cost for the item. That calculation is performed and added to the

total cost of shipping, `HSP_CartShippingCost`, with

```
$HSP_CartShippingCost+=$(HSP_ItemQuantity*$HSP_ItemShippingValue*$HS_ShippingQuantityRate);
```

`Else:$HSP_ItemShippingType=='Weight'`; handles the third and final case where the shipping type is *Weight* based. Weight based shipping will require multiple cases to be set up where weights in a certain range cost X monetary units, weights in a different range cost Y monetary units, etc. The default script uses the following weights and costs:

<u>Weight</u>	<u>Cost</u>
0 to 25	15.00
25 to 50	20.00
50 to 100	25.00
100 or more	50.00

For each weight range, the shipping cost is calculated by multiplying the quantity of the item in the shopping cart, `HSP_ItemQuantity`, by the cost of shipping an item in the weight range. For example, the first weight range for items weighing less than 25 (note that there is no designation of what the weight measurement units are--it just needs to be consistent) is designated with `If:$HSP_ItemShippingValue<=25.00;`. The calculation is performed with `$HSP_CartShippingCost+=$(HSP_ItemQuantity*15.00);`. A similar code structure is repeated until all possible ranges have been satisfied.

Note that a lower bound is not checked for any of the weight ranges. This is not necessary because the ranges begin by checking smaller values first. If the weight of the item does not fall into a smaller range, then the next range essentially checks the lower bound by not satisfying the upper bound of the previous range. That probably was not the most elegant explanation, but hopefully it was comprehensible.

Shipping scripts can be considerably more complex than this example script by incorporating destination, handling fees, etc. This script is just provided as an example to get started. If nothing else, at least be sure to check and adjust the weight based shipping costs and ranges if weight based shipping is used.

Taxation

/hostedstore/includes/calculations/calculate_tax.lasso

This file is loaded every time that tax need to be calculated. Tax is calculated prior to displaying each of the following content pages:

- Default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- Default order review content page (*/hostedstore/storefront/html/order_display.lasso*).
- Order review page in *Store Administration*.

Support is added for calculating and displaying tax for several jurisdictions on the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*). This feature was added for users that wanted to give a true total to their customers on the shopping cart display page without having to prod the customer for their city, state, country, etc before checkout. This, along with display of shipping costs on the shopping cart display page, helps eliminate the "sticker shock" at checkout that leads to many incomplete orders. Use of this feature may or may not meet the needs of the store.

Public Variables Inputs

Twelve different public variables are available for calculating tax. If additional information is needed, then the appropriate code can be added to the file. The public variables are summarized below:

- **HS_TaxableStates** - Preference variable set in *Store Configuration* that contains an array of all the states that are subject to taxation.
- **HS_TaxableStatesRates** - Preference variable set in *Store Configuration* that contains an array of all the tax rates for the states that are subject to taxation.
- **HS_TaxShipping** - Preference variable set in *Store Configuration* that will have a value of **y** if shipping costs are taxable.
- **HSP_CartSubTotal** - The total value of all products in the shopping cart.
- **HSP_CartShippingCost** - The total shipping cost for the products in the shopping cart. Note that this value will only be meaningful in the tax calculation if the **HS_TaxShipping** preference variable has been set to a value of **y** in the *Store Configuration*.
- **HSP_ShippingCity** - The city to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingState** - The state to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingCountry** - The country to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_ShippingPostal** - The postal code to which the order will be shipped. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingCity** - The city in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingState** - The state in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingCountry** - The country in the customer's billing address. Note that this variable will not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).
- **HSP_BillingPostal** - The postal code in the customer's billing address. Note that this variable will

not have a value when shipping is calculated for the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*).

- **HSP_PriceGroupTaxExempt** - Has a value of **Y** if the customer belongs to a tax exempt price group.
- **HSP_TaxInformation** - An array of arrays that contains information about ordered items that may be needed for complex tax scenarios that require taxing specific classes of products in different jurisdictions. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ProductID** field for the item from the **Item** table.
 2. The **Quantity** field for the item from the **Item** table as an *INTEGER* type.
 3. The **Price** field for the item from the **Item** table as a *DECIMAL* type.
 4. The **TotalPrice** field for the item from the **Item** table as a *DECIMAL* type.
 5. The **Taxable** field for the item from the **Item** table.
 6. The **TaxClass** field for the item from the **Item** table.
 7. The **ShippingType** field for the item from the **Item** table.
 8. The **ShippingValue** field for the item from the **Item** table as a *DECIMAL* type.

Public Variables Outputs

One variable must be passed back from the tax calculation script so that the total tax for the order can be properly recorded. The variable to pass back will depend upon if the tax is being calculated for the shopping cart display content page or for the order review content page. How can it be determined which scenario is using the tax calculation script? This is easily determined by checking any of the public variables detailed above that begin with **HSP_Shipping**. If the variable has a value, then the customer is in the order review process so pass back the value of **HSP_CartTaxValues**. If the variable does not have a value, then the customer is viewing the shopping cart so pass back the value of **HSP_CartTax**.

Important: Make sure that the variable passed back is appropriate for the situation. If the incorrect variable is passed back, then tax may not be calculated properly. Ever been responsible for paying tax to the government that was not collected from the consumer? Not fun. Do not be someone it happens to! No joke.

- **HSP_CartTax** - This variable contains the tax for the order. Pass back this variable when the customer is in the order review process.
- **HSP_CartTaxValues** - This variable contains an array of taxes for each of the taxable jurisdictions. If the default shopping cart display content page (*/hostedstore/storefront/html/cart_display.lasso*) and the default tax calculation script are used, then the order of tax rates in the array must match the order of the states in the **HS_TaxableStates** preference variable set in the *Store Configuration*.

Default Script

HostedStore is installed with a default tax calculation script. This script is provided as an example of how to write a script that meets the needs and policies of the store. Review the script and make the modifications necessary. Familiarity with LDML is required to make modifications. If the process of modification is overwhelming, seek out the aid of a Lasso developer or ask questions on the HostedStore support list.

The entire default tax calculation script is provided below as a reference in case modifications are made and there is a need to return to the default script:

```
<?LassoScript
// Declare variables.
Var: 'HSP_TempTaxRate'=(Decimal),
     'HSP_TempTax'=(Decimal),
     'HSP_TempTotalWithTax'=(Decimal),
     'HSP_CartTax'=(Decimal),
     'HSP_CartTaxValues'=(Array);,
     'HSP_TaxableStates'=Array,
```

```

'HSP_TaxableStatesRates'=Array;

If:(Var:'HSP_PriceGroupTaxExempt')!='Y';
// Build array of taxable shipping states if necessary.
If:$HS_TaxableStates;
  $HSP_TaxableStates=$HS_TaxableStates->Split:'|';
/If;
// Build array of tax rates.
$HSP_TaxableStatesRates=$HS_TaxableStatesRates->Split:'|';

// If shipping is taxable, add shipping to the taxable total.
If:$HS_TaxShipping=='Y';
  $HSP_CartShippingCost->SetFormat:-Precision=$HS_CurrencyPrecision;
  $HSP_CartTaxableTotal+=(Decimal:(String:$HSP_CartShippingCost));
/If;

// Calculate tax(es).
If:$HSP_ShippingState;

// #####
//
// Shipping state is known.
// Calculate tax for that state only.
//
// #####

// Loop through taxable states to find the matching state.
Loop:($HSP_TaxableStates->Size);
  If:$HSP_ShippingState==($HSP_TaxableStates->Get:Loop_Count);
// This is the matching state.
// Get the tax rate.
  $HSP_TempTaxRate=($HSP_TaxableStatesRates->Get:Loop_Count);
  Loop_Abort;
/If;
/Loop;
// Calculate tax.
  $HSP_CartTax=$HSP_CartTaxableTotal
    *$HSP_TempTaxRate;
Else;

// #####
//
// Shipping state is not known.
// Calculate tax for all taxable states.
//
// #####

// Loop through all taxable states.
Loop:($HSP_TaxableStates->Size);
// Get tax rate.
  $HSP_TempTaxRate=($HSP_TaxableStatesRates->Get:Loop_Count);
// Calculate tax.
  $HSP_TempTax=$HSP_CartTaxableTotal
    *$HSP_TempTaxRate;
// Set display formatting.
  $HSP_CartSubTotal->SetFormat:-Precision=$HS_CurrencyPrecision;
  $HSP_CartShippingCost->SetFormat:-Precision=$HS_CurrencyPrecision;
  $HSP_TempTax->SetFormat:-Precision=$HS_CurrencyPrecision;

```

```
// Calculate order total with tax and shipping.
    $HSP_TempTotalWithTax=(Decimal:(String:$HSP_CartSubTotal))
        +(Decimal:(String:$HSP_CartShippingCost))
        +(Decimal:(String:$HSP_TempTax));
// Format monetary values.
    $HSP_TempTax=(HS_FormatCurrency:$HSP_TempTax);
    $HSP_TempTotalWithTax=(HS_FormatCurrency:$HSP_TempTotalWithTax);
// Populate HSP_CartTaxValues.
// Insert an array containing the name of the state, the tax, and the order total.
    $HSP_CartTaxValues->Insert:(Array:( $HSP_TaxableStates->Get:Loop_Count),
        $HSP_TempTax,
        $HSP_TempTotalWithTax);

    /Loop;
    /If;
/?>
```

Script Explanation

The code is commented as well as any explanation that could be given here, but an explanation follows nonetheless. Note that much of the code in this script could be compacted, but has been written in several steps to make it easier for users to understand how it works and make appropriate modifications.

The code is split into three parts:

1. Declare temporary variables.
2. Set initial values to be used for calculations.
3. Calculate the tax. This section is split into two sections:
 1. Calculate tax for one state only. This section is used during the checkout process when the customer has identified the state to which the order will be shipped.
 2. Calculate tax for all taxable states. This section is used for the cart when the shipping destination is not known. Tax for all taxable states is calculated because research has shown that customers often abandon shopping carts during the checkout process once the total tax and shipping are calculated. The theory is that by displaying the tax on the default shopping cart display content page customers will know the entire order total and will be less likely to abandon the shopping cart during the checkout process.

Variable Declaration

Five variables are set in one compound variable declaration statement. Three are for temporary use in the tax calculation script and two are used to return a value for further processing outside of the tax calculation script.

- `HSP_TempTaxRate` - Decimal value that stores the tax rate for the current state when tax is calculated for multiple states.
- `HSP_TempTax` - Decimal value that stores the tax for the current state when tax is calculated for multiple states.
- `HSP_TempTotalWithTax` - Decimal value that stores the order total (subtotal, shipping, and tax) for the current state when tax is calculated for multiple states.
- `HSP_CartTax` - Decimal value that stores the calculated tax when the shipping destination is known.
- `HSP_CartTaxValues` - Array of arrays that stores the state name, tax, and order total (subtotal, shipping, and tax) for each of the taxable states.

Set Initial Values

This section sets two array variables that are used to store the names of taxable states and their respective tax rates. If shipping is taxable, then the cost of shipping is added to the taxable cart total.

`Var:'HSP_TaxableStates'=$HS_TaxableStates->Split:'|'|'`; builds an array of taxable states by placing the || delimited values of the `HS_TaxableStates` preference variable set in *Store Configuration* into an array.

`Var:'HSP_TaxableStatesRates'=$HS_TaxableStatesRates->Split:'|'|'`; builds an array of taxable state rates by placing the || delimited values of the `HS_TaxableStatesRates` variable (set in the store preferences file) into an array.

The code block beginning with `If:$HS_TaxShipping=='Y'`; checks if shipping is subject to taxation. If it is, then the cost of shipping, `HSP_CartShippingCost`, is added to the taxable total, `HSP_CartTaxableTotal`, with the following line of code: `$HSP_CartTaxableTotal+=$HSP_CartShippingCost;`

Calculate Tax

Tax is calculated during checkout when the shipping destination is known and for the shopping cart display content page where the shipping destination is unknown. The code used to calculate tax differs for each scenario.

Shipping State Known

`If:$HSP_ShippingState;` separates the block of code that executes when the shipping state, `HSP_ShippingState`, has a value.

There may be multiple states subject to taxation, but the rate for only one state is needed. Because the array containing the taxable states, `HSP_TaxableStates`, and the array containing the tax rates, `HSP_TaxableStatesRates`, are in the same order with respect to each other, it is necessary to find the position of the element that matches `HSP_ShippingState` in the `HSP_TaxableStates` array in order to know which element in the `HSP_TaxableStatesRates` array contains the appropriate tax rate.

The code block beginning with `Loop:($HSP_TaxableStates->Size);` will loop through the list of states subject to taxation. The code block beginning with `If:$HSP_ShippingState==($HSP_TaxableStates->Get:Loop_Count);` will execute the code in the block only for the state that matches the shipping state. `$HSP_TempTaxRate=($HSP_TaxableStatesRates->Get:Loop_Count);` gets the tax rate from the `HSP_TaxableStatesRates` array where the position of the element equals the position of the matching state in `HSP_TaxableStates`.

The tax for the order, `HSP_CartTax`, is calculated by multiplying the taxable total, `HSP_CartTaxableTotal`, by the tax rate, `HSP_TempTaxRate`, with `$HSP_CartTax=$HSP_CartTaxableTotal*$HSP_TempTaxRate;`

Shipping State Unknown

The `Else;` condition of the `If:$HSP_ShippingState;` conditional statement separates the block of code that executes when the shipping state, `HSP_ShippingState`, does not have a value.

Three values must be determined for every state in the `HSP_TaxableStates` array:

- The state name.
- The calculated tax for the state.
- The calculated order total (subtotal, shipping, and tax) for the state.

The code block beginning with `Loop:($HSP_TaxableStates->Size);` will loop through the list of states subject to taxation. `$HSP_TempTaxRate=($HSP_TaxableStatesRates->Get:Loop_Count);` gets the tax rate

Payment Processing

HostedStore supports several payment gateways through a built-in processing script and supplied forms. If needed, users can write custom scripts and forms to handle additional payment gateways or payment methods. Alternatively, all requests to support additional gateways will be entertained. An appropriate modification request can be submitted in the support area of the HostedStore Web site (www.hostedstore.com/support/)

Important: Verifying and comparing the monetary totals on all transactions in *Order Administration* and in the online interface provided by the payment gateways is highly recommended. There are malicious people that attempt to find weaknesses in ecommerce applications and payment gateways. If a weakness is found, it could lead to the changing of prices, sending bogus authentication responses, etc. While every attempt has been made to protect against such actions, it does not substitute for common sense and verification by the merchant. Check all totals to make sure the value is correct, check that the totals reported in HostedStore match the totals reported by the payment gateway, and verify that an approved transaction in HostedStore is also reported as approved by the payment gateway.

Sections

- [Built-In Forms](#) - Payment gateways that are supported by HostedStore.
- [Custom Payment Processing](#) - Create a script to integrate with a payment gateway not currently supported by HostedStore.

Built-In Forms

HostedStore handles the behind the scenes processes of validating approval from the payment gateway, updating inventory, sending emails, etc for its supported gateways. All of the forms are located in the */hostedstore/storefront/gateways/* directory. The information that is submitted to the gateway is configurable. For example, some stores may want to send shipping information to a gateway even though it is not required, while others may elect not to do so. Consult the documentation from the payment gateway vendor for specifications on form parameter names and required parameters.

Each form is passed `HSP_Processing`, an array of arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array of fields from the customer billing address record in the **Customer** table. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the customer.
 2. The **LastName** field for the customer.
 3. The **Company** field for the customer.
 4. The **Address1** field for the customer.
 5. The **Address2** field for the customer.
 6. The **City** field for the customer.
 7. The **State** field for the customer.
 8. The **Postal** field for the customer.
 9. The **Country** field for the customer.
 10. The **EmailAddress** field for the customer.
 11. The **Phone** field for the customer.
 12. The **Fax** field for the customer.
 13. The **CustomerID** field for the customer.
2. An array containing shipping address information for the order from the **Invoice** table. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the shipping address.
 2. The **LastName** field for the shipping address.
 3. The **Company** field for the shipping address.
 4. The **Address1** field for the shipping address.
 5. The **Address2** field for the shipping address.
 6. The **City** field for the shipping address.
 7. The **State** field for the shipping address.
 8. The **Postal** field for the shipping address.
 9. The **Country** field for the shipping address.
 10. The **EmailAddress** field for the shipping address.
 11. The **Phone** field for the shipping address.
 12. The **Fax** field for the shipping address.
3. An array containing credit card information for the order from the **Invoice** table. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field.
 2. The **Account** field. This will only have a value for check and credit card payment methods (may be a credit card account number or checking account number depending upon what payment methods are supported and the payment method used).
 3. *NOT USED*
 4. The month portion of the **Expiration** field. This only has a value for a credit card payment method.
 5. The year portion of the **Expiration** field. This only has a value for a credit card payment method.
 6. The **Routing** field. This only has a value for a check payment method.
 7. The **CheckNumber** field. This only has a value for a check payment method.
 8. The **PONumber** field. This only has a value for a purchase order payment method.
4. An array containing additional order information information for the order from the **Invoice** table. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ID** field for the order.

2. The **ItemCount** field for the order.
3. The **SubTotal** field for the order.
4. The **Shipping** field for the order.
5. The **Tax** field for the order.
6. The **Discount** field for the order.
7. The **Total** field for the order.

Edit the appropriate form to enter the URL of the payment gateway, account name, login information, etc. The return hyperlink from the payment gateway to HostedStore should be set to the variable `HSP_PathFromGateway`. That variable will insert the path to the script that will process the reply from the gateway.

Sections

- [Credit Card Processing](#) - Payment gateways to process credit cards.
- [Check Processing](#) - Payment gateways to process checks.
- [PayPal \(IPN version 1.2\)](#) - Payment processing via PayPal.

Credit Card Processing

Documentation for shopping cart integration should have been supplied when the payment gateway account was established or should be available for download from the support area of the payment gateway's Web site. Please consult the payment gateway's documentation to ensure that all necessary information and wanted optional information is properly specified in the form used to post to the gateway.

Forms for several different payment gateways are provided with HostedStore. Please review the form and make sure that it conforms to the payment gateway's specification and the needs of the store. The minimum hidden form elements that HostedStore requires for proper functionality are detailed below for each form. Note that the minimum hidden form elements required by HostedStore may not directly correspond to the minimum hidden form elements required by the payment gateway.

Sections

- [Offline Processing](#)
- [ACH Direct](#)
- [Authorize.Net Advanced Integration Method \(AIM\)](#)
- [BluePay™](#)
- [DirectOne \(Direct Interface\)](#)
- [Echo](#)
- [eProcessing Network](#)
- [E-xact](#)
- [IO•NGATE™](#)
- [LinkPoint Basic](#)
- [NETbilling \(DIRECT MODE\)](#)
- [PayOffshore.com](#)
- [Planet Payment™ ADC Direct Response](#)
- [Plug'nPay™](#)
- [PSiGate](#)
- [SecurePay](#)
- [SkipJack](#)
- [Verisign® Payflow Link \(SM\)](#)
- [Verisign® Payflow Pro \(SM\)](#)

Offline Processing

/hostedstore/storefront/gateways/cc_offline.lasso

Offline processing does not send the payment information to a payment gateway. All of the information is stored internally by HostedStore and then the credit card must be processed manually later via a credit card swipe terminal or the Web interface provided by the payment gateway.

The following items are required by HostedStore for the order processing form:

- The form action should be `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso` and should not be modified.
- A hidden form element with a `name` parameter with a value of `Offline` and a `value` parameter with a value of `[$HSP_Order->Get:10]`. The code should look like: `<input type="hidden" name="Offline" value="[$HSP_Order->Get:10]">`.
- Unlike many other HostedStore forms, the name of the submit button is irrelevant.

ACH Direct (DSI)

/hostedstore/storefront/gateways/cc_achdirect.lasso

Not implemented yet.

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the ACH Direct payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the ACH Direct payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `pg_merchant_id` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `pg_password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `pg_transaction_type` parameter passed to the gateway. Use *11* to pre-authorize a transaction for later capture and processing. Use *10* to immediately process the transaction.
- **4** - The degree of AVS checking to use. This corresponds to the `pg_avs_method` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of *Y*. If declined, then the variable `HSP_Approved` will have a value of *N*. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `pg_response_code` value.
- **AuthorizationCode** - `pg_authorization_code` value.
- **TransactionID** - `pg_trace_number` value.
- **AVSResult** - `pg_avs_result` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Authorize.Net ADC Direct Response

/hostedstore/storefront/gateways/cc_authorizenet_adc.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Authorize.Net payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Authorize.Net payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the AuthorizeNet account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Authorize.Net Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Automated Direct Connect (ADC) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `x_response_code` value.
- **AuthorizationCode** - `x_auth_code` value.
- **TransactionID** - `x_trans_id` value.
- **AVSResult** - `x_avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Authorize.Net Advanced Integration Method (AIM)

/hostedstore/storefront/gateways/cc_authorizenet_aim.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Authorize.Net payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Authorize.Net payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the AuthorizeNet account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **7** - Transaction key. This corresponds to the `x-Tran_Key` parameter passed to the gateway. This value is set by Authorize.Net and can be obtained in the account administration area of the Authorize.Net Web site.
- **8** - Test mode. This corresponds to the `x_Test_Request` parameter passed to the gateway. Use `FALSE` if transactions are to be processed in live mode. Use `TRUE` if transactions are to be processed in test mode.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Authorize.Net Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Advanced Integration Method (AIM) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `x_response_code` value.
- **AuthorizationCode** - `x_auth_code` value.
- **TransactionID** - `x_trans_id` value.
- **AVSResult** - `x_avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

BluePay™

/hostedstore/storefront/gateways/cc_bluepay.lasso

Not implemented yet.

Two steps are necessary for integrating with the BluePay™ payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the **MERCHANT** parameter passed to the gateway.
- **2** - The transaction type. This corresponds to the **TRANSACTION_TYPE** parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: <https://bluepay.onlinedatacorp.com/test/bluepaylitetest.asp>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a **name** parameter with a value of **MERCHANT** and a **value** parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="MERCHANT" value="[$HS_CCParam1]">`.
- A hidden form element with a **name** parameter with a value of **TRANSACTION_TYPE** and a **value** parameter with a value of `[$HS_CCParam2]`. The code should look like: `<input type="hidden" name="TRANSACTION_TYPE" value="[$HS_CCParam2]">`.
- A hidden form element with a **name** parameter with a value of **Order_ID** and a **value** parameter with a value of `[$HSP_Processing->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="Order_ID" value="[$HSP_Processing->Get:4->Get:1]">`.
- A hidden form element with a **name** parameter with a value of **Approved_URL** and a **value** parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="Approved_URL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a **name** parameter with a value of **Declined_URL** and a **value** parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="Declined_URL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a **name** parameter with a value of **Missing_URL** and a **value** parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="Missing_URL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `ApprovalCode` value.
- **AuthorizationCode** - `ApprovalCode` value.
- **TransactionID** - `ApprovalCode` value.
- **AVSResult** - `AVS` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

DirectOne (Direct Interface)

/hostedstore/storefront/gateways/cc_directone.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the ACH Direct payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the DirectOne payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `vendor_name` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `vendor_password` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of *Y*. If declined, then the variable `HSP_Approved` will have a value of *N*. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `summary_code` value.
- **AuthorizationCode** - `response_code` value.
- **TransactionID** - `payment_number` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

Echo

/hostedstore/storefront/gateways/cc_echo.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Echo payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from the Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Echo payment gateway occurs directly rather than via the submission of a form as with some other payment gateways. This is done to protect information that Echo has deemed confidential. The confidential field values are set in the **Credit Card Gateway Parameters** section in *Store Configuration*. The documentation provided by Echo will aid in determining the appropriate values to use for the parameters.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The transaction type. This corresponds to the `transaction_type` parameter passed to the gateway. Use *AS* to pre-authorize a transaction for later capture and processing. Use *AV* to pre-authorize a transaction with address verification for later capture and processing. Use *ES* to immediately process the transaction. Use *EV* to immediately process the transaction with address verification.
- **2** - The order type. This corresponds to the `order_type` parameter passed to the gateway.
- **3** - The merchant ID. This corresponds to the `merchant_echo_id` parameter passed to the gateway.
- **4** - The merchant PIN. This corresponds to the `merchant_pin` parameter passed to the gateway.
- **5** - The merchant email address. This corresponds to the `merchant_email` parameter passed to the gateway.
- **6** - The ISP ID. This corresponds to the `isp_echo_id` parameter passed to the gateway.
- **7** - The ISP PIN. This corresponds to the `isp_pin` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of *Y*. If declined, then the variable `HSP_Approved` will have a value of *N*. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `auth_code` or `decline_code` value depending on if the transaction was approved or declined.
- **AuthorizationCode** - `auth_code` or `decline_code` value depending on if the transaction was approved or declined.
- **TransactionID** - `order_number` value.
- **AVSResult** - `avs_result` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

eProcessing Network

/hostedstore/storefront/gateways/cc_eprocessingnetwork.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Echo payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the eProcessing Network payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant account ID. This corresponds to the `ePNAccount` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - the first comma-delimited value.
- **AuthorizationCode** - the first comma-delimited value.
- **TransactionID** - the first comma-delimited value.
- **AVSResult** - the second comma-delimited value.
- **ApprovalMessage** - A `\r` delimited list of all three comma-delimited values.

E-xact

/hostedstore/storefront/gateways/cc_exact.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the E-xact payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from www.datawebcenter.com. Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the E-xact payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `Post_ExactID` parameter passed to the gateway.
- **2** - The merchant password. This corresponds to the `Post_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `Post_TransactionType` parameter passed to the gateway. Use *PreAuthorization* to pre-authorize a transaction for later capture and processing. Use *Purchase* to immediately process the transaction.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of *Y*. If declined, then the variable `HSP_Approved` will have a value of *N*. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `Rqst_Transaction_Approved` value.
- **AuthorizationCode** - `Rqst_Authorization_Num` value.
- **TransactionID** - `Rqst_CTR` value.
- **AVSResult** - `Rqst_AVS` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

IO•NGATE™

/hostedstore/storefront/gateways/cc_iongate.lasso

As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the **LOGIN** parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: <https://secure.iongate.com/iongate.asp>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a **name** parameter with a value of *RECEIPTURL* and a **value** parameter with a value of *[\$HSP_PathFromGateway]*. The code should look like: `<input type="hidden" name="RECEIPTURL" value="[$HSP_PathFromGateway]">`.
- A hidden form element with a **name** parameter with a value of *INVOICENO* and a **value** parameter with a value of *[\$HSP_Processing->Get:4->Get:1]*. The code should look like: `<input type="hidden" name="INVOICENO" value="[$HSP_Processing->Get:4->Get:1]">`.

Processing Results

If the transaction was successfully processed, then the variable *HSP_Approved* will have a value of *Y*. If declined, then the variable *HSP_Approved* will have a value of *N*. The value of *HSP_Approved* can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - *replycode* value.
- **AuthorizationCode** - *authsource* value.
- **TransactionID** - *transdate* and *transtime* values.
- **AVSResult** - *avsresponse* value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

LinkPoint Basic

/hostedstore/storefront/gateways/cc_linkpoint_basic.lasso

Not implemented yet.

Two steps are necessary for integrating with the LinkPoint payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The store name. This corresponds to the `storename` parameter passed to the gateway.
- **2** - The payment mode. This corresponds to the `mode` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `https://secure.linkpt.net/lpcentral/servlet/lppay`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `storename` and a `value` parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="storename" value="[$HS_CCParam1]">`.
- A hidden form element with a `name` parameter with a value of `mode` and a `value` parameter with a value of `[$HS_CCParam2]`. The code should look like: `<input type="hidden" name="mode" value="[$HS_CCParam2]">`.
- A hidden form element with a `name` parameter with a value of `oid` and a `value` parameter with a value of `[$HSP_Order->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="oid" value="[$HSP_Order->Get:4->Get:1]">`.

NEED TO VERIFY RETURN URLs

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `approval_code` value.
- **AuthorizationCode** - `approval_code` value.
- **TransactionID** - `approval_code` value.
- **AVSResult** - `approval_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

NETbilling (DIRECT MODE)

/hostedstore/storefront/gateways/cc_netbilling.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the NETbilling payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the NETbilling payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The account number. This corresponds to the `GEN_ACCOUNT` parameter passed to the gateway.
- **2** - The Crypto-hash key associated with the account. This value can be obtained from the NETbilling account administration system.
- **3** - The transaction type. This corresponds to the `GEN_TRANS_TYPE` parameter passed to the gateway. Use `AUTH` to pre-authorize a transaction for later capture and processing. Use `SALE` to immediately process the transaction.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `RET_STATUS` value.
- **AuthorizationCode** - `RET_AUTH_CODE` value.
- **TransactionID** - `RET_TRANS_ID` value.
- **AVSResult** - `RET_AVS_MSG` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

PayOffshore.com

/hostedstore/storefront/gateways/cc_payoffshore.lasso

Two steps are necessary for integrating with the PayOffshore.com payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `MerchantID` parameter passed to the gateway.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Account Information > Currency Accounts** tab of the Merchant Panel. Again, consult the payment gateway documentation for the most current configuration information.

- The **Gateway Accepted Address** should be set to the secure path to the store with `/HostedStore.LassoApp` appended to the end. For example, for the domain `www.hostedstore.com`, the entry would look like:
`https://www.hostedstore.com/HostedStore.LassoApp`
- The **Gateway Redirect Address** should be set to the secure path to the store with `/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso` appended to the end. For example, for the domain `www.hostedstore.com`, the entry would look like:
`https://www.hostedstore.com/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

Form URL and Parameters

The URL for the form must be: `https://secure.payoffshore.com/gateway.asp`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `MerchantID` and a `value` parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="MerchantID" value="[$HS_CCParam1]">`.
- A hidden form element with a `name` parameter with a value of `Identifier` and a `value` parameter with a value of `[$HSP_Order->Get:10]`. The code should look like: `<input type="hidden" name="Identifier" value="[$HSP_Order->Get:10]">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - **ERRORCODES** value.
- **AuthorizationCode** - **AUTHCODE** value.
- **TransactionID** - **REFERENCE** value.
- **ApprovalMessage** - A \r delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Planet Payment ADC Direct Response

/hostedstore/storefront/gateways/cc_planetpayment_adc.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Planet Payment payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Planet Payment payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the Planet Payment account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Planet Payment Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Automated Direct Connect (ADC) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `x_response_code` value.
- **AuthorizationCode** - `x_auth_code` value.
- **TransactionID** - `x_trans_id` value.
- **AVSResult** - `x_avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Plug'nPay™

/hostedstore/storefront/gateways/cc_plugnpay.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Plug'nPay payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Plug'nPay payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The account username. This corresponds to the `publisher-name` parameter passed to the gateway.
- **2** - The transaction type. This corresponds to the `auth-type` parameter passed to the gateway. Use `authonly` to pre-authorize a transaction for later capture and processing. Use `authpostauth` to immediately process the transaction.
- **3** - The server name and domain name to use to communicate with the payment gateway server. This value will be assigned by Plug'nPay.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `auth-code` value.
- **AuthorizationCode** - `auth-code` value.
- **TransactionID** - `orderid` value.
- **AVSResult** - `avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

PSiGate

/hostedstore/storefront/gateways/cc_psigate.lasso

As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID number. This corresponds to the `MerchantID` parameter passed to the gateway.
- **2** - The transaction type. This corresponds to the `ChargeType` parameter passed to the gateway. Use `1` to pre-authorize a transaction for later capture and processing. Use `0` to immediately process the transaction.
- **3** - The test property. This corresponds to the `Result` parameter passed to the gateway. Use `0` if the processing is live (not in test mode). Use `1` to simulate an approved transaction. Use `2` to simulate a duplicate transaction. Use `3` to simulate a declined transaction.

Form URL and Parameters

The URL for the form must be set to the URL provided in the PSiGate documentation. The URL is currently: <https://order.psigate.com/psigate.asp>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `MerchantID` and a `value` parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="MerchantID" value="[$HS_CCParam1]">`.
- A hidden form element with a `name` parameter with a value of `ChargeType` and a `value` parameter with a value of `[$HS_CCParam2]`. The code should look like: `<input type="hidden" name="ChargeType" value="[$HS_CCParam2]">`.
- A hidden form element with a `name` parameter with a value of `Result` and a `value` parameter with a value of `[$HS_CCParam3]`. The code should look like: `<input type="hidden" name="Result" value="[$HS_CCParam3]">`.
- A hidden form element with a `name` parameter with a value of `ThanksURL` and a `value` parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="ThanksURL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a `name` parameter with a value of `NoThanksURL` and a `value` parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="NoThanksURL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a `name` parameter with a value of `Oid` and a `value` parameter with a value of `[$HSP_Processing->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="Oid" value="[$HSP_Processing->Get:4->Get:1]">`.
- A hidden form element with a `name` parameter with a value of `Userid` and a `value` parameter with a value of `[$HSP_Processing->Get:1->Get:13]`. The code should look like: `<input type="hidden" name="Userid" value="[$HSP_Processing->Get:1->Get:13]">`.

The default form passes more information than is minimally required by the payment gateway. If there is

extraneous information that should not be passed, then remove the hidden form elements that are not needed. Just be sure that the minimum necessary information is still present or communication with the payment gateway will fail.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `RET_STATUS` value.
- **AuthorizationCode** - `RET_AUTH_CODE` value.
- **TransactionID** - `RET_TRANS_ID` value.
- **AVSResult** - `RET_AVS_MSG` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

SecurePay

/hostedstore/storefront/gateways/cc_securepay.lasso

As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The merchant ID number. This corresponds to the `MERCH_ID` parameter passed to the gateway.
- **2** - The transaction type. This corresponds to the `Tr_Type` parameter passed to the gateway. Use `PREAUTH` to pre-authorize a transaction for later capture and processing. Use `SALE` to immediately process the transaction.
- **3** - The level of AVS verification. This corresponds to the `AVSREQ` parameter passed to the gateway. Use `0` for no AVS verification. Use `1` for credit card authorization and a full AVS verification (street address and zip code). Use `3` for credit card authorization and a zip code AVS verification.

Form URL and Parameters

The URL for the form must be set to the URL provided in the SecurePay documentation. The URL is currently: <https://www.securepay.com/secure15/index.cfm>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `MERCH_ID` and a `value` parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="MERCH_ID" value="[$HS_CCParam1]">`.
- A hidden form element with a `name` parameter with a value of `Tr_Type` and a `value` parameter with a value of `[$HS_CCParam2]`. The code should look like: `<input type="hidden" name="Tr_Type" value="[$HS_CCParam2]">`.
- A hidden form element with a `name` parameter with a value of `AVSREQ` and a `value` parameter with a value of `[$HS_CCParam3]`. The code should look like: `<input type="hidden" name="AVSREQ" value="[$HS_CCParam3]">`.
- A hidden form element with a `name` parameter with a value of `SUCCESS_URL` and a `value` parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="SUCCESS_URL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a `name` parameter with a value of `FAILURE_URL` and a `value` parameter with a value of `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`. The code should look like: `<input type="hidden" name="FAILURE_URL" value="[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso">`.
- A hidden form element with a `name` parameter with a value of `SECUREPAYINVOICE` and a `value` parameter with a value of `[$HSP_Processing->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="SECUREPAYINVOICE" value="[$HSP_Processing->Get:4->Get:1]">`.
- A hidden form element with a `name` parameter with a value of `SECUREPAYCUSTOMER` and a `value` parameter with a value of `[$HSP_Processing->Get:1->Get:13]`. The code should look like: `<input type="hidden" name="SECUREPAYCUSTOMER" value="[$HSP_Processing->Get:1->Get:13]">`.

The default form may pass more information than is minimally required by the payment gateway. If there is extraneous information that should not be passed, then remove the hidden form elements that are not needed. Just be sure that the minimum necessary information is still present or communication with the payment gateway will fail.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `Approv_Num` value.
- **AuthorizationCode** - `Approv_Num` value.
- **TransactionID** - `VoidRecNum` value.
- **AVSResult** - `AVS_Response` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

SkipJack

/hostedstore/storefront/gateways/cc_skipjack.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the SkipJack payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the SkipJack payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The account number. This corresponds to the `Serialnumber` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `szAuthorizationResponseCode` value.
- **AuthorizationCode** - `szAuthorizationResponseCode` value.
- **TransactionID** - `szOrderNumber` value.
- **AVSResult** - `szAVSResponseCode` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

Verisign® Payflow Link (SM)

/hostedstore/storefront/gateways/cc_verisign_payflowlink.lasso

Two steps are necessary for integrating with the Verisign payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The login name. This corresponds to the **LOGIN** parameter passed to the gateway.
- **2** - The partner name. This corresponds to the **PARTNER** parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the **TYPE** parameter passed to the gateway. Use **A** to pre-authorize a transaction for later capture and processing. Use **S** to immediately process the transaction.

Form URL and Parameters

The URL for the form must be: <https://payflowlink.verisign.com/payflowlink.cfm>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a **name** parameter with a value of **LOGIN** and a **value** parameter with a value of `[$HS_CCParam1]`. The code should look like: `<input type="hidden" name="LOGIN" value="[$HS_CCParam1]">`.
- A hidden form element with a **name** parameter with a value of **PARTNER** and a **value** parameter with a value of `[$HS_CCParam2]`. The code should look like: `<input type="hidden" name="PARTNER" value="[$HS_CCParam2]">`.
- A hidden form element with a **name** parameter with a value of **TYPE** and a **value** parameter with a value of `[$HS_CCParam3]`. The code should look like: `<input type="hidden" name="TYPE" value="[$HS_CCParam3]">`.
- A hidden form element with a **name** parameter with a value of **ECHODATA** and a **value** parameter with a value of `True`. The code should look like: `<input type="hidden" name="ECHODATA" value="True">`.
- A hidden form element with a **name** parameter with a value of **INVOICE** and a **value** parameter with a value of `[$HSP_Order->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="INVOICE" value="[$HSP_Order->Get:4->Get:1]">`.
- A hidden form element with a **name** parameter with a value of **CUSTID** and a **value** parameter with a value of `[$HSP_Order->Get:1->Get:13]`. The code should look like: `<input type="hidden" name="CUSTID" value="[$HSP_Order->Get:1->Get:13]">`.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Account Info > Payflow Link Info** tab. Again, consult the payment gateway documentation for the most current configuration information.

- The **Return URL Method** should be set to *Post*.
- The **Return URL** should be set to the secure return path to the store with `/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso` appended to the end. For example, for the domain `www.hostedstore.com`, the entry would look like:
`https://www.hostedstore.com/HostedStore.LassoApp?-`

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of *Y*. If declined, then the variable **HSP_Approved** will have a value of *N*. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - *RESULT* value.
- **AuthorizationCode** - *AUTHCODE* value.
- **TransactionID** - *PNFREF* value.
- **AVSResult** - *AVSDATA* value.
- **ApprovalMessage** - A *\r* delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Verisign® Payflow Pro (SM)

/hostedstore/storefront/gateways/cc_verisign_payflowpro.lasso

Important: The Verisign Payflow Pro payment gateway can only be used with Lasso Professional 6. Due to some conflicts between the tag used and the LJAPI implementation in LP 6.04, LP 6.04 cannot be used with Verisign Payflow Pro.

The necessary tag is provided with the HostedStore distribution. Instructions for installation and configuration can be found in the [Installation and Configuration](#) > [Required Software](#) section. Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 6 documentation for guidance on how to properly install Java.

Communication with the Verisign Payflow Pro payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Credit Card Gateway Parameters** in *Store Configuration*:

- **1** - The host address. This corresponds to the **HOSTADDRESS** parameter passed to the gateway.
- **2** - The host port. This corresponds to the **HOSTPORT** parameter passed to the gateway.
- **3** - The timeout value. This corresponds to the **TIMEOUT** parameter passed to the gateway.
- **4** - The transaction type. This corresponds to the **TRXTYPE** parameter passed to the gateway.
- **5** - The tender type. This corresponds to the **TENDER** parameter passed to the gateway.
- **6** - The partner. This corresponds to the **PARTNER** parameter passed to the gateway.
- **7** - The vendor. This corresponds to the **VENDOR** parameter passed to the gateway.
- **8** - The username. This corresponds to the **USER** parameter passed to the gateway.
- **9** - The password. This corresponds to the **PWD** parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of **Y**. If declined, then the variable **HSP_Approved** will have a value of **N**. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - **RESULT** value.
- **AuthorizationCode** - **AUTHCODE** value.
- **TransactionID** - **PNFREF** value.
- **AVSResult** - **AVSDATA** value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: **NAME: VALUE**

Check Processing

Documentation for shopping cart integration should have been supplied when the payment gateway account was established or should be available for download from the support area of the payment gateway's Web site. Please consult the payment gateway's documentation to ensure that all necessary information and wanted optional information is properly specified in the form used to post to the gateway.

Forms for several different payment gateways are provided with HostedStore. Please review the form and make sure that it conforms to the payment gateway's specification and the needs of the store. The minimum hidden form elements that HostedStore requires for proper functionality are detailed below for each form. Note that the minimum hidden form elements required by HostedStore may not directly correspond to the minimum hidden form elements required by the payment gateway.

Sections

- [Offline Processing](#)
- [ACH Direct](#)
- [Authorize.Net ADC Direct Response](#)
- [Authorize.Net Advanced Integration Method \(AIM\)](#)
- [LinkPoint Basic](#)
- [NETbilling \(DIRECT MODE\)](#)
- [Pay by Check](#)
- [Planet Payment™ ADC Direct Response](#)
- [Plug'nPay™](#)
- [Verisign® Payflow Link \(SM\)](#)
- [Verisign® Payflow Pro \(SM\)](#)

Offline Processing

/hostedstore/storefront/gateways/cc_offline.lasso

Offline processing does not send the payment information to a payment gateway. All of the information is stored internally by HostedStore and then the credit card must be processed manually later via the Web interface provided by the payment gateway.

The following items are required by HostedStore for the order processing form:

- The form action should be `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso` and should not be modified.
- A hidden form element with a `name` parameter with a value of `Offline` and a `value` parameter with a value of `[$HSP_Order->Get:10]`. The code should look like: `<input type="hidden" name="Offline" value="[$HSP_Order->Get:10]">`.
- Unlike many other HostedStore forms, the name of the submit button is irrelevant.

ACH Direct (DSI)

`/hostedstore/storefront/gateways/chk_achdirect.lasso`

Not implemented yet.

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the ACH Direct payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the ACH Direct payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The merchant ID. This corresponds to the `pg_merchant_id` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `pg_password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `pg_transaction_type` parameter passed to the gateway. Use `11` to pre-authorize a transaction for later capture and processing. Use `10` to immediately process the transaction.
- **4** - The degree of AVS checking to use. This corresponds to the `pg_avs_method` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `pg_response_code` value.
- **AuthorizationCode** - `pg_authorization_code` value.
- **TransactionID** - `pg_trace_number` value.
- **AVSResult** - `pg_avs_result` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

Authorize.Net ADC Direct Response

/hostedstore/storefront/gateways/chk_authorizenet_adc.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Authorize.Net payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Authorize.Net payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the AuthorizeNet account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Authorize.Net Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Automated Direct Connect (ADC) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of *Y*. If declined, then the variable **HSP_Approved** will have a value of *N*. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - *x_response_code* value.
- **AuthorizationCode** - *x_auth_code* value.
- **TransactionID** - *x_trans_id* value.
- **AVSResult** - *x_avs_code* value.
- **ApprovalMessage** - A *\r* delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Authorize.Net Advanced Integration Method (AIM)

/hostedstore/storefront/gateways/chk_authorizenet_aim.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Authorize.Net payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Authorize.Net payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the AuthorizeNet account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **7** - Transaction key. This corresponds to the `x-Tran_Key` parameter passed to the gateway. This value is set by Authorize.Net and can be obtained in the account administration area of the Authorize.Net Web site.
- **8** - Test mode. This corresponds to the `x_Test_Request` parameter passed to the gateway. Use `FALSE` if transactions are to be processed in live mode. Use `TRUE` if transactions are to be processed in test mode.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Authorize.Net Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Advanced Integration Method (AIM) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`.

The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (`/hostedstore/storefront/html/order_result.lasso`) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `x_response_code` value.
- **AuthorizationCode** - `x_auth_code` value.
- **TransactionID** - `x_trans_id` value.
- **AVSResult** - `x_avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

LinkPoint Basic

/hostedstore/storefront/gateways/chk_linkpoint_basic.lasso

Not implemented yet.

Two steps are necessary for integrating with the LinkPoint payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The store name. This corresponds to the `storename` parameter passed to the gateway.
- **2** - The payment mode. This corresponds to the `mode` parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `https://secure.linkpt.net/lpcentral/servlet/lppay`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `storename` and a `value` parameter with a value of `[$HS_CheckParam1]`. The code should look like: `<input type="hidden" name="storename" value="[$HS_CheckParam1]">`.
- A hidden form element with a `name` parameter with a value of `mode` and a `value` parameter with a value of `[$HS_CheckParam2]`. The code should look like: `<input type="hidden" name="mode" value="[$HS_CheckParam2]">`.
- A hidden form element with a `name` parameter with a value of `oid` and a `value` parameter with a value of `[$HSP_Order->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="oid" value="[$HSP_Order->Get:4->Get:1]">`.

NEED TO VERIFY RETURN URLS

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `approval_code` value.
- **AuthorizationCode** - `approval_code` value.
- **TransactionID** - `approval_code` value.
- **AVSResult** - `approval_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

NETbilling

/hostedstore/storefront/gateways/chk_netbilling.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the NETbilling payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the NETbilling payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The account number. This corresponds to the `GEN_ACCOUNT` parameter passed to the gateway.
- **2** - The Crypto-hash key associated with the account. This value can be obtained from the NETbilling account administration system.
- **3** - The transaction type. This corresponds to the `GEN_TRANS_TYPE` parameter passed to the gateway. Use `AUTH` to pre-authorize a transaction for later capture and processing. Use `SALE` to immediately process the transaction.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoicetable** that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `RET_STATUS` value.
- **AuthorizationCode** - `RET_AUTH_CODE` value.
- **TransactionID** - `RET_TRANS_ID` value.
- **AVSResult** - `RET_AVS_MSG` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

Pay by Check

/hostedstore/storefront/gateways/chk_paybycheck.lasso

Configuration

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the PayByCheck payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the PayByCheck payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The merchant ID. This corresponds to the **UserID** parameter passed to the gateway.
- **2** - The account password. This corresponds to the **Password** parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the **Type** parameter passed to the gateway. Use **CAO** to pre-authorize a transaction for later capture and processing. Use **DCP** to immediately process the transaction.
- **4** - The currency type. This corresponds to the **Csd** parameter passed to the gateway. Use **USD** for US dollars.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a **name** parameter with a value of *method* and a **value** parameter with a value of *Check*. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

NEED TO UPDATE ONCE ACCESS IS GRANTED AGAIN

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of *Y*. If declined, then the variable **HSP_Approved** will have a value of *N*. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - **RET_STATUS** value.
- **AuthorizationCode** - **RET_AUTH_CODE** value.
- **TransactionID** - **RET_TRANS_ID** value.
- **AVSResult** - **AVS Code** value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: **NAME: VALUE**

Planet Payment ADC Direct Response

/hostedstore/storefront/gateways/chk_planetpayment_adc.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Planet Payment payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Planet Payment payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The merchant ID. This corresponds to the `x_Login` parameter passed to the gateway.
- **2** - The account password. This corresponds to the `x_Password` parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the `x_Type` parameter passed to the gateway. Use `AUTH_ONLY` to pre-authorize a transaction for later capture and processing. Use `AUTH_CAPTURE` to immediately process the transaction.
- **4** - The MD5 secret associated with the account used to verify the `x_md5_hash` parameter passed back from the gateway. This value can be set in the Planet Payment account administration system.
- **5** - Gateway should email a receipt to the customer. This corresponds to the `x_Email_Customer` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.
- **6** - Gateway should email a receipt to the merchant. This corresponds to the `x_Email_Merchant` parameter passed to the gateway. Use `FALSE` if the gateway should not email a receipt.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Settings** menu of the account administration area of the Planet Payment Web site. Again, consult the payment gateway documentation for the most current configuration information.

Configuration for the **Automated Direct Connect (ADC) Settings**:

- The **Delimited Response** should be set to `YES`.
- The **Default Field Separator** should be set to `| (pipe)`.
- The **Default Field Encapsulation Character** should be set to `None`.
- The **Current Secret** should be set to the value set for parameter **4** in **Credit Card Gateway Parameters** above.
- The **Require Password for ALL Transactions** should be set to `YES`.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of *Y*. If declined, then the variable **HSP_Approved** will have a value of *N*. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - *x_response_code* value.
- **AuthorizationCode** - *x_auth_code* value.
- **TransactionID** - *x_trans_id* value.
- **AVSResult** - *x_avs_code* value.
- **ApprovalMessage** - A *\r* delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Plug'nPay™

/hostedstore/storefront/gateways/chk_plugnpay.lasso

Important: Due to limitations of Lasso Professional 5, a third party tag must be used to connect to the Plug'nPay payment gateway. The tag, **[Include_Secure_URL] LJAPI Tag**, can be purchased from The Data Web Center (www.datawebcenter.com). Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 5 documentation for guidance on how to properly install Java.

Communication with the Plug'nPay payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The account username. This corresponds to the `publisher-name` parameter passed to the gateway.
- **2** - The transaction type. This corresponds to the `auth-type` parameter passed to the gateway. Use `authonly` to pre-authorize a transaction for later capture and processing. Use `authpostauth` to immediately process the transaction.
- **3** - The server name and domain name to use to communicate with the payment gateway server. This value will be assigned by Plug'nPay.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

The following items are required by HostedStore for the order processing form:

- A hidden form element with a `name` parameter with a value of `method` and a `value` parameter with a value of `Check`. The code should look like: `<input type="hidden" name="method" value="Check">`.

Processing Results

If the transaction was successfully processed, then the variable `HSP_Approved` will have a value of `Y`. If declined, then the variable `HSP_Approved` will have a value of `N`. The value of `HSP_Approved` can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoicetable** that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - `auth-code` value.
- **AuthorizationCode** - `auth-code` value.
- **TransactionID** - `orderid` value.
- **AVSResult** - `avs_code` value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: `NAME: VALUE`

Verisign® Payflow Link (SM)

/hostedstore/storefront/gateways/chk_verisign_payflowlink.lasso

Two steps are necessary for integrating with the Verisign payment gateway: building the submitted form to pass the necessary information and configuring the account properly so that the proper return URL is specified, the gateway is active and ready to accept transactions, etc. As with all payment gateways, read the documentation provided by the gateway for details about the information necessary for submitted forms and for details on how to configure the gateway settings through their provided Web interface.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in the store configuration file:

- **1** - The login name. This corresponds to the **LOGIN** parameter passed to the gateway.
- **2** - The partner name. This corresponds to the **PARTNER** parameter passed to the gateway.
- **3** - The transaction type. This corresponds to the **TYPE** parameter passed to the gateway. Use **A** to pre-authorize a transaction for later capture and processing. Use **S** to immediately process the transaction.

Form URL and Parameters

The URL for the form must be: <https://payflowlink.verisign.com/payflowlink.cfm>

The following items are required by HostedStore for the order processing form:

- A hidden form element with a **name** parameter with a value of **LOGIN** and a **value** parameter with a value of `[$HS_CheckParam1]`. The code should look like: `<input type="hidden" name="LOGIN" value="[$HS_CheckParam1]">`.
- A hidden form element with a **name** parameter with a value of **PARTNER** and a **value** parameter with a value of `[$HS_CheckParam2]`. The code should look like: `<input type="hidden" name="PARTNER" value="[$HS_CheckParam2]">`.
- A hidden form element with a **name** parameter with a value of **TYPE** and a **value** parameter with a value of `[$HS_CheckParam3]`. The code should look like: `<input type="hidden" name="TYPE" value="[$HS_CheckParam3]">`.
- A hidden form element with a **name** parameter with a value of **ECHODATA** and a **value** parameter with a value of **True**. The code should look like: `<input type="hidden" name="ECHODATA" value="True">`.
- A hidden form element with a **name** parameter with a value of **INVOICE** and a **value** parameter with a value of `[$HSP_Order->Get:4->Get:1]`. The code should look like: `<input type="hidden" name="INVOICE" value="[$HSP_Order->Get:4->Get:1]">`.
- A hidden form element with a **name** parameter with a value of **CUSTID** and a **value** parameter with a value of `[$HSP_Order->Get:1->Get:13]`. The code should look like: `<input type="hidden" name="CUSTID" value="[$HSP_Order->Get:1->Get:13]">`.

The following items are required by HostedStore for the gateway configuration. As of the time of this writing, all of these settings are made on the **Account Info > Payflow Link Info** tab. Again, consult the payment gateway documentation for the most current configuration information.

- The **Return URL Method** should be set to **Post**.
- The **Return URL** should be set to the secure return path to the store with `/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso` appended to the end. For example, for the domain `www.hostedstore.com`, the entry would look like:
`https://www.hostedstore.com/HostedStore.LassoApp?-`

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of *Y*. If declined, then the variable **HSP_Approved** will have a value of *N*. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - *RESULT* value.
- **AuthorizationCode** - *AUTHCODE* value.
- **TransactionID** - *PNFREF* value.
- **AVSResult** - *AVSDATA* value.
- **ApprovalMessage** - A *\r* delimited list of all parameter name/value pairs in the form: *NAME: VALUE*

Verisign® Payflow Pro (SM)

/hostedstore/storefront/gateways/chk_verisign_payflowpro.lasso

Important: The Verisign Payflow Pro payment gateway can only be used with Lasso Professional 6. Due to some conflicts between the tag used and the LJAPI implementation in LP 6.04, LP 6.04 cannot be used with Verisign Payflow Pro.

The necessary tag is provided with the HostedStore distribution. Instructions for installation and configuration can be found in the [Installation and Configuration](#) > [Required Software](#) section. Note that this tag relies on a proper installation of Java. Please consult the Lasso Professional 6 documentation for guidance on how to properly install Java.

Communication with the Verisign Payflow Pro payment gateway occurs directly rather than via the submission of a form as with some other payment gateways.

Configuration

The following values should be assigned to the **Check Gateway Parameters** in *Store Configuration*:

- **1** - The host address. This corresponds to the **HOSTADDRESS** parameter passed to the gateway.
- **2** - The host port. This corresponds to the **HOSTPORT** parameter passed to the gateway.
- **3** - The timeout value. This corresponds to the **TIMEOUT** parameter passed to the gateway.
- **4** - The transaction type. This corresponds to the **TRXTYPE** parameter passed to the gateway.
- **5** - The tender type. This corresponds to the **TENDER** parameter passed to the gateway.
- **6** - The partner. This corresponds to the **PARTNER** parameter passed to the gateway.
- **7** - The vendor. This corresponds to the **VENDOR** parameter passed to the gateway.
- **8** - The username. This corresponds to the **USER** parameter passed to the gateway.
- **9** - The password. This corresponds to the **PWD** parameter passed to the gateway.

Form URL and Parameters

The URL for the form must be: `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=ordercomplete.lasso`

No parameters are necessary.

Processing Results

If the transaction was successfully processed, then the variable **HSP_Approved** will have a value of **Y**. If declined, then the variable **HSP_Approved** will have a value of **N**. The value of **HSP_Approved** can be checked on the order result content page (*/hostedstore/storefront/html/order_result.lasso*) so that the appropriate message can be displayed.

Listed below are the fields in the **Invoice** table that will be updated with the parameters passed back from the gateway:

- **ApprovalCode** - **RESULT** value.
- **AuthorizationCode** - **AUTHCODE** value.
- **TransactionID** - **PNFREF** value.
- **AVSResult** - **AVSDATA** value.
- **ApprovalMessage** - A `\r` delimited list of all parameter name/value pairs in the form: **NAME: VALUE**

PayPal (IPN version 1.2)

/hostedstore/storefront/gateways/cc_paypal.lasso

Not implemented yet.

Custom Payment Processing

Custom Forms

A custom form should be named *other_form.lasso* and placed in the */hostedstore/storefront/gateways/* directory. Like the built-in forms, the `HSP_Processing` variable is available. Details about the `HSP_Processing` variable can be found in the [Payment Processing > Built-In Forms](#) section.

The return hyperlink from the payment gateway to HostedStore should be set to the variable `HSP_PathFromGateway`. That variable will insert the path to the script that will process the reply from the gateway.

Custom Scripts

Custom scripts to process the payment gateway result should be named *other_script.lasso* and placed in the */hostedstore/storefront/gateways/* directory.

The custom script should handle processing the payment gateway result to determine if the transaction was approved. Any updating of the databases to reflect the transaction process should also be performed.

It is optional whether or not to perform the other functions necessary for completion of the order if the transaction was approved: transferring items from the cart to the order, transferring wishlist items, updating inventory, etc. If correctly performing all of the processing is overwhelming (and it is), then HostedStore can handle it if the appropriate variables are set with the expected values:

- `HSP_Approved` - Should have a value of `Y`.
- `HSP_OrderID` - Should be set to the value from the **OrderID** field in the **Invoice** table for the appropriate order.

If `HSP_Approved` has a value `Y` and `HSP_OrderID` has any value, then HostedStore will perform all of the processing for the approved order and return the order result content page (*/hostedstore/storefront/html/order_result.lasso*).

Modifying HostedStore

HostedStore uses a few custom tags to aid with consolidating code that is used in many different places. The [Modifying HostedStore](#) > [HostedStore Custom Tags](#) section details the functionality and syntax of each tag. Feel free to use the tags in any modifications made to the store templates, content pages, or email files.

HostedStore is built around a template system that allows for two files to completely control the look of a site. A few templates are provided for stores that need to get up and running immediately, but it is likely that most stores will use a store template that reflects the look of the rest of the Web site and/or better reflects the corporate image. The [Modifying HostedStore](#) > [Store Templates](#) > [Creating Storefront Templates](#) section provides instructions for easily creating a customized template for HostedStore.

Each of the content pages in HostedStore is an HTML file that contains the text of the page as well as any LDML code that creates the dynamic text of the page. The text portions of the content page can be easily modified if the default text is not suitable for the store. All of the LDML code is also documented in case additional or less content is wanted. Special care should be exercised when modifying the existing LDML code or adding new code. Improper syntax and/or content can affect the performance of the Web site. A list of consultants familiar with the HostedStore application is available on the HostedStore Web site if professional help is needed. Each content page is documented separately in the [Modifying HostedStore](#) > [Content Pages](#) section.

Important: Make a backup copy of all of the original files in the `/hostedstore/` directory before making any modifications. Writing code is often a trial and error process. A backup copy will ensure that a working copy can always be returned to for reference.

It is recommended that any changes made to the content pages are documented for future reference. Future features of HostedStore may incorporate changes to the default content pages. A log of past changes will aid in adding the new features to modified content pages or applying past modifications to the new default content pages.

Sections

- [HostedStore Custom Tags](#) - Custom tags provided with HostedStore.
- [Select Lists](#) - Lists of credit cards accepted, states/provinces, countries, etc that should be modified according to store policies.
- [Cookies](#) - Description of the contents for all cookies used by HostedStore.
- [Store Templates](#) - The template that provides the overall "look and feel" of the storefront.
- [Content Pages](#) - All of the pages that are inserted into the store template.
- [Email Messages](#) - All of the email messages sent from the store.

HostedStore Custom Tags

HostedStore uses a few custom tags to aid with consolidating code that is used in many different places. Most of the tags deal with formatting data for display. The tags are detailed below in case one or more may be helpful when making code modifications to the default content pages and email format files. Be warned that most of the tags do not use any internal error checking so be sure to follow the directions for each tag. Failure to do so may result in adverse results.

Sections

- [HS_ArrayRandomSort](#) - Randomizes the order of the elements in an array.
- [HS_FormatCurrency](#) - Formats monetary values according to preferences.
- [HS_FormatDate](#) - Formats dates according to preferences.
- [HS_FormatTime](#) - Formats times according to preferences.
- [HS_GetFavoriteProducts](#) - Gets an array of favorite products based on historical sales.
- [HS_GetFeaturedProducts](#) - Gets an array of favorite products.
- [HS_GetRecommendedProducts](#) - Gets an array of recommended products based on sales.
- [HS_IsRequired](#) - Outputs an asterisk if a variable does not contain a value.
- [HS_Percentage](#) - Divides two numbers and formats the result according to preferences.

HS_ArrayRandomSort

Outputs an array with its elements randomly sorted.

Syntax

`[HS_ArrayRandomSort : ARRAY]`, where *ARRAY* is replaced with a variable of type *ARRAY*.

HS_FormatCurrency

Outputs a string that is formatted according to currency formatting preferences specified in *Store Configuration* from a decimal input.

Syntax

```
[HS_FormatCurrency:DECIMAL VALUE]
```

Examples

```
[HS_FormatCurrency:'9.00']
```

HS_FormatDate

Outputs a string that is formatted according to date formatting preferences specified in *Store Configuration* from a string input in the form YYYY-MM-DD.

Syntax

```
[HS_FormatDate: 'YYYY-MM-DD VALUE' ]
```

Examples

```
[HS_FormatDate: '2002-02-02' ]
```

HS_FormatTime

Outputs a string that is formatted according to time formatting preferences specified in *Store Configuration* from a string input in the form HH:MM:SS.

Syntax

```
[HS_FormatTime:'HH:MM:SS VALUE']
```

Examples

```
[HS_FormatTime:'02:02:02']
```

HS_GetFavoriteProducts

Outputs an array of arrays that has information about top selling products. Products are determined to be top selling by sorting on the **QuantitySold** field in the **Product** table for those products that have a **QuantitySold** field value greater than zero, a **WebSell** field value of **Y**, and an **OutOfStockDisplay** field value of **Y** or an **OnHand** field value that is greater than the **WebSellQty** field value. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named **ID** to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.

Syntax

`[HS_GetFavoriteProducts:INTEGER VALUE]`, where *INTEGER VALUE* is replaced with the number of products to return.

The number of products returned will range from zero (if no products meet the search criteria) up to the value passed to the tag.

Optional Parameters

- **-M** (or **-Member**) - Adds an additional restriction so that only products with a **MemberOnlyItem** field value of **Y** will be returned if the passed value is **Y**. This parameter can be used to return top selling member only products. Adds an additional restriction so that only products with a **MemberOnlyItem** field value not equal to **Y** will be returned if the passed value is **N**. This parameter can be used to return top selling products that are not member only. If the parameter is not used, then all products will be returned regardless of the **MemberOnlyItem** field value.

Examples

`[HS_GetFavoriteProducts:10]` will return the top 10 selling products in the store (provided there are ten products that meet the restrictions detailed above).

`[HS_GetFavoriteProducts:10,-M='Y']` will return the top 10 selling member only products in the store (provided there are ten member only products that meet the restrictions detailed above).

`[HS_GetFavoriteProducts:10,-M='N']` will return the top 10 selling products in the store that are not member only products (provided there are ten products that are not member only that meet the restrictions detailed above).

HS_GetFeaturedProducts

Outputs an array of arrays that has information about featured products. Products are featured if the **Featured** field in the **Product** table has a value of **Y**, a **WebSell** field value of **Y**, and an **OutOfStockDisplay** field value of **Y** or an **OnHand** field value that is greater than the **WebSellQty** field value. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named **ID** to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.

Syntax

`[HS_GetFeaturedProducts:INTEGER VALUE]`, where `INTEGER VALUE` is replaced with the number of products to return.

The number of products returned will range from zero (if no products meet the search criteria) up to the value passed to the tag.

Optional Parameters

- **-M** (or **-Member**) - Adds an additional restriction so that only products with a **MemberOnlyItem** field value of **Y** will be returned if the passed value is **Y**. This parameter can be used to return featured member only products. Adds an additional restriction so that only products with a **MemberOnlyItem** field value not equal to **Y** will be returned if the passed value is **N**. This parameter can be used to return featured products that are not member only. If the parameter is not used, then all products will be returned regardless of the **Featured** field value.
- **-R** (or **-Random**) - Randomizes the order of products in the returned array. Otherwise the products are sorted by the **QuantitySold** field in the **Product** table so that top selling products have a lower index value in the array.

Examples

`[HS_GetFeaturedProducts:10]` will return the **10** ten selling featured products in the store (provided there are ten featured products that meet the restrictions detailed above).

`[HS_GetFeaturedProducts:10,-M='Y']` will return the top **10** selling featured member only products in the store (provided there are ten featured member only products that meet the restrictions detailed above).

`[HS_GetFeaturedProducts:10,-M='N']` will return the top **10** selling featured products in the store that are not member only (provided there are ten featured products that are not member only that meet the restrictions detailed above).

`[HS_GetFeaturedProducts:10,-R]` will return the top **10** selling featured products in the store (provided there are ten featured products that meet the restrictions detailed above) in a random order.

`[HS_GetFeaturedProducts:10,-M='Y',-R]` will return the top **10** selling featured member only products in the store (provided there are ten featured member only products that meet the restrictions detailed above) in a random order.

`[HS_GetFeaturedProducts:10,-M='N',-R]` will return the top 10 selling featured products in the store that are not member only (provided there are ten featured products that are not member only that meet the restrictions detailed above) in a random order.

HS_GetRecommendedProducts

Outputs an array of arrays that has information about recommended products. Recommended products are products that have been purchased in combination with the product(s) passed to the custom tag, a **WebSell** field value of **Y**, and an **OutOfStockDisplay** field value of **Y** or an **OnHand** field value that is greater than the **WebSellQty** field value. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named **ID** to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. The total quantity (sum of the **Quantity** field from the **Item** table) of the product ordered in combination with the product(s) passed to the custom tag.

Important: The results of this custom tag can take a long time to process if sev

Syntax

This custom tag can be used in two different ways:

1. `[HS_GetRecommendedProducts:STRING VALUE,INTEGER VALUE]`, where *STRING VALUE* is replaced with an order ID (**OrderID** field from the **Item** or **Invoice** table) and *INTEGER VALUE* is replaced with the number of products to return.
2. `[HS_GetRecommendedProducts:ARRAY VALUE,INTEGER VALUE]`, where *ARRAY VALUE* is replaced with an array of product IDs (**ProductID** field from the **Product** table) and *INTEGER VALUE* is replaced with the number of products to return.

The number of products returned will range from zero (if no products meet the search criteria) up to the value passed to the tag.

Optional Parameters

- **-M** (or **-Member**) - Adds an additional restriction so that only products with a **MemberOnlyItem** field value of **Y** will be returned if the passed value is **Y**. This parameter can be used to return featured member only or non-member only products. Adds an additional restriction so that only products with a **MemberOnlyItem** field value not equal to **Y** will be returned if the passed value is **N**. This parameter can be used to return featured member only or non-member only products. If the parameter is not used, then all products will be returned regardless of the **MemberOnlyItem** field value.
- **-O** (or **-Order**) - Specifies whether or not the top selling or lowest selling products will be returned. A value of **A** sorts the found set in ascending order on the sum of the **Quantity** field from the **Item** table so the lowest selling products will be returned first. A value of **D** sorts the found set in descending order on the sum of the **Quantity** field from the **Item** table so the top selling products will be returned first.
- **-R** (or **-Random**) - Randomizes the order of products in the returned array. Otherwise the products are sorted by the **QuantitySold** field in the **Product** table so that top selling products have a lower index value in the array.

Examples

[HS_GetRecommendedProducts:'D8AA260b1C00a3233FNU2AD138F',10] will return 10 products that were purchased with the products in order *D8AA260b1C00a3233FNU2AD138F* (provided there are ten products that meet the restrictions detailed above).

[HS_GetRecommendedProducts:(Array:'ProductID1','ProductID2'),10] will return 10 products that were purchased with the products identified by the *ProductID1* and *ProductID2* product IDs (provided there are ten products that meet the restrictions detailed above).

[HS_GetRecommendedProducts:'D8AA260b1C00a3233FNU2AD138F',10,-M='Y'] will return 10 member only products that were purchased with the products in order *D8AA260b1C00a3233FNU2AD138F* (provided there are ten member only products that meet the restrictions detailed above).

[HS_GetRecommendedProducts:'D8AA260b1C00a3233FNU2AD138F',10,-M='N'] will return 10 products that are not member only that were purchased with the products in order *D8AA260b1C00a3233FNU2AD138F* (provided there are ten products that are not member only that meet the restrictions detailed above).

[HS_GetRecommendedProducts:(Array:'ProductID1','ProductID2'),10,-M='Y'] will return 10 member only products that were purchased with the products identified by the *ProductID1* and *ProductID2* product IDs (provided there are ten member only products that meet the restrictions detailed above).

[HS_GetRecommendedProducts:(Array:'ProductID1','ProductID2'),10,-M='N'] will return 10 products that are not member only that were purchased with the products identified by the *ProductID1* and *ProductID2* product IDs (provided there are ten products that are not member only that meet the restrictions detailed above).

HS_IsRequired

Outputs `*` if any of the variables passed do not have a value. Use the `-EncodeNone` keyword with this tag to ensure that the HTML output from the tag is interpreted by the browser. Note that this tag is used for display purposes only. This tag will not do any processing to verify that the variable has a value.

Syntax

```
[HS_IsRequired:'VARIABLE 1 NAME',..., 'VARIABLE n NAME',-EncodeNone]
```

Examples

```
[HS_IsRequired:'Login','Password',-EncodeNone]
```

HS_Percentage

Outputs a string that is formatted according to percentage formatting preferences specified in *Store Configuration* from two decimal inputs.

Syntax

```
[HS_Percentage:DECIMAL VALUE 1,DECIMAL VALUE 2]
```

Examples

```
[HS_Percentage:33.3,100.0]
```

Select Lists

Static select lists are used to provide selections for billing countries, shipping countries, billing and shipping states/provinces/regions, credit cards accepted, and credit card expiration month and year. All of the select lists are located in the */hostedstore/includes/selectlists/* directory and can be modified to meet the policies of the store.

Sections

- [States](#) - States for billing and shipping addresses.
- [Countries](#) - Countries for billing addresses.
- [Shipping Countries](#) - Countries for shipping addresses.
- [Credit Cards](#) - Credit cards accepted.
- [Months](#) - Months for credit card expiration dates.
- [Years](#) - Years for credit card expiration dates.

States

/hostedstore/includes/selectlists/states.lasso

This select list is used for the state/province/region for billing and shipping addresses. This file is open for modification so that the appropriate values can be set for the geographical regions serviced by the store.

In case this file is somehow deleted or altered accidentally, details of how the file should be structured are provided below:

- The first `<option>` tag should have an empty `value` parameter. The text entry between the `<option>` and `</option>` should reflect what the customer should see. It is set to nothing in the default file.
- The rest of the `<option>` tags should have a `value` parameter for the state/province/region that is recognized by the postal service or shipping vendor. The text entry between the `<option>` and `</option>` tags should reflect what the customer should see. It is set to the appropriate full text state/province/region name in the default file. A variable named `7` is used to hold the selected value. In order to preserve selected values if the form is returned due to an error, the following line of code can be added to each `<option>` tag: `[If:$7=='STATE/PROVINCE/REGION VALUE'] selected[/If]>`

The code for the default file (note that only US States and Canadian Provinces are specified):

```
<option value="" [If:!$7] selected[/If]></option>
<option value="Outside US/Canada" [If:$7=='Outside US/Canada'] selected[/If]>
Outside US/Canada</option>
<option value=""></option>
<option value="AL" [If:$7=='AL'] selected[/If]>Alabama</option>
<option value="AK" [If:$7=='AK'] selected[/If]>Alaska</option>
<option value="AZ" [If:$7=='AZ'] selected[/If]>Arizona</option>
<option value="AR" [If:$7=='AR'] selected[/If]>Arkansas</option>
<option value="CA" [If:$7=='CA'] selected[/If]>California</option>
<option value="CO" [If:$7=='CO'] selected[/If]>Colorado</option>
<option value="CT" [If:$7=='CT'] selected[/If]>Connecticut</option>
<option value="DE" [If:$7=='DE'] selected[/If]>Delaware</option>
<option value="DC" [If:$7=='DC'] selected[/If]>District of Columbia</option>
<option value="FL" [If:$7=='FL'] selected[/If]>Florida</option>
<option value="GA" [If:$7=='GA'] selected[/If]>Georgia</option>
<option value="HA" [If:$7=='HI'] selected[/If]>Hawaii</option>
<option value="ID" [If:$7=='ID'] selected[/If]>Idaho</option>
<option value="IL" [If:$7=='IL'] selected[/If]>Illinois</option>
<option value="IN" [If:$7=='IN'] selected[/If]>Indiana</option>
<option value="IA" [If:$7=='IA'] selected[/If]>Iowa</option>
<option value="KS" [If:$7=='KS'] selected[/If]>Kansas</option>
<option value="KY" [If:$7=='KY'] selected[/If]>Kentucky</option>
<option value="LA" [If:$7=='LA'] selected[/If]>Louisiana</option>
<option value="ME" [If:$7=='ME'] selected[/If]>Maine</option>
<option value="MD" [If:$7=='MD'] selected[/If]>Maryland</option>
<option value="MA" [If:$7=='MA'] selected[/If]>Massachusetts</option>
<option value="MI" [If:$7=='MI'] selected[/If]>Michigan</option>
<option value="MN" [If:$7=='MN'] selected[/If]>Minnesota</option>
<option value="MS" [If:$7=='MS'] selected[/If]>Mississippi</option>
<option value="MO" [If:$7=='MO'] selected[/If]>Missouri</option>
<option value="MT" [If:$7=='MT'] selected[/If]>Montana</option>
<option value="NE" [If:$7=='NE'] selected[/If]>Nebraska</option>
<option value="NV" [If:$7=='NV'] selected[/If]>Nevada</option>
```

<option value="NH"[If:\$7=='NH'] selected[/If]>New Hampshire</option>
<option value="NJ"[If:\$7=='NJ'] selected[/If]>New Jersey</option>
<option value="NM"[If:\$7=='NM'] selected[/If]>New Mexico</option>
<option value="NY"[If:\$7=='NY'] selected[/If]>New York</option>
<option value="NC"[If:\$7=='NC'] selected[/If]>North Carolina</option>
<option value="ND"[If:\$7=='ND'] selected[/If]>North Dakota</option>
<option value="OH"[If:\$7=='OH'] selected[/If]>Ohio</option>
<option value="OK"[If:\$7=='OK'] selected[/If]>Oklahoma</option>
<option value="OR"[If:\$7=='OR'] selected[/If]>Oregon</option>
<option value="PA"[If:\$7=='PA'] selected[/If]>Pennsylvania</option>
<option value="RI"[If:\$7=='RI'] selected[/If]>Rhode Island</option>
<option value="SC"[If:\$7=='SC'] selected[/If]>South Carolina</option>
<option value="SD"[If:\$7=='SD'] selected[/If]>South Dakota</option>
<option value="TN"[If:\$7=='TN'] selected[/If]>Tennessee</option>
<option value="TX"[If:\$7=='TX'] selected[/If]>Texas</option>
<option value="UT"[If:\$7=='UT'] selected[/If]>Utah</option>
<option value="VT"[If:\$7=='VT'] selected[/If]>Vermont</option>
<option value="VA"[If:\$7=='VA'] selected[/If]>Virginia</option>
<option value="WA"[If:\$7=='WA'] selected[/If]>Washington</option>
<option value="WV"[If:\$7=='WV'] selected[/If]>West Virginia</option>
<option value="WI"[If:\$7=='WI'] selected[/If]>Wisconsin</option>
<option value="WY"[If:\$7=='WY'] selected[/If]>Wyoming</option>
<option value=""></option>
<option value="AB"[If:\$7=='AB'] selected[/If]>Alberta</option>
<option value="BC"[If:\$7=='BC'] selected[/If]>British Columbia</option>
<option value="MB"[If:\$7=='MB'] selected[/If]>Manitoba</option>
<option value="NB"[If:\$7=='NB'] selected[/If]>New Brunswick</option>
<option value="NF"[If:\$7=='NF'] selected[/If]>Newfoundland</option>
<option value="NS"[If:\$7=='NS'] selected[/If]>Nova Scotia</option>
<option value="NT"[If:\$7=='NT'] selected[/If]>Northwest Territories</option>
<option value="NU"[If:\$7=='NU'] selected[/If]>Nunavut</option>
<option value="ON"[If:\$7=='ON'] selected[/If]>Ontario</option>
<option value="PE"[If:\$7=='PE'] selected[/If]>Prince Edward Island</option>
<option value="QC"[If:\$7=='QC'] selected[/If]>Quebec</option>
<option value="SK"[If:\$7=='SK'] selected[/If]>Saskatchewan</option>
<option value="YT"[If:\$7=='YT'] selected[/If]>Yukon</option>

Countries

/hostedstore/includes/selectlists/countries.lasso

This select list is used for the country for billing and shipping addresses if separate supported shipping countries are not specified with the `HS_ShipCountry` preference variable in *Store Configuration*. This file is open for modification so that the appropriate values can be set for the geographical regions serviced by the store.

In case this file is somehow deleted or altered accidentally, details of how the file should be structured are provided below:

- The first `<option>` tag should have an empty `value` parameter. The text entry between the `<option>` and `</option>` should reflect what the customer should see. It is set to nothing in the default file.
- The rest of the `<option>` tags should have a `value` parameter for the country that is recognized by the postal service or shipping vendor. The text entry between the `<option>` and `</option>` tags should reflect what the customer should see. It is set to the appropriate full text country name in the default file. Some country names are so long that they are abbreviated in the default file because otherwise a wide select list would produce adverse formatting results when rendered in a browser.

The code for the default file:

```
<option value=""></option>
<option value="AL">Albania</option>
<option value="DZ">Algeria</option>
<option value="AS">American Samoa</option>
<option value="AD">Andorra</option>
<option value="AO">Angola</option>
<option value="AI">Anguilla</option>
<option value="AG">Antigua and Barbuda</option>
<option value="AR">Argentina</option>
<option value="AM">Armenia</option>
<option value="AW">Aruba</option>
<option value="AU">Australia</option>
<option value="AT">Austria</option>
<option value="AZ">Azerbaijan</option>
<option value="AP">Azores</option>
<option value="BS">Bahamas</option>
<option value="BH">Bahrain</option>
<option value="BD">Bangladesh</option>
<option value="BB">Barbados</option>
<option value="BY">Belarus</option>
<option value="BE">Belgium</option>
<option value="BZ">Belize</option>
<option value="BJ">Benin</option>
<option value="BM">Bermuda</option>
<option value="BT">Bhutan</option>
<option value="BO">Bolivia</option>
<option value="BL">Bonaire</option>
<option value="BA">Bosnia and Herzgovina</option>
<option value="BW">Botswana</option>
<option value="BV">Bouvet Island</option>
<option value="BR">Brazil</option>
<option value="IO">Brit. Indian Ocean Terr.</option>
```

<option value="BN">Brunei Darussalam</option>
<option value="BG">Bulgaria</option>
<option value="BF">Burkina Faso</option>
<option value="BI">Burundi</option>
<option value="KH">Cambodia</option>
<option value="CM">Cameroon</option>
<option value="CA">Canada</option>
<option value="IC">Canary Islands</option>
<option value="CV">Cape Verde</option>
<option value="KY">Cayman Islands</option>
<option value="CF">Central African Rep.</option>
<option value="TD">Chad</option>
<option value="CD">Channel Islands</option>
<option value="CL">Chile</option>
<option value="CN">China</option>
<option value="CX">Christmas Island</option>
<option value="CC">Cocos (Keeling) Islands</option>
<option value="CO">Colombia</option>
<option value="KM">Comoros</option>
<option value="ZP">Congo (Democratic Republic of)</option>
<option value="CK">Cook Islands</option>
<option value="CR">Costa Rica</option>
<option value="CI">Cote D'ivoire (Ivory Coast)</option>
<option value="HR">Croatia (Hrvatska)</option>
<option value="CB">Curacao</option>
<option value="CY">Cyprus</option>
<option value="CZ">Czech Republic</option>
<option value="DK">Denmark</option>
<option value="DJ">Djibouti</option>
<option value="DM">Dominica</option>
<option value="DO">Dominican Republic</option>
<option value="TP">East Timor</option>
<option value="EC">Ecuador</option>
<option value="EG">Egypt</option>
<option value="SV">El Salvador</option>
<option value="EN">England</option>
<option value="GQ">Equatorial Guinea</option>
<option value="ER">Eritrea</option>
<option value="EE">Estonia</option>
<option value="ET">Ethiopia</option>
<option value="FO">Faeroe Islands</option>
<option value="FK">Falkland Islands</option>
<option value="FJ">Fiji</option>
<option value="FI">Finland</option>
<option value="FR">France</option>
<option value="GF">French Guiana</option>
<option value="PF">French Polynesia</option>
<option value="TF">French Souther Terr.</option>
<option value="GA">Gabon</option>
<option value="GM">Gambia</option>
<option value="GE">Georgia</option>
<option value="DE">Germany</option>
<option value="GH">Ghana</option>
<option value="GI">Gibraltar</option>
<option value="GB">Great Britain (UK)</option>
<option value="GR">Greece</option>
<option value="GL">Greenland</option>

<option value="GD">Grenada</option>
<option value="GP">Guadeloupe</option>
<option value="GU">Guam</option>
<option value="GT">Guatemala</option>
<option value="GN">Guinea</option>
<option value="GW">Guinea-Bissau</option>
<option value="GY">Guyana</option>
<option value="HT">Haiti</option>
<option value="HM">Heard Is. & McDonald Is.</option>
<option value="HN">Honduras</option>
<option value="HK">Hong Kong</option>
<option value="HU">Hungary</option>
<option value="IS">Iceland</option>
<option value="IN">India</option>
<option value="ID">Indonesia</option>
<option value="IE">Ireland</option>
<option value="IL">Israel</option>
<option value="IT">Italy</option>
<option value="JM">Jamaica</option>
<option value="JP">Japan</option>
<option value="JO">Jordan</option>
<option value="KZ">Kazakhstan</option>
<option value="KE">Kenya</option>
<option value="KI">Kiribati</option>
<option value="KO">Kosrae</option>
<option value="KW">Kuwait</option>
<option value="KG">Kyrgyzstan</option>
<option value="LA">Laos</option>
<option value="LV">Latvia</option>
<option value="LB">Lebanon</option>
<option value="LS">Lesotho</option>
<option value="LR">Liberia</option>
<option value="LY">Libya</option>
<option value="LI">Liechtenstein</option>
<option value="LT">Lithuania</option>
<option value="LU">Luxembourg</option>
<option value="MO">Macau</option>
<option value="MK">Macedonia</option>
<option value="MG">Madagascar</option>
<option value="ME">Madeira</option>
<option value="MW">Malawi</option>
<option value="MY">Malaysia</option>
<option value="MV">Maldives</option>
<option value="ML">Mali</option>
<option value="MT">Malta</option>
<option value="MH">Marshall Islands</option>
<option value="MQ">Martinique</option>
<option value="MR">Mauritania</option>
<option value="MU">Mauritius</option>
<option value="YT">Mayotte</option>
<option value="MX">Mexico</option>
<option value="FM">Micronesia</option>
<option value="MD">Moldova</option>
<option value="MC">Monaco</option>
<option value="MN">Mongolia</option>
<option value="MS">Montserrat</option>
<option value="MA">Morocco</option>

<option value="MZ">Mozambique</option>
<option value="MM">Myanmar</option>
<option value="NA">Namibia</option>
<option value="NR">Nauru</option>
<option value="NP">Nepal</option>
<option value="NL">Netherlands</option>
<option value="AN">Netherlands Antilles</option>
<option value="NT">Neutral Zone</option>
<option value="NC">New Caledonia</option>
<option value="NZ">New Zealand (Aotearoa)</option>
<option value="NI">Nicaragua</option>
<option value="NE">Niger</option>
<option value="NG">Nigeria</option>
<option value="NU">Niue</option>
<option value="NF">Norfolk Island</option>
<option value="KP">North Korea</option>
<option value="NB">Northern Ireland</option>
<option value="MP">Northern Mariana Islands</option>
<option value="NO">Norway</option>
<option value="OM">Oman</option>
<option value="PK">Pakistan</option>
<option value="PW">Palau</option>
<option value="PA">Panama</option>
<option value="PG">Papua New Guinea</option>
<option value="PY">Paraguay</option>
<option value="PE">Peru</option>
<option value="PH">Philippines</option>
<option value="PN">Pitcairn</option>
<option value="PL">Poland</option>
<option value="PO">Ponape</option>
<option value="PT">Portugal</option>
<option value="PR">Puerto Rico</option>
<option value="QA">Qatar</option>
<option value="RE">Reunion</option>
<option value="RO">Romania</option>
<option value="RT">Rota</option>
<option value="RU">Russian Federation</option>
<option value="RW">Rwanda</option>
<option value="SS">Saba</option>
<option value="KN">Saint Kitts and Nevis</option>
<option value="LC">Saint Lucia</option>
<option value="VC">Saint Vincent and the Grenadines</option>
<option value="SP">Saipan</option>
<option value="WS">Samoa</option>
<option value="SM">San Marino</option>
<option value="ST">Sao Tome And Principe</option>
<option value="SA">Saudi Arabia</option>
<option value="SF">Scotland</option>
<option value="SN">Senegal</option>
<option value="SC">Seychelles</option>
<option value="SL">Sierra Leone</option>
<option value="SG">Singapore</option>
<option value="SK">Slovak Republic</option>
<option value="SI">Slovenia</option>
<option value="SB">Solomon Islands</option>
<option value="SO">Somalia</option>
<option value="ZA">South Africa</option>

<option value="GS">S. Georgia & S. Sand. Is.</option>
<option value="KR">South Korea</option>
<option value="ES">Spain</option>
<option value="LK">Sri Lanka</option>
<option value="NT">St. Barthelemy</option>
<option value="SW">St. Christopher</option>
<option value="VI">St. Croix</option>
<option value="EU">St. Eustatius</option>
<option value="SH">St. Helena</option>
<option value="UV">St. John</option>
<option value="KN">St. Kitts and Nevis</option>
<option value="LC">St. Lucia</option>
<option value="MB">St. Maarten</option>
<option value="TB">St. Martin</option>
<option value="PM">St. Pierre & Miquelon</option>
<option value="VL">St. Thomas</option>
<option value="VC">St. Vincent/Grenadine</option>
<option value="SD">Sudan</option>
<option value="SR">Suriname</option>
<option value="SJ">Svalbard & Jan Mayen Is.</option>
<option value="SZ">Swaziland</option>
<option value="SE">Sweden</option>
<option value="CH">Switzerland</option>
<option value="SY">Syria</option>
<option value="TA">Tahiti</option>
<option value="TW">Taiwan</option>
<option value="TJ">Tajikistan</option>
<option value="TZ">Tanzania</option>
<option value="TH">Thailand</option>
<option value="TI">Tinian</option>
<option value="TG">Togo</option>
<option value="TK">Tokelau</option>
<option value="TO">Tonga</option>
<option value="TL">Tortola</option>
<option value="TT">Trinidad & Tobago</option>
<option value="TU">Truk</option>
<option value="TN">Tunisia</option>
<option value="TR">Turkey</option>
<option value="TM">Turkmenistan</option>
<option value="TC">Turks & Caicos Is.</option>
<option value="TV">Tuvalu</option>
<option value="UG">Uganda</option>
<option value="UA">Ukraine</option>
<option value="UI">Union Island</option>
<option value="AE">United Arab Emirates</option>
<option value="US">United States of America</option>
<option value="UY">Uruguay</option>
<option value="UM">US Minor Outlying Islands</option>
<option value="SU">USSR (former)</option>
<option value="UZ">Uzbekistan</option>
<option value="VU">Vanuatu</option>
<option value="VA">Vatican City State (Holy See)</option>
<option value="VE">Venezuela</option>
<option value="VN">Vietnam</option>
<option value="VR">Virgin Gorda</option>
<option value="VG">Virgin Islands (British)</option>
<option value="VI">Virgin Islands (U.S.)</option>

<option value="WL">Wales</option>
<option value="WF">Wallis And Futuna Is.</option>
<option value="WS">Western Samoa</option>
<option value="YA">Yap</option>
<option value="YE">Yemen</option>
<option value="YU">Yugoslavia</option>
<option value="ZR">Zaire</option>
<option value="ZM">Zambia</option>
<option value="ZW">Zimbabwe</option>

Shipping Countries

/hostedstore/includes/selectlists/countries_shipping.lasso

This select list is used for the country for shipping addresses if separate supported shipping countries are specified with the `HS_ShipCountry` preference variable in *Store Configuration*. This file is open for modification so that the appropriate values can be set for the geographical regions serviced by the store.

The code for the default file:

```
<option value=""></option>
<option value="AL">Albania</option>
<option value="DZ">Algeria</option>
<option value="AS">American Samoa</option>
<option value="AD">Andorra</option>
<option value="AO">Angola</option>
<option value="AI">Anguilla</option>
<option value="AG">Antigua and Barbuda</option>
<option value="AR">Argentina</option>
<option value="AM">Armenia</option>
<option value="AW">Aruba</option>
<option value="AU">Australia</option>
<option value="AT">Austria</option>
<option value="AZ">Azerbaijan</option>
<option value="AP">Azores</option>
<option value="BS">Bahamas</option>
<option value="BH">Bahrain</option>
<option value="BD">Bangladesh</option>
<option value="BB">Barbados</option>
<option value="BY">Belarus</option>
<option value="BE">Belgium</option>
<option value="BZ">Belize</option>
<option value="BJ">Benin</option>
<option value="BM">Bermuda</option>
<option value="BT">Bhutan</option>
<option value="BO">Bolivia</option>
<option value="BL">Bonaire</option>
<option value="BA">Bosnia and Herzgovina</option>
<option value="BW">Botswana</option>
<option value="BV">Bouvet Island</option>
<option value="BR">Brazil</option>
<option value="IO">Brit. Indian Ocean Terr.</option>
<option value="BN">Brunei Darussalam</option>
<option value="BG">Bulgaria</option>
<option value="BF">Burkina Faso</option>
<option value="BI">Burundi</option>
<option value="KH">Cambodia</option>
<option value="CM">Cameroon</option>
<option value="CA">Canada</option>
<option value="IC">Canary Islands</option>
<option value="CV">Cape Verde</option>
<option value="KY">Cayman Islands</option>
<option value="CF">Central African Rep.</option>
<option value="TD">Chad</option>
<option value="CD">Channel Islands</option>
```

<option value="CL">Chile</option>
<option value="CN">China</option>
<option value="CX">Christmas Island</option>
<option value="CC">Cocos (Keeling) Islands</option>
<option value="CO">Colombia</option>
<option value="KM">Comoros</option>
<option value="ZP">Congo (Democratic Republic of)</option>
<option value="CK">Cook Islands</option>
<option value="CR">Costa Rica</option>
<option value="CI">Cote D'ivoire (Ivory Coast)</option>
<option value="HR">Croatia (Hrvatska)</option>
<option value="CB">Curacao</option>
<option value="CY">Cyprus</option>
<option value="CZ">Czech Republic</option>
<option value="DK">Denmark</option>
<option value="DJ">Djibouti</option>
<option value="DM">Dominica</option>
<option value="DO">Dominican Republic</option>
<option value="TP">East Timor</option>
<option value="EC">Ecuador</option>
<option value="EG">Egypt</option>
<option value="SV">El Salvador</option>
<option value="EN">England</option>
<option value="GQ">Equatorial Guinea</option>
<option value="ER">Eritrea</option>
<option value="EE">Estonia</option>
<option value="ET">Ethiopia</option>
<option value="FO">Faeroe Islands</option>
<option value="FK">Falkland Islands</option>
<option value="FJ">Fiji</option>
<option value="FI">Finland</option>
<option value="FR">France</option>
<option value="GF">French Guiana</option>
<option value="PF">French Polynesia</option>
<option value="TF">French Souther Terr.</option>
<option value="GA">Gabon</option>
<option value="GM">Gambia</option>
<option value="GE">Georgia</option>
<option value="DE">Germany</option>
<option value="GH">Ghana</option>
<option value="GI">Gibraltar</option>
<option value="GB">Great Britain (UK)</option>
<option value="GR">Greece</option>
<option value="GL">Greenland</option>
<option value="GD">Grenada</option>
<option value="GP">Guadeloupe</option>
<option value="GU">Guam</option>
<option value="GT">Guatemala</option>
<option value="GN">Guinea</option>
<option value="GW">Guinea-Bissau</option>
<option value="GY">Guyana</option>
<option value="HT">Haiti</option>
<option value="HM">Heard Is. & McDonald Is.</option>
<option value="HN">Honduras</option>
<option value="HK">Hong Kong</option>
<option value="HU">Hungary</option>
<option value="IS">Iceland</option>

<option value="IN">India</option>
<option value="ID">Indonesia</option>
<option value="IE">Ireland</option>
<option value="IL">Israel</option>
<option value="IT">Italy</option>
<option value="JM">Jamaica</option>
<option value="JP">Japan</option>
<option value="JO">Jordan</option>
<option value="KZ">Kazakhstan</option>
<option value="KE">Kenya</option>
<option value="KI">Kiribati</option>
<option value="KO">Kosrae</option>
<option value="KW">Kuwait</option>
<option value="KG">Kyrgyzstan</option>
<option value="LA">Laos</option>
<option value="LV">Latvia</option>
<option value="LB">Lebanon</option>
<option value="LS">Lesotho</option>
<option value="LR">Liberia</option>
<option value="LY">Libya</option>
<option value="LI">Liechtenstein</option>
<option value="LT">Lithuania</option>
<option value="LU">Luxembourg</option>
<option value="MO">Macau</option>
<option value="MK">Macedonia</option>
<option value="MG">Madagascar</option>
<option value="ME">Madeira</option>
<option value="MW">Malawi</option>
<option value="MY">Malaysia</option>
<option value="MV">Maldives</option>
<option value="ML">Mali</option>
<option value="MT">Malta</option>
<option value="MH">Marshall Islands</option>
<option value="MQ">Martinique</option>
<option value="MR">Mauritania</option>
<option value="MU">Mauritius</option>
<option value="YT">Mayotte</option>
<option value="MX">Mexico</option>
<option value="FM">Micronesia</option>
<option value="MD">Moldova</option>
<option value="MC">Monaco</option>
<option value="MN">Mongolia</option>
<option value="MS">Montserrat</option>
<option value="MA">Morocco</option>
<option value="MZ">Mozambique</option>
<option value="MM">Myanmar</option>
<option value="NA">Namibia</option>
<option value="NR">Nauru</option>
<option value="NP">Nepal</option>
<option value="NL">Netherlands</option>
<option value="AN">Netherlands Antilles</option>
<option value="NT">Neutral Zone</option>
<option value="NC">New Caledonia</option>
<option value="NZ">New Zealand (Aotearoa)</option>
<option value="NI">Nicaragua</option>
<option value="NE">Niger</option>
<option value="NG">Nigeria</option>

<option value="NU">Niue</option>
<option value="NF">Norfolk Island</option>
<option value="KP">North Korea</option>
<option value="NB">Northern Ireland</option>
<option value="MP">Northern Mariana Islands</option>
<option value="NO">Norway</option>
<option value="OM">Oman</option>
<option value="PK">Pakistan</option>
<option value="PW">Palau</option>
<option value="PA">Panama</option>
<option value="PG">Papua New Guinea</option>
<option value="PY">Paraguay</option>
<option value="PE">Peru</option>
<option value="PH">Philippines</option>
<option value="PN">Pitcairn</option>
<option value="PL">Poland</option>
<option value="PO">Ponape</option>
<option value="PT">Portugal</option>
<option value="PR">Puerto Rico</option>
<option value="QA">Qatar</option>
<option value="RE">Reunion</option>
<option value="RO">Romania</option>
<option value="RT">Rota</option>
<option value="RU">Russian Federation</option>
<option value="RW">Rwanda</option>
<option value="SS">Saba</option>
<option value="KN">Saint Kitts and Nevis</option>
<option value="LC">Saint Lucia</option>
<option value="VC">Saint Vincent and the Grenadines</option>
<option value="SP">Saipan</option>
<option value="WS">Samoa</option>
<option value="SM">San Marino</option>
<option value="ST">Sao Tome And Principe</option>
<option value="SA">Saudi Arabia</option>
<option value="SF">Scotland</option>
<option value="SN">Senegal</option>
<option value="SC">Seychelles</option>
<option value="SL">Sierra Leone</option>
<option value="SG">Singapore</option>
<option value="SK">Slovak Republic</option>
<option value="SI">Slovenia</option>
<option value="SB">Solomon Islands</option>
<option value="SO">Somalia</option>
<option value="ZA">South Africa</option>
<option value="GS">S. Georgia & S. Sand. Is.</option>
<option value="KR">South Korea</option>
<option value="ES">Spain</option>
<option value="LK">Sri Lanka</option>
<option value="NT">St. Barthelemy</option>
<option value="SW">St. Christopher</option>
<option value="VI">St. Croix</option>
<option value="EU">St. Eustatius</option>
<option value="SH">St. Helena</option>
<option value="UV">St. John</option>
<option value="KN">St. Kitts and Nevis</option>
<option value="LC">St. Lucia</option>

<option value="MB">St. Maarten</option>
<option value="TB">St. Martin</option>
<option value="PM">St. Pierre & Miquelon</option>
<option value="VL">St. Thomas</option>
<option value="VC">St. Vincent/Grenadine</option>
<option value="SD">Sudan</option>
<option value="SR">Suriname</option>
<option value="SJ">Svalbard & Jan Mayen Is.</option>
<option value="SZ">Swaziland</option>
<option value="SE">Sweden</option>
<option value="CH">Switzerland</option>
<option value="SY">Syria</option>
<option value="TA">Tahiti</option>
<option value="TW">Taiwan</option>
<option value="TJ">Tajikistan</option>
<option value="TZ">Tanzania</option>
<option value="TH">Thailand</option>
<option value="TI">Tinian</option>
<option value="TG">Togo</option>
<option value="TK">Tokelau</option>
<option value="TO">Tonga</option>
<option value="TL">Tortola</option>
<option value="TT">Trinidad & Tobago</option>
<option value="TU">Truk</option>
<option value="TN">Tunisia</option>
<option value="TR">Turkey</option>
<option value="TM">Turkmenistan</option>
<option value="TC">Turks & Caicos Is.</option>
<option value="TV">Tuvalu</option>
<option value="UG">Uganda</option>
<option value="UA">Ukraine</option>
<option value="UI">Union Island</option>
<option value="AE">United Arab Emirates</option>
<option value="US">United States of America</option>
<option value="UY">Uruguay</option>
<option value="UM">US Minor Outlying Islands</option>
<option value="SU">USSR (former)</option>
<option value="UZ">Uzbekistan</option>
<option value="VU">Vanuatu</option>
<option value="VA">Vatican City State (Holy See)</option>
<option value="VE">Venezuela</option>
<option value="VN">Vietnam</option>
<option value="VR">Virgin Gorda</option>
<option value="VG">Virgin Islands (British)</option>
<option value="VI">Virgin Islands (U.S.)</option>
<option value="WL">Wales</option>
<option value="WF">Wallis And Futuna Is.</option>
<option value="WS">Western Samoa</option>
<option value="YA">Yap</option>
<option value="YE">Yemen</option>
<option value="YU">Yugoslavia</option>
<option value="ZR">Zaire</option>
<option value="ZM">Zambia</option>
<option value="ZW">Zimbabwe</option>

Credit Cards

/hostedstore/includes/selectlists/creditcards.lasso

This select list is used for the credit cards accepted in the store. This file is open for modification so that the proper credit cards can be entered.

In case this file is somehow deleted or altered accidentally, details of how the file should be structured are provided below:

- The first `<option>` tag should have an empty `value` parameter. The text entry between the `<option>` and `</option>` should reflect what the customer should see. It is set to nothing in the default file.
- The rest of the `<option>` tags should have a `value` parameter of the credit card name in a format expected by the payment gateway. The text entry between the `<option>` and `</option>` tags should reflect what the customer should see. It is set to the name of the credit card in the default file (note that this value does not necessarily match the value expected by the payment gateway). A variable named `14` is used to hold the selected value. In order to preserve selected values if the form is returned due to an error, the following line of code can be added to each `<option>` tag:
`[If:$14=='CREDIT CARD GATEWAY VALUE'] selected[/If]`

The code for the default file:

```
<option value="" [If:!$14] selected[/If]></option>
<option value="AMEX" [If:$14=='AMEX'] selected[/If]>
American Express</option>
<option value="Discover" [If:$14=='Discover'] selected[/If]>
Discover Card</option>
<option value="MasterCard" [If:$14=='MasterCard'] selected[/If]>
MasterCard</option>
<option value="Visa" [If:$14=='Visa'] selected[/If]>Visa</option>
```

Months

/hostedstore/includes/selectlists/months.lasso

This select list is used for the month portion of credit card expiration dates. This file is open for modification so that the value of the select list displayed to the customer can be altered. Other than that, this file should not be modified.

In case this file is somehow deleted or altered accidentally, details of how the file should be structured are provided below:

- The first `<option>` tag should have an empty `value` parameter. The text entry between the `<option>` and `</option>` should reflect what the customer should see. It is set to *Month* in the default file.
- The rest of the `<option>` tags should have a `value` parameter of the month in two digit (MM) format. The text entry between the `<option>` and `</option>` tags should reflect what the customer should see. It is set to the appropriate two digit month in the default file. A variable named `17` is used to hold the selected value. In order to preserve selected values if the form is returned due to an error, the following line of code can be added to each `<option>` tag:
`[If:$17=='MM VALUE'] selected[/If]`

The code for the default file:

```
<option value="" [If:!$17] selected[/If]>Month</option>
<option value="01" [If:$17=='01'] selected[/If]>01</option>
<option value="02" [If:$17=='02'] selected[/If]>02</option>
<option value="03" [If:$17=='03'] selected[/If]>03</option>
<option value="04" [If:$17=='04'] selected[/If]>04</option>
<option value="05" [If:$17=='05'] selected[/If]>05</option>
<option value="06" [If:$17=='06'] selected[/If]>06</option>
<option value="07" [If:$17=='07'] selected[/If]>07</option>
<option value="08" [If:$17=='08'] selected[/If]>08</option>
<option value="09" [If:$17=='09'] selected[/If]>09</option>
<option value="10" [If:$17=='10'] selected[/If]>10</option>
<option value="11" [If:$17=='11'] selected[/If]>11</option>
<option value="12" [If:$17=='12'] selected[/If]>12</option>
```

Years

/hostedstore/includes/selectlists/years.lasso

This select list is used for the year portion of credit card expiration dates. This file is open for modification so that new years can be added and old years removed in order to keep the year select list current. Other than that, this file should not be modified.

In case this file is somehow deleted or altered accidentally, details of how the file should be structured are provided below:

- The first `<option>` tag should have an empty `value` parameter. The text entry between the `<option>` and `</option>` should reflect what the customer should see. It is set to `Year` in the default file.
- The rest of the `<option>` tags should have a `value` parameter of the year in four digit (YYYY) format. The text entry between the `<option>` and `</option>` tags should reflect what the customer should see. It is set to the appropriate four digit year in the default file. A variable named `18` is used to hold the selected value. In order to preserve selected values if the form is returned due to an error, the following line of code can be added to each `<option>` tag:
`[If:$18=='YYYY VALUE'] selected[/If]`

The code for the default file:

```
<option value="" [If:!$18] selected[/If]>Year</option>
<option value="2002" [If:$18=='2002'] selected[/If]>2002</option>
<option value="2003" [If:$18=='2003'] selected[/If]>2003</option>
<option value="2004" [If:$18=='2004'] selected[/If]>2004</option>
<option value="2005" [If:$18=='2005'] selected[/If]>2005</option>
<option value="2006" [If:$18=='2006'] selected[/If]>2006</option>
<option value="2007" [If:$18=='2007'] selected[/If]>2007</option>
<option value="2008" [If:$18=='2008'] selected[/If]>2008</option>
<option value="2009" [If:$18=='2009'] selected[/If]>2009</option>
<option value="2010" [If:$18=='2010'] selected[/If]>2010</option>
<option value="2011" [If:$18=='2011'] selected[/If]>2011</option>
<option value="2012" [If:$18=='2012'] selected[/If]>2012</option>
```

Cookies

Changes to the content pages may involve using values that are stored in cookies. Each cookie is encrypted with the `HS_Encrypt` preference variable set in *Store Configuration* used as the key. To decrypt a cookie use the `Decrypt_Blowfish` LDML tag with the `HS_Encrypt` preference variable as the value for the `-Seed` parameter. To encrypt a value to reset a cookie use the `Encrypt_Blowfish` LDML tag with the `HS_Encrypt` preference variable as the value for the `-Seed` parameter. Example uses of each of these tags:

- Decrypting a value: `[Decrypt_Blowfish:VALUE,-Seed=$HS_Encrypt]`
- Encrypting a value: `[Encrypt_Blowfish:VALUE,-Seed=$HS_Encrypt]`

Cookies that hold multiple values will have each values separated by two adjacent pipe characters (`||`). Each cookie is detailed below:

- `HS_Account` - Identifies a customer who has successfully logged into *Customer Account Administration* or logged in during the checkout process. Two values are contained in the cookie. The position of each value is numbered below along with an explanation of the value:
 1. The key value for the customer record (**ID** field from the **Customer** table).
 2. The customer ID (**ID** field from the **Customer** table).
 3. The customer price group ID (**PriceGroupID** field from the **Customer** table).
 4. The price group type (**Type** field from the **PriceGroup** table).
 5. The price group value (**Value** field from the **PriceGroup** table).
 6. `Y` if the **ApprovedForPO** field for the customer record has a value of `Y`.
- `HS_Affiliate` - Identifies an affiliate who has successfully logged into *Affiliate Account Administration*. The cookie has one value, the affiliate ID (**ID** field from the **Affiliate** table).
- `HS_AID` - Tracks the affiliate ID that was passed in a referral hyperlink. The cookie has one value, the affiliate ID (**ID** field from the **Affiliate** table).
- `HS_Cart` - Tracks the order ID for the order and the values used to display the static shopping cart for the default store templates. Four values are contained in the cookie. The position of each value is numbered below along with an explanation of the value:
 1. The order ID for cart items and the order if an order has been created (**OrderID** field from the **Item** and **Invoice** tables, respectively).
 2. The number of items in the cart.
 3. The subtotal of items in the cart formatted according to currency formatting preferences set in *Store Configuration*.
 4. The key value for the order record if an order has been created (**ID** field from the **Invoice** table).
- `HS_Member` - Has a value of `Y` if a customer that has a current membership has logged into *Customer Account Administration* or logged in during the checkout process.
- `HS_PID` - This cookie is used to track the promotion code that was passed in a hyperlink. The cookie has one value, the promotion code (**PromotionCode** field from the **Promotion** table).

Store Templates

Store template files are used for the overall "look and feel" of the store. By using a template approach, all elements that are common to every store page are located in one set of files so that one simple change can be made and globally applied to all store pages. HostedStore ships with some template files to provide examples of how templates are used. While these templates can be used for a store, it is expected that most stores will use their own templates to match their company image.

Additional storefront and Store Administration templates from HostedStore and third parties may be available. Check the HostedStore Web site for more information.

Sections

- [Modifying Supplied Storefront Templates](#) - Instructions for modifying the templates that are provided with HostedStore.
- [Creating Storefront Templates](#) - Instructions for creating a new storefront template.
- [Creating Store Administration Templates](#) - Instructions for creating a new *Store Administration* template.

Modifying Supplied Storefront Templates

Any of the supplied storefront templates may be modified as needed. Details about how storefront template files are structured can be found in the [Modifying HostedStore > Store Templates > Creating Storefront Templates](#) section.

If a supplied storefront template is used, the minimum modification that should be made is to modify the *logo.gif* file in the *images/* directory for the template to display the store/company logo.

Important: The supplied storefront templates are provided as examples of how storefront templates are structured. The templates may or may not have properly structured and valid HTML. While it is possible to use the supplied storefront templates for a store, it is recommended that the HTML is reviewed if valid HTML is wanted.

All of the HostedStore default content pages assume a style sheet is used to change the appearance of fonts. The details of the classes are provided below. If a style sheet is not used, then HTML `` will need to be added to all of the default content pages to change the appearance otherwise all of the content will appear using the default font setting in the site visitor's browser.

- `title` - Used for the title displayed at the top of every content page.
- `subtitle` - Used to title subsections of content page.
- `lg` - Used when a larger font size is needed. Mainly used to title sections on a page.
- `med` - Used for most of the text throughout the store.
- `sm` - Used when a smaller font size is needed. Mainly used to indicate wishlist purchases and product attributes.
- `header` - Used for the header labels in tables.
- `label` - Used for field name labels in forms.
- `asterisk` - Used for the asterisk that is used to indicate required fields.
- `warning` - Used to display results messages or errors.
- `select` - Used for HTML `<select>` tags.
- `input` - Used for HTML `<input>` tags.
- `textarea` - Used for HTML `<textarea>` tags.
- `invoice_title` - Used to title the invoices generated in *Order Administration*.
- `invoice_lg` - Used for large sized text on the invoices generated in *Order Administration*.
- `invoice_med` - Used for medium sized text on the invoices generated in *Order Administration*.
- `invoice_sm` - Used for small sized text on the invoices generated in *Order Administration*.

Creating Storefront Templates

Creating a new storefront template is a straightforward process, but care should be taken to properly follow each of the steps. Feel free to look at any of the provided templates in the `/hostedstore/templates/` directory to see examples.

1. Create and design the page HTML with a text editor or GUI editor.
2. If a dynamic page title and dynamic meta tags are wanted, then substitute the following values in the appropriate place in the document head of the HTML page:
 - `<title>[$HS_PageTitle]</title>`
 - `<meta name="keywords" content="[$HS_MetaKeywords]">`
 - `<meta name="description" content="[$HS_MetaDescription]">`
3. All image `src` parameters must be replaced with the following path in front of the image name:
 - `[$HS_FilePathToStore]/templates/[$HS_TemplateName]/images/`
4. If the navigation for category, subcategory, etc selection is to be based off the `/hostedstore/includes/navigation/browseproducts.lasso` file, then place the following code where the navigation is to appear:
 - `[Include:($HS_IncludeFilePathToStore+ '/includes/navigation/browseproducts.lasso')]`
5. If footer navigation for areas of the store is to be based off the `/hostedstore/includes/navigation/navfooter.lasso` file, then place the following code where the footer is to appear:
 - `[Include:($HS_IncludeFilePathToStore+ '/includes/navigation/navfooter.lasso')]`
6. To display a static shopping cart, use `[$HSP_CartItemCount]` to display the number of items in the shopping cart and `[$HSP_CartSubTotal]` to display the shopping cart subtotal.
7. Links to the shopping cart should use the following the hyperlink code:
 - ``
8. Links to the checkout pages should use the following the hyperlink code:
 - ``
9. Determine where the content pages will be inserted into the template and take the upper half of the HTML document and save it with the name `top.lasso`. Save the lower half of the template as `bottom.lasso`.
10. Create a new directory with the name of the template and place it in the `/hostedstore/templates/` directory. The `top.lasso` and `bottom.lasso` files should go into the new template directory.
11. Create a new directory named `images` and place it in the new template directory. All images for the template should be placed in the new `images` directory.
12. Shortcuts have been built into the search script that allow for the placement of a quick search form that searches for a keyword across multiple fields in the **Product** table. Details of requirements for this feature are provided below:
 - The form action should be `HostedStore.LassoApp?-ResponseLassoApp=search.lasso` and should not be modified. Unlike many other HostedStore forms, the name of the submit button is irrelevant.
 - A form parameter named `Search` is used to capture the keyword to use in the search.
 - A form parameter named `SearchFields` is used to specify a comma delimited list of the fields to include in the search. The fields will be searched using a `contains` search operator. A list of the fields and the corresponding number is provided below:
 1. The **ProductName** field in the **Product** table.
 2. The **Category** field in the **Product** table.
 3. The **SubCategory** field in the **Product** table.
 4. The **SubCategory2** field in the **Product** table.
 5. The **SubCategory3** field in the **Product** table.
 6. The **Department** field in the **Product** table.
 7. The **Manufacturer** field in the **Product** table.
 8. The **Description** field in the **Product** table.
 9. The **OnSale** field in the **Product** table.
 10. The **Price** field in the **Product** table.
 11. The **Price** field in the **Product** table.
 - A form parameter named `sop` can be used to specify the logical search operator as `AND` or `OR`. If not specified, the logical search operator defaults to `OR`.
 - A form parameter named `sf` can be used to specify the sort field for the search results.
 - A form parameter named `so` can be used to specify the sort order for the search results.

Here is an example of how a quick search form might be written to search the **ProductName** and **Description** fields in the **Product** table:

```
<form action="HostedStore.LassoApp?-ResponseLassoApp=search.lasso" method="post">
  <input type="hidden" name="SearchFields" value="1,8">
  <input type="text" name="Search" size="10">
  <input type="submit" value="Search">
</form>
```

13. All of the HostedStore default content pages assume a style sheet is used to change the appearance of fonts. The details of the classes are provided below. If a style sheet is not used, then HTML `` will need to be added to all of the content pages to change appearance or all of the content will appear using the default font setting in the site visitor's browser.
- `title` - Used for the title displayed at the top of every content page.
 - `subtitle` - Used to title subsections of content page.
 - `lg` - Used when a larger font size is needed. Mainly used to title sections on a page.
 - `med` - Used for most of the text throughout the store.
 - `sm` - Used when a smaller font size is needed. Mainly used to indicate wishlist purchases and product attributes.
 - `header` - Used for the header labels in tables.
 - `label` - Used for field name labels in forms.
 - `asterisk` - Used for the asterisk that is used to indicate required fields.
 - `warning` - Used to display results messages or errors.
 - `select` - Used for HTML `<select>` tags.
 - `input` - Used for HTML `<input>` tags.
 - `textarea` - Used for HTML `<textarea>` tags.
 - `invoice_title` - Used to title the invoices generated in *Order Administration*.
 - `invoice_lg` - Used for large sized text on the invoices generated in *Order Administration*.
 - `invoice_med` - Used for medium sized text on the invoices generated in *Order Administration*.
 - `invoice_sm` - Used for small sized text on the invoices generated in *Order Administration*.

Creating Store Administration Templates

Creating a *Store Administration* template is a straightforward process, but care should be taken to properly follow each of the steps.

1. Create and design the page HTML with a text editor or GUI editor.
2. For links to areas in Store Administration, use the link structures as outlined in the *Store Administration Links* section below.
3. Determine where the content pages will be inserted into the template and take the upper half of the HTML document and save it with the name *top.lasso*. Save the lower half of the template as *bottom.lasso*.
4. Create a new directory with the name of the template and place it in the */hostedstore/adminstore/templates/* directory. The *top.lasso* and *bottom.lasso* files should go into the new template directory.

Store Administration Links

Links to areas in *Store Administration* should use the link structure outlined below:

- *Admin User Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/adminuser.lasso`
- *Affiliate Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/affiliate.lasso`
- *Autoship Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/autoship.lasso`
- *Customer Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/customer.lasso`
- *Maintenance*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/maintenance.lasso`
- *Modules*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/module.lasso`
- *Order Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/order.lasso`
- *Price Group Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/pricegroup.lasso`
- *Product Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/product.lasso`
- *Promotion Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/promotion.lasso`
- *Reports*
`[HS_PathToStore]/adminstore/HostedStore_Reports.LassoApp`
- *Review*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/review.lasso`
- *Store Status*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/index.lasso`
- *Supplier Administration*
`HostedStore.LassoApp?-ResponseLassoApp=adminstore/supplier.lasso`

Content Pages

Each of the content pages contains text that can be easily modified to suit the needs of the store. Many of the content pages deal with code that is processed through form submissions. Each of those pages contains code that uses `HSP_MessageList`, an array that contains a list of error messages or other messages to provide feedback to the user. It is recommended that `HSP_MessageList` remain on those pages. All other variables that HostedStore makes available for use on the page will have an `HSP_` prefix (HostedStore Public variable). HostedStore also uses other prefixes: `HS_` (preference variables set in *Store Configuration*) and `HSI_` (HostedStore Internal variables). It is highly recommended that if custom code is added to the default content pages that the prefixes `HS_`, `HSI_`, and `HSP_` are avoided so that there are not any complications in this or future versions of HostedStore.

Note that just because a piece of information is available in a variable does not necessarily mean that that information is used in the default HostedStore content pages. The additional information is provided as a convenience for those who would like to use that information. An example of this is the Category and SubCategory fields for a product. That information is available on the product detail page, but the default HostedStore content pages do not display that information. If needed, then just make the necessary modifications to the default content pages.

If additional information is wanted, but not provided by HostedStore, then the necessary LDML code can be added to the appropriate content page. If Lasso programming is unfamiliar, here are some suggestions on how to get the modifications needed:

- Request that the feature is added to HostedStore. If the feature is of general interest, then it will likely be available in a future release.
- If the feature request is denied or if it cannot be implemented quick enough, then make a request for fee-based customization.
- Post to the HostedStore Support List and another user may provide free or fee-based guidance.
- Look on the HostedStore Web site for developers that are well-versed in LDML and HostedStore.
- Learn LDML. :) It's not too difficult and can actually be fun at times!

Each default content page is detailed below. Note that the main intent is to document the information that is available to each default content page. In general, there will not be a discussion of the actual code and code structure in the default content pages.

Sections

- [Navigation Pages](#)
- [Storefront](#)
- [Customer Administration](#)
- [Store Administration](#)
- [Affiliate Administration](#)
- [Supplier Administration](#)
- [Help Center](#)

Navigation Pages

Two global navigation pages are located in the */hostedstore/includes/navigation/* directory.

Sections

- [Browse Products](#) - Navigation menu for store template.
- [Template Footer](#) - Navigation footer for store templates.

Browse Products

/hostedstore/includes/navigation/browseproducts.lasso

This file is used in the default store templates to provide quick navigation hyperlinks to certain product categories. This file can be used to create hard-coded hyperlinks or dynamically generated hyperlinks. This file is loaded each time a page in the store is called so it is recommended that dynamically generated hyperlinks are avoided to eliminate extra overhead from the database calls. Although in stores where the product offerings change frequently it may make sense or may be absolutely necessary to dynamically generate the links.

Special parameters have been coded into the search script so that categorical hyperlinks from this file can be passed to the search script to generate a list of matching products for display to the customer. All of the parameters perform an exact match search (the search operator is **equals**) against a corresponding field in the **Product** table. Each of the parameters and the field the parameter searches are detailed below:

- **productname** matches against the **ProductName** field in the **Product** table.
- **category** matches against the **Category** field in the **Product** table.
- **subcategory** matches against the **SubCategory** field in the **Product** table.
- **subcategory2** matches against the **SubCategory2** field in the **Product** table.
- **subcategory3** matches against the **SubCategory3** field in the **Product** table.
- **department** matches against the **Department** field in the **Product** table.
- **manufacturer** matches against the **Manufacturer** field in the **Product** table.
- **onsale** matches against the **OnSale** field in the **Product** table. Use a value of **Y** to display products that are currently on sale.
- **show** matches against the **Category** field in the **Product** table, but loads the search subcategories content page (*/hostedstore/storefront/html/search_subcategories.lasso*) instead of the search results list content page (*/hostedstore/storefront/html/search_list.lasso*) like all of the other parameters. Details about the purpose of this file and how it can be modified to meet the needs of the store can be found in the [Modifying HostedStore > Content Pages > Storefront > Search SubCategories](#) section.
- **class** matches against the **RestrictClass** field in the **Product** table.

Links to the product search script should use hyperlinks formatted like: `[$HS_PathToStore] /HostedStore.LassoApp?-ResponseLassoApp=search.lasso&NAME=VALUE`

So, for example, to search the product category *Cameras*, the hyperlink would look like:

`[$HS_PathToStore] /HostedStore.LassoApp?-ResponseLassoApp=search.lasso&category=Cameras`

If an intermediate page is needed before executing the product search script (to allow the customer to select subcategories or manufacturers for a large product category, for instance) or if a way to bypass the product search script altogether is needed, then the following hyperlink can be used: `[$HS_PathToStore] /HostedStore.LassoApp?-ResponseLassoApp=searchpage.lasso`.

That page will pass through all parameters it is sent and will return the intermediate search content page (*/hostedstore/storefront/html/search_searchpage.lasso*). If a different content page should be returned, then specify the name of the page or the directory followed by the page name using a parameter named **page**. For example, to return the content page */hostedstore/storefront/html/MySearchPage.lasso*, use the following link along with any additional parameters that should be passed: `[$HS_PathToStore] /HostedStore.LassoApp?-ResponseLassoApp=searchpage.lasso&page=MySearchPage.lasso`. If multiple custom intermediate search pages will be used, it may make maintenance easier to create a directory to hold all of the search pages. If the directory name is *dir* and the page is named *MySearchPage.lasso*, then use the following link along with any additional parameters that should be passed: `[$HS_PathToStore] /HostedStore.LassoApp?-ResponseLassoApp=searchpage.lasso&page=dir/MySearchPage.lasso`.

The following code could be used to dynamically generate an alphabetical list of hyperlinks for all product categories. The code below is the code provided in the default browse products format file. However, as mentioned above, this may be less than ideal because an extra database call is performed for each page.

```
[ Inline: -Database=$HS_ProductDB,
```

```

        -SQL='SELECT DISTINCT '+$HS_ProductDB+'.Product.Category '
        + 'FROM '+$HS_ProductDB+'.Product '
        + 'WHERE '+$HS_ProductDB+'.Product.WebSell=\"Y\" '
        + 'ORDER BY '+$HS_ProductDB+'.Product.Category',
    -MaxRecords='All',
    -Username=$HS_Username,
    -Password=$HS_Password]
[Records]
    <a href="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=search.lasso&category=[Field:'Category']">
    [Field:'Category']
    </a>
    <br>
[/Records]
[/Inline]

```

Or, for those partial to LassoScript:

```

<?LassoScript
Inline: -Database=$HS_ProductDB,
        -SQL='SELECT DISTINCT '+$HS_ProductDB+'.Product.Category '
        + 'FROM '+$HS_ProductDB+'.Product '
        + 'WHERE '+$HS_ProductDB+'.Product.WebSell=\"Y\" '
        + 'ORDER BY '+$HS_ProductDB+'.Product.Category',
    -MaxRecords='All',
    -Username=$HS_Username,
    -Password=$HS_Password;
Records;
    Output: ' <a href=" '
        + $HS_PathToStore
        + '/HostedStore.LassoApp?-ResponseLassoApp=search.lasso&category='
        + (Field:'Category')
        + '"> '
        + (Field:'Category')
        + '</a><br>', -EncodeNone;
/Records;
/Inline;
?>

```

Template Footer

/hostedstore/includes/navigation/navfooter.lasso

All of the default store templates provided with HostedStore make use of this template footer file. The purpose of the file is to provide text-based hyperlinks to areas of the store. Most of the hyperlinks have counterparts at the top of the default store template file. Having one common footer file makes it easy to switch between store template files and keep the navigation intact. Use of this file is optional in custom store templates or it can even be eliminated if only one of the default store templates will be used.

Here are some hyperlinks that may be useful in a text based navigation footer:

- `[$HS_PathToStore]/HostedStore.LassoApp` for the store home content page.
- `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=search.lasso` for the product advanced search form content page.
- `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=wishlist.lasso` for the wish list search form content page.
- `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=helpcenter.lasso` for the *Help Center*.
- `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=admincustomer/index.lasso` for *Customer Account Administration*
- `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso` for the shopping cart display content page.
- `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=checkout.lasso` to checkout.

Storefront

Each of the content pages located in the */hostedstore/storefront/html/* directory are detailed below.

Sections

- [Shopping Cart Display](#)
- [Shopping Cart Modify Item](#)
- [Recover Shopping Cart](#)
- [Save Shopping Cart](#)
- [Customer Information](#)
- [Customer Login Form](#)
- [Email a Friend](#)
- [Home Page](#)
- [Order Display](#)
- [Order Result](#)
- [Modify Payment Method](#)
- [Product Detail](#)
- [Review Detail](#)
- [Review Form](#)
- [Review List](#)
- [Search Form](#)
- [Search List](#)
- [Intermediate Search Page](#)
- [Search SubCategories](#)
- [Shipping Info](#)
- [Modify Shipping Info](#)
- [Store Closed](#)
- [Wish List Detail](#)
- [Wish List List](#)
- [Wish List Login Form](#)
- [Wish List Search Form](#)

Shopping Cart Display

/hostedstore/storefront/html/cart_display.lasso

HSP_EnforceInventory is an array of arrays that contains information about ordered items with a quantity that exceeds the current inventory. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ProductID** field for the item from the **Item** table.
2. The **ProductName** field for the item from the **Item** table.
3. The **Quantity** field for the item from the **Item** table.
4. The **OnHand** field for the item from the **Product** table.

HSP_RestrictedProduct is an array of arrays that contains information about ordered items that cannot be purchased because the restriction requirements have not been met. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ProductID** field for the item from the **Item** table.
2. The **ProductName** field for the item from the **Item** table.
3. The **RestrictTo** field for the item from the **Item** table.

HSP_LimitedProduct is an array of arrays that contains information about ordered items that cannot be purchased because the quantity limitations have not been met. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ProductID** field for the item from the **Item** table.
2. The **ProductName** field for the item from the **Item** table.
3. The **Quantity** field for the item from the **Item** table.
4. The **PerOrderMin** field for the product from the **Product** table.
5. The **PerOrderMax** field for the product from the **Product** table.
6. The **PerCustomerMax** field for the product from the **Product** table less the sum of all quantities ordered previously (the sum of the **Quantity** field from the **Item** table for all past orders and the current order).

Important: The treatment of inventory enforcement, restricted products, and order limits on this page assumes that all three act independently of each other. Writing a generalized case for the code logic was not possible because of multiple interaction possibilities between each of the features. It may be necessary to change the script logic to account for cases where two or more of the features will act in unison. Otherwise some of the messages may lead to confusion. For example, if there are only 2 of product X in inventory, the minimum order quantity is 5, and the customer orders 3, then two conflicting messages are displayed. The first message advises the customer that only 2 of the item are inventory so 1 needs to be removed. The second message notifies the customer that at least 5 of the product must be ordered so at least 2 more should be added to the cart. Clearly these two messages conflict.

HSP_Cart is an array of values and arrays that contains information about the ordered items. The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The value that should be passed as a form parameter named **modify** to the product modification content page. Hyperlinks to the product modification content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=modifyitem.lasso&modify=VALUE`. Note that this link is only relevant for products that have attributes.
 2. The value that should be passed as a form parameter named **ID** to the product modification

content page. Hyperlinks to the product modification content page should use:

`[$HS_PathToStore]/HostedStore.LassoApp?-`

`ResponseLassoApp=detail.lasso&ID=VALUE`. Note that this link is only relevant for products that do not have attributes.

3. The **ProductID** field for the item from the **Item** table.
4. The **ProductName** field for the item from the **Item** table.
5. Has a value of **y** if the item was purchased from a wishlist.
6. An array that contains the names of the item attributes if any exist.
7. An array that contains the chosen value for item attributes if any exist.
8. The **Quantity** field for the item from the **Item** table.
9. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
10. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
2. The calculated total item count for the order.
3. The calculated subtotal for the order formatted according to currency formatting preferences specified in *Store Configuration*.
4. The calculated shipping for the order formatted according to currency formatting preferences specified in *Store Configuration*.
5. The calculated total for the order (based on no tax) formatted according to currency formatting preferences specified in *Store Configuration*.
6. An array that contains tax and order totals for all taxable states/provinces specified in *Store Configuration*. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The state/province to which the tax and order total correspond.
 2. The calculated tax for the order (based on the state/province) formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The calculated total for the order (based on the state/province) formatted according to currency formatting preferences specified in *Store Configuration*.

This content page contains a form for submitting the information necessary to email a quote (a copy of the shopping cart) if the `HS_UseQuotes` preference variable has been set to a value of **y** in *Store Configuration*. Unlike other HostedStore forms, the form input elements are not named numerically. Each form element is detailed below:

- **Email** - The email address of the quote recipient. (**REQUIRED**)

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=cart.lasso` and should not be modified if the default shopping cart function capabilities are to be used. If a third party module is used to perform shopping cart functions, then consult the module documentation to modify the form action parameter as needed.

The name parameters of the submit input elements are `Submitted_Update` and `Submitted_EmailQuote` (for shopping cart update and email a quote, respectively) and should not be changed. The value parameters can be changed to suit the needs of the store. If a third party module is used to perform cart functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Shopping Cart Modify Item

/hostedstore/storefront/html/cart_modifyitem.lasso

HSP_Product is an array that has information about the product that is to be modified. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named **ID** to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. An array of prices in different formats. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field for the product from the **Product** table depending upon which applies).
 2. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field for the product from the **Product** table depending upon which applies) formatted according to currency formatting preferences specified in *Store Configuration*.
 3. An HTML formatted list of all of the prices (regular, sale and/or member).
6. An array of arrays that contains the item attributes. The number of arrays corresponds to the number of attributes for the item. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The attribute name
 2. The attribute values
 3. The attribute prices
 4. The type of attribute description file (if any).
 5. A single value for an attribute used internally.
 6. A single value for an attribute displayed to the user.
7. **Y** if some of the attributes has associated description files.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=modifyitem.lasso` and should not be modified if the default product modification capabilities are to be used. If a third party module is used to perform product modification, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_ModifyItem` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform product modification, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Recover Shopping Cart

/hostedstore/storefront/html/cart_recoverform.lasso

`HSP_CartRecovered` has a value of `Y` if the shopping cart has been successfully recovered. This variable allows for easy switching of page content depending on whether the user is still filling out the form to recover the shopping cart or if the shopping cart has already been recovered.

This content page also contains a form for submitting the information necessary to recover a saved cart. Like all HostedStore forms, the form input elements are named numerically and prepended with an `f`. All of the items are required so should not be removed (but can certainly be repositioned on the form. Each numerical form element is detailed below:

- `f1` - The name to assign to the shopping cart (*REQUIRED*).
- `f2` - The customer's email address (*REQUIRED*).
- `f3` - Set to a value of `Y` if the items currently in the shopping cart (if any) should be removed before adding the saved shopping cart items to the shopping cart.
- `f4` - Set to a value of `Y` if the saved shopping cart should be deleted after adding the saved shopping cart items to the shopping cart. Note that if any errors were encountered while adding the saved shopping cart items (could not find saved shopping cart, item is no longer for sale, item attributes have changed, etc), then the saved shopping cart will not be deleted.

The form action parameter for this content page is `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cartrecover.lasso` and should not be modified if the default savable shopping cart features are to be used. If a third party module is used to perform savable shopping cart functions, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_RecoverCart` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform savable shopping cart functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Save Shopping Cart

`/hostedstore/storefront/html/cart_saveform.lasso`

`HSP_CartSaved` has a value of `Y` if the shopping cart has been successfully saved. This variable allows for easy switching of page content depending on whether the user is still filling out the form to save the shopping cart or if the shopping cart has already been saved.

This content page also contains a form for submitting the information necessary to save a shopping cart for later retrieval. Like most HostedStore forms, the form input elements are named numerically and prepended with an `f`. All of the items are required so should not be removed (but can certainly be repositioned on the form. Each numerical form element is detailed below:

- `f1` - The name to assign to the shopping cart (*REQUIRED*).
- `f2` - The customer's email address (*REQUIRED*).

The form action parameter for this content page is `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cartsave.lasso` and should not be modified if the default savable shopping cart features are to be used. If a third party module is used to perform savable shopping cart functions, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_SaveCart` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform savable shopping cart functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Customer Information

/hostedstore/storefront/html/customer_info.lasso

The page also contains a form for creating the billing information for a new customer account. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - First name (*REQUIRED*)
- `f2` - Last name (*REQUIRED*)
- `f3` - Company
- `f4` - Address (*REQUIRED*)
- `f5` - Additional address line
- `f6` - City (*REQUIRED*)
- `f7` - State/Province (*REQUIRED*)
- `f8` - Postal Code (*REQUIRED*)
- `f9` - Country (*REQUIRED*)
- `f10` - Email address (*REQUIRED*, but it was already submitted when the customer created their account)
- `f11` - Phone number (*REQUIRED*)
- `f12` - Fax number
- `f13` - A checkbox that enables the customer to specify that the billing information submitted should be used for the shipping information too.
- `f15` - A checkbox that enables the customer to subscribe to the store newsletter. This requires that the `HS_Newsletter` preference variable is set to `y` in *Store Configuration*.

Two hidden input elements are at the bottom of the form. These should not be modified or removed!

- `f10` is the form parameter that contains the email address for the account.
- `f14` is the form parameter that contains the password for the account.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=checkout.lasso` and should not be modified if the default customer account creation capabilities are to be used. If a third party module is used to perform customer account creation, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_CreateAccount` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform customer account creation, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Customer Login Form

/hostedstore/storefront/html/customer_login.lasso

The page also contains a form for logging into an existing customer account or for creating a new customer account. Login is necessary to ensure that the order is associated with the proper customer. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - Radio button that specifies whether the customer is a new or existing customer. (*REQUIRED*)
- `f2` - The email address for the customer account. (*REQUIRED*)
- `f3` - The verified email address for the customer account. (*REQUIRED* for new customers)
- `f4` - The password for the customer account. (*REQUIRED*)
- `f5` - The verified password for the customer account. (*REQUIRED* for new customers)
- `f6` - A checkbox that enables existing customers to have their forgotten password emailed to the email address given.

The checkbox can be eliminated from the form. However, it is a convenient feature for users that may have forgotten their password so consider carefully whether or not to remove it. Having the checkbox can also reduce the number of customer service requests regarding forgotten passwords.

Note that there is a little trickery involved to use form input elements named `f2` and `f4` twice depending upon if the customer is new or existing.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=checkout.lasso` and should not be modified if the default customer account login capabilities are to be used. If a third party module is used to perform customer account login, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Continue` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform customer account login, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Email a Friend

`/hostedstore/storefront/html/email_friend.lasso`

`HSP_EmailSent` has a value of `Y` if an email has been successfully sent. This variable allows for easy switching of page content depending on whether the user is still filling out the form to send the email or if the email has already been sent.

`HSP_Product` is an array that contains information about the product that will be emailed. It is placed on the page by default as a reminder of product details for the user sending the email. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.

The page also contains a form for submitting the information necessary to process the email. Like all HostedStore forms, the form input elements are named numerically and prepended with an `f`. Some of the items are required so should not be removed (but can certainly be repositioned on the form). Each numerical form element is detailed below:

- `f1` - The sender's email address (*REQUIRED*).
- `f2` - The recipient's email address (*REQUIRED*).
- `f3` - A checkbox to BCC the sender.
- `f4` - A personal message to include in the email body.

Three hidden input elements are at the bottom of the form. These should not be modified or removed!

- `ID` is the form parameter of the same name that specifies which product is to be emailed.
- `ProductName` is the name of the product to be emailed.

The body of the email message is specified by the file located at `/hostedstore/storefront/email/email_friend.lasso`. Details regarding modification of that file are provided in the [Modifying HostedStore](#) > [Email Messages](#) > [Storefront](#) > [Email a Friend](#) section.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=emailfriend.lasso` and should not be modified if the default email a friend functions are to be used. If a third party module is used to perform email a friend functions, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Email` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform email a friend functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Home Page

`/hostedstore/storefront/html/index.lasso`

`HSP_Featured` is an array of arrays. The number of arrays is determined by the number of products that have been marked as featured in the **Product** table and the value of the `HS_FeaturedLimit` preference variable set in *Store Configuration*. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`. Hyperlinks to the shopping cart display content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`.
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.

`HSP_Favorites` is an array of arrays. The number of arrays is determined by the value of the `HS_FavoritesLimit` preference variable set in *Store Configuration*. The array contains the top selling products based off of historical sales so it is kind of self-fulfilling in nature if customers purchase products based off of inclusion in this list. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`. Hyperlinks to the shopping cart display content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`.
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.

If favorites are not wanted, `HSP_Favorites` can be prevented from displaying by removing the code or wrapping it in a statement that verifies the `HS_DisplayFavorites` preference variable set in *Store Configuration* has a value of `Y`. Then set the `HS_DisplayFavorites` preference variable value to `N`. If favorites are wanted at a future date, then it is a simple matter of changing the `HS_DisplayFavorites` preference variable value back to `Y`.

Additional Form Parameters

An optional form parameter, `CartAddQty`, can be added to the form or URL that adds items to the shopping cart. The quantity specified by the parameter value will be the quantity of the item to add to the shopping cart. This parameter can be used in a number of ways:

- **As a select list** - By using a select list, the customer will be able to choose a quantity to add to the shopping cart. The following example allows for the selection of between 1 and 25 of an item:

```
<select name="CartAddQty">
  [Loop:25]
```

```

    <option value="[Loop_Count]">
  [/Loop]
</select>

```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

Example for featured items

```

<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Featured->Get:Loop_Count->Get:1]">
  <select name="CartAddQty">
    [Loop:25]
    <option value="[Loop_Count]">
  [/Loop]
  </select>
  <input type="submit" value="Add to Cart">
</form>

```

Example for favorite items

```

<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Favorites->Get:Loop_Count->Get:1]">
  <select name="CartAddQty">
    [Loop:25]
    <option value="[Loop_Count]">
  [/Loop]
  </select>
  <input type="submit" value="Add to Cart">
</form>

```

- **As a text box** - By using a text box, the customer will be able to specify any quantity to add to the shopping cart.

```

<input type="text" name="CartAddQty" size="5" maxlength="5">

```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

Example for featured items

```

<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Featured->Get:Loop_Count->Get:1]">
  <input type="text" name="CartAddQty" size="5" maxlength="5">
  <input type="submit" value="Add to Cart">
</form>

```

Example for favorite items

```

<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Favorites->Get:Loop_Count->Get:1]">
  <input type="text" name="CartAddQty" size="5" maxlength="5">
  <input type="submit" value="Add to Cart">
</form>

```

- **As a URL** - By using a URL, images or hyperlinked text can be used to add a fixed quantity of an item to the shopping cart. The following example adds 10 of a non-configurable item to the shopping cart:

Example for featured items

```
<a href="[$HS_PathToStore]/HostedStore.LassoApp?-  
ResponseLassoApp=cart.lasso&add=[$HSP_Featured->Get:Loop_Count-  
>Get:1]&AddCartQty=10">Buy 10 Now!</a>
```

Example for favorite items

```
<a href="[$HS_PathToStore]/HostedStore.LassoApp?-  
ResponseLassoApp=cart.lasso&add=[$HSP_Favorites->Get:Loop_Count-  
>Get:1]&AddCartQty=10">Buy 10 Now!</a>
```

Order Display

/hostedstore/storefront/html/order_display.lasso

`HSP_Order` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table.
 3. The **Expiration** field for the order from the **Invoice** table.
 4. The **Routing** field for the order from the **Invoice** table.
 5. The **CheckNumber** field for the order from the **Invoice** table.
 6. The **PONumber** field for the order from the **Invoice** table.
2. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
3. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The value that should be passed as a form parameter named `modify` to the product modification content page. Hyperlinks to the product modification content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=modifyitem.lasso&modify=VALUE`
 2. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`. Note that this link is only relevant for products that do not have attributes.
 3. The **ProductID** field for the item from the **Item** table.
 4. The **ProductName** field for the item from the **Item** table.
 5. Has a value of `y` if the item was purchased from a wishlist.
 6. An array that contains the names of the item attributes if any exist.
 7. An array that contains the chosen value for item attributes if any exist.
 8. The **Quantity** field for the item from the **Item** table.
 9. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
4. The **ItemCount** field for the order from the **Invoice** table.
5. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
6. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
7. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
8. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
9. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.

The appropriate form based on the payment gateway used will be substituted toward the bottom of the page.

The name parameter of the submit input element is `Submitted_Purchase` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform payment handling, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Order Printer Friendly Invoice

`/hostedstore/storefront/html/order_printinvoice.lasso`

`HSP_OrderConfirmation` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array that contains the billing information for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Customer** table.
 2. The **LastName** field for the order from the **Customer** table.
 3. The **Company** field for the order from the **Customer** table.
 4. The **Address1** field for the order from the **Customer** table.
 5. The **Address2** field for the order from the **Customer** table.
 6. The **City** field for the order from the **Customer** table.
 7. The **State** field for the order from the **Customer** table.
 8. The **Postal** field for the order from the **Customer** table.
 9. The **Country** field for the order from the **Customer** table.
 10. The **EmailAddress** field for the order from the **Customer** table.
 11. The **Phone** field for the order from the **Customer** table.
 12. The **Fax** field for the order from the **Customer** table.
2. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
 10. The **EmailAddress** field for the order from the **Invoice** table.
 11. The **Phone** field for the order from the **Invoice** table.
 12. The **Fax** field for the order from the **Invoice** table.
3. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table.
 3. The month portion of the **Expiration** field for the order from the **Invoice** table.
 4. The year portion of the **Expiration** field for the order from the **Invoice** table.
 5. *NOT USED*
 6. The **Routing** field for the order from the **Invoice** table.
 7. The **CheckNumber** field for the order from the **Invoice** table.
 8. The **PONumber** field for the order from the **Invoice** table.
 9. The **PromotionCode** field for the order from the **Invoice** table.
4. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ProductName** field for the item from the **Item** table.
 2. Has a value of `y` if the item was purchased from a wishlist.
 3. An array that contains the names of the item attributes if any exist.
 4. An array that contains the chosen value for item attributes if any exist.
 5. The **Quantity** field for the item from the **Item** table.
 6. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. An array of arrays that contains the order totals information for the order. The elements of the array

are detailed below (the number corresponds to the position in the array):

1. The **ItemCount** field for the order from the **Invoice** table.
2. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
3. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
4. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
6. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
6. NOT USED - This is an artifact of using common code. The value here does not apply to complete orders.

Order Result

/hostedstore/storefront/html/order_result.lasso

This content page is returned after an order transaction has been processed. `HSP_Approved` will have a value of `y` if the transaction was approved.

If the transaction was approved, then `HSP_OrderConfirmation`, an array of values and arrays that contains information about the order, can be used to show a receipt to the customer (an order confirmation receipt is also emailed). The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array that contains the billing information for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the customer from the **Customer** table.
 2. The **LastName** field for the customer from the **Customer** table.
 3. The **Company** field for the customer from the **Customer** table.
 4. The **Address1** field for the customer from the **Customer** table.
 5. The **Address2** field for the customer from the **Customer** table.
 6. The **City** field for the customer from the **Customer** table.
 7. The **State** field for the customer from the **Customer** table.
 8. The **Postal** field for the customer from the **Customer** table.
 9. The **Country** field for the customer from the **Customer** table.
 10. The **EmailAddress** field for the customer from the **Customer** table.
 11. The **Phone** field for the customer from the **Customer** table.
 12. The **Fax** field for the customer from the **Customer** table.
2. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
 10. The **EmailAddress** field for the order from the **Invoice** table.
 11. The **Phone** field for the order from the **Invoice** table.
 12. The **Fax** field for the order from the **Invoice** table.
3. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table.
 3. The month portion of the **Expiration** field for the order from the **Invoice** table.
 4. The year portion of the **Expiration** field for the order from the **Invoice** table.
 5. *NOT USED*
 6. The **Routing** field for the order from the **Invoice** table.
 7. The **CheckNumber** field for the order from the **Invoice** table.
 8. The **PONumber** field for the order from the **Invoice** table.
 9. The **PromotionCode** field for the order from the **Invoice** table.
4. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. NOT USED - This is an artifact of using common code. The value here does not apply to complete orders.
 2. The **ProductName** field for the item from the **Item** table.
 3. Has a value of `y` if the item was purchased from a wishlist.
 4. An array that contains the names of the item attributes if any exist.

5. An array that contains the chosen value for item attributes if any exist.
6. The **Quantity** field for the item from the **Item** table.
7. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
8. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. An array of arrays that contains the quantity and monetary totals information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ItemCount** field for the order from the **Invoice** table.
 2. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 4. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 5. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 6. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
6. The value that should be passed as a form parameter named **ID** to the printer friendly invoice page. Hyperlinks to the printer friendly invoice content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=orderprintinvoice.lasso&ID=VALUE`. To open the receipt in a new browser window, use `target="_blank"` as a parameter in the hyperlink.

Modify Payment Method

/hostedstore/storefront/html/payment_modify.lasso

This content page contains a form for submitting the information necessary to modify the billing information for an order. Like other HostedStore forms, the form input elements are named numerically and prepended with an *f*. However, unlike other forms, the numbering does not start with 1. Each form element is detailed below:

- *f14* - The credit card type. (*REQUIRED* if *f21* has a value of *Credit Card*)
- *f15* - The credit card number. (*REQUIRED* if *f21* has a value of *Credit Card*)
- *f16* - The credit card verification code. (*REQUIRED* if *HS_UseSecurityCode* is set to *Y* in the */hostedstore/includes/HS_datafile.lasso* store configuration file and *f21* has a value of *Credit Card*).
- *f17* - The month of credit card expiration. (*REQUIRED* if *f21* has a value of *Credit Card*)
- *f18* - The year of credit card expiration. (*REQUIRED* if *f21* has a value of *Credit Card*)
- *f21* - The payment method. (*REQUIRED*) Supported values are *Check*, *Check by Mail/Phone*, *Credit Card*, *PayPal*, and *Purchase Order*.
- *f22* - The checking account number. (*REQUIRED* if *f21* has a value of *Check*)
- *f23* - The checking account routing number. (*REQUIRED* if *21* has a value of *Check*)
- *f24* - The check number. (*REQUIRED* if *f21* has a value of *Check*)
- *f25* - The purchase order number. (*REQUIRED* if *f21* has a value of *Purchase Order*)

The form action parameter for this content page is *HostedStore.LassoApp?-ResponseLassoApp=modifypayment.lasso* and should not be modified if the default billing information modification capabilities are to be used. If a third party module is used to perform billing information modification, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is *Submitted_Update* and should not be changed, but the value parameter can be changed to suit the needs of the store. If a third party module is used to perform billing information modification, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Product Detail

/hostedstore/storefront/html/product_detail.lasso

`HSP_Product` is an array that has information about the product. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `add` to the product review content page or the shopping cart display content page. Hyperlinks to the product review content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=review.lasso&add=VALUE`
Hyperlinks to the shopping cart display content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. An array of prices in different formats. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field from the **Product** table depending upon which applies).
 2. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field from the **Product** table depending upon which applies) formatted according to currency formatting preferences specified in *Store Configuration*.
 3. An HTML formatted list of all of the prices (regular, sale and/or member).
 4. A map of arrays that contains price type, unformatted prices, formatted prices, and discounts for prices that apply to a product. Possible map keys are MSRP, Standard, Member, Sale, and Base Cost. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. Price as a *DECIMAL* type.
 2. Price formatted according to currency formatting preferences specified in *Store Configuration*.
 3. Price group type applied (if applicable). Possible values are: Discount (%), At Retail, Discount from Retail (%), At Cost, and Markup Cost (%).
 4. Discount (the difference between full price and the price) as a *DECIMAL* type.
 5. Discount formatted according to currency formatting preferences specified in *Store Configuration*.
 6. Y/N flag that indicates whether or not the price was used as the final price that is displayed to the customer.
6. This will either be a text message of `No reviews at this time.` if there are not any reviews for the product or an array that contains a summary of the product reviews. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The number of reviews for the product.
 2. The average rating for all of the reviews.
7. An array of arrays that contains information about complementary products if any exist. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. An array of the values that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
 2. An array of the **ProductID** fields for the products from the **Product** table.
 3. An array of the **ProductName** fields for the products from the **Product** table.
 4. An array of integers that corresponds to the number of the complement field. Product complements from the **Complement1** field will have a value of `1`, product complements from the **Complement2** field will have a value of `2`, etc.
8. An array of arrays that contains the item attributes. The number of arrays corresponds to the number of attributes for the item. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The attribute name.

2. The attribute values.
3. The attribute prices.
4. The type of attribute description file (if any).
5. A single value for an attribute used internally.
6. A single value for an attribute displayed to the user.
9. A value of **Y** if the product has attributes with description files.
10. A value that must be passed in the product configuration form (if present) as a hidden form element named `options`.
11. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field from the **Product** table depending upon which applies) formatted according to currency formatting preferences specified in *Store Configuration*.
12. A value of **Y** if the `HS_UseZoomImages` preference variable is set to **Y** in the *Store Configuration* and an image for the product exists in the `/hostedstore/images/products/zoom/` directory.
13. An array of additional fields and calculations for the product record that are not used in the default product detail page. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **Category** field for the item from the **Product** table.
 2. The **SubCategory** field for the item from the **Product** table.
 3. The **SubCategory2** field for the item from the **Product** table.
 4. The **SubCategory3** field for the item from the **Product** table.
 5. The **Department** field for the item from the **Product** table.
 6. The **Manufacturer** field for the item from the **Product** table.
 7. The **OnHand** field for the item from the **Product** table.
 8. The **WebSellQty** field for the item from the **Product** table.
 9. The **OutOfStockDisplay** field for the item from the **Product** table if the **OnHand** field from the **Product** table has a quantity less than or equal to zero. A value of **Y** indicates that the out of stock product should be displayed, but not sold.
 10. The **OutOfStockMessage** field for the item from the **Product** table if the **OnHand** field from the **Product** table has a quantity less than or equal to zero. If the **OutOfStockMessage** field is empty, then the message defaults to the message set with the `HS_OutOfStockMessage` preference variable set in *Store Configuration*.
 11. The **MemberOnlyItem** field for the item from the **Product** table.
 12. The sales ranking for the item. The sales ranking is determined where the item is located a list of item generated by sorting the entire product catalog by the **QuantitySold** field for the item from the **Product** table. If two items have the same value for the **QuantitySold** field, then both items will have the same ranking.
 13. The age of the product record (the number of days between the current date and date in the **DateCreated** field from the **Product** table). This value can be compared against an arbitrary value to display a **New!** message or image.

For example, if products that are less than 30 days old are to be displayed with an image named `new.gif` from the `/hostedstore/images/` directory, then the following code could be used:

```
[ If: ($HSP_Product->Get:13->Get:13) < 30 ]
  
[/If]
```

Or, for shared SSL certificate stores:

```
[ If: ($HSP_Product->Get:13->Get:13) < 30 ]
  
[/If]
```

This content page contains a form that allows a customer to add their name to a wait list to receive email notification when an out of stock product is in stock again. On the default content page this form is only displayed if the product is out of stock and the product has been setup in *Product Administration* to display but not sell the product. Each form element is detailed below:

- `f1` - The email address of the customer. (**REQUIRED**)

A hidden input element is at the bottom of the form. It should not be modified or removed!

- `ID` is the form parameter that specifies which product to return in the product detail page.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=detail.lasso` and should not be modified if the default wait list capabilities are to be used. If a third party module is used to perform wait list functions, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_EmailNotify` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform wait list functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

After the form is submitted, the same product detail content page will be returned. If the customer's email address was successfully added to the wait list, then both the `Submitted_EmailNotify` and `f1` variables will have a value. If there was a problem adding the email address to the wait list, then only the `Submitted_EmailNotify` variable will have a value. By checking each variable for a value an appropriate response message can be returned to the customer.

This content page also contains a form for submitting the information necessary to customize a product. Each form element is detailed below:

- `f1, f2, ..., fn` - The nth attribute for the product. (*REQUIRED*)

Two hidden input elements are at the bottom of the form. Both should not be modified or removed!

- `add` is the form parameter that specifies which product is to be added to the cart or wishlist.
- `options` is the form parameter that specifies the number of attributes for a product.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=cart.lasso` and should not be modified if the default product configuration capabilities are to be used. If a third party module is used to perform product configuration, then consult the module documentation to modify the form action parameter as needed.

The name parameters of the submit input elements are `Submitted_Configured` and `Submitted_Wishlist` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform product configuration, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Additional Form Parameters

An optional form parameter, `CartAddQty`, can be added to the form or URL that adds items to the shopping cart. The quantity specified by the parameter value will be the quantity of the item to add to the shopping cart. This parameter can be used in a number of ways:

- **As a select list** - By using a select list, the customer will be able to choose a quantity to add to the shopping cart. The following example allows for the selection of between 1 and 25 of an item:

```
<select name="CartAddQty">
  [Loop:25]
  <option value="[Loop_Count]">
[/Loop]
</select>
```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Product->Get:1]">
  <select name="CartAddQty">
    [Loop:25]
    <option value="[Loop_Count]">
    [/Loop]
  </select>
  <input type="submit" value="Add to Cart">
</form>
```

- **As a text box** - By using a text box, the customer will be able to specify any quantity to add to the shopping cart.

```
<input type="text" name="CartAddQty" size="5" maxlength="5">
```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_Product->Get:1]">
  <input type="text" name="CartAddQty" size="5" maxlength="5">
  <input type="submit" value="Add to Cart">
</form>
```

- **As a URL** - By using a URL, images or hyperlinked text can be used to add a fixed quantity of an item to the shopping cart. The following example adds 10 of a non-configurable item to the shopping cart:

```
<a href="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso&add=[$HSP_Product->Get:1]&AddCartQty=10">Buy 10
Now!</a>
```

Review Detail

/hostedstore/storefront/html/review_detail.lasso

`HSP_Product` is an array that has information about the product that was reviewed. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
Hyperlinks to the shopping cart display content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. The **Attributes** field for the product from the **Product** table.

`HSP_ReviewDetail` is an array that contains the product review information. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The number of times the review has been read.
2. The number of times the review has been rated as helpful.
3. The date portion of the **DateCreated** field for the review from the **Review** table formatted according to date formatting preferences specified in *Store Configuration*.
4. The time portion of the **DateCreated** field for the review from the **Review** table formatted according to time formatting preferences specified in *Store Configuration*.
5. The **Author** field for the review from the **Review** table.
6. The **Rating** field for the review from the **Review** table.
7. The **ReviewText** field for the review from the **Review** table.

`HSP_ReviewList` is an array of arrays. The number of arrays is determined by the number of reviews that exist for the product. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `review` to the review detail content page. Hyperlinks to the review detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=review.lasso&read=VALUE`
2. The **DateCreated** field for the review from the **Review** table.
3. The **Author** field for the review from the **Review** table.
4. The **Rating** field for the review from the **Review** table.
5. The **TimesRead** field for the review from the **Review** table.
6. The **TimesHelpful** field for the review from the **Review** table.

The page also contains a form for submitting the information necessary to rate a review. A hidden input element is at the bottom of the form. This should not be modified or removed!

- `rate` is the form parameter that contains the variable `read` which identifies the review to rate.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=review.lasso` and should not be modified if the default review rating capabilities are to be used. If a third party module is used to perform review rating functions, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Helpful` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform review rating functions, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Review Form

`/hostedstore/storefront/html/review_form.lasso`

`HSP_Product` is an array that has information about the product that is to be reviewed. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail page or as a form parameter named `add` to the cart page. Hyperlinks to the product detail page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
Hyperlinks to the cart page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`
2. The **ProductID** field for the product.
3. The **ProductName** field for the product.
4. The **BriefDescription** field for the product formatted according to whether or not it contains HTML.

This content page contains a form for submitting the information necessary to login to add a product review. Login is necessary if the `HS_LoginForProductReviews` preference variable is set to `Y` in *Store Configuration*. Login will only be possible if the reviewer has been a customer and hence created a customer account. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - The email address of the reviewer. (*REQUIRED*)
- `f2` - The password of the reviewer. (*REQUIRED*)
- `f3` - A checkbox that enables the reviewer to have their forgotten password emailed to the email address given.

The checkbox can be eliminated from the form. However, it is a convenient feature for users that may have forgotten their password so consider carefully whether or not to remove it. Having the checkbox can also reduce the number of customer service requests regarding forgotten passwords.

The content page also contains a form for submitting the information necessary to post a review. Like all HostedStore forms, the form input elements are named numerically and prepended with an `f`. Some of the items are required so should not be removed (but can certainly be repositioned on the form. Each numerical form element is detailed below:

- `f1` - The reviewer's name.
- `f2` - The rating. (*REQUIRED*)
- `f3` - The review text. (*REQUIRED*)

A hidden input element is at the bottom of the form. It should not be modified or removed!

- `add` is the form parameter of the same name that specifies which product is to be reviewed.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=review.lasso` and should not be modified if the default review capabilities are to be used. If a third party module is used to perform reviews, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_PostReview` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform reviews, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Review List

/hostedstore/storefront/html/review_list.lasso

`HSP_Product` is an array that has information about the product that was reviewed. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
Hyperlinks to the shopping cart display content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. The **Attributes** field for the product from the **Product** table.

`HSP_ReviewList` is an array of arrays. The number of arrays is determined by the number of reviews that exist for the product. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `review` to the review detail content page. Hyperlinks to the review detail page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=review.lasso&read=VALUE`
2. The **DateCreated** field for the review from the **Review** table.
3. The **Author** field for the review from the **Review** table.
4. The **Rating** field for the review from the **Review** table.
5. The **TimesRead** field for the review from the **Review** table.
6. The **TimesHelpful** field for the review from the **Review** table.

Search Form

/hostedstore/storefront/html/search_form.lasso

`HSP_CatalogSize` contains the number of products that have been flagged for selling in the store (the **WebSell** field from the **Product** table has a value of `Y`).

This content page contains a form for submitting the information necessary to search the product catalog. Like other HostedStore forms, the form input elements are named numerically. Not all form elements may be wanted, so only those wanted need be added. Each form element is detailed below:

- `s1` - The **ProductName** field for the product in the **Product** table.
- `s2` - The **Category** field for the product in the **Product** table.
- `s3` - The **SubCategory** field for the product in the **Product** table.
- `s4` - The **SubCategory2** field for the product in the **Product** table.
- `s5` - The **SubCategory3** field for the product in the **Product** table.
- `s6` - The **Department** field for the product in the **Product** table.
- `s7` - The **Manufacturer** field for the product in the **Product** table.
- `s8` - The **Description** field for the product in the **Product** table.
- `s9` - The **OnSale** field for the product in the **Product** table.
- `s10` - Verifies that the submitted price is greater than the value in the **Price** field for the product in the **Product** table.
- `s11` - Verifies that the submitted price is less than the value in the **Price** field for the product in the **Product** table.
- `s12` - The **RestrictClass** field for the product in the **Product** table.

A second set of form parameters is provided to allow for the creation of complex forms that allow the user to specify the search operator (*begins with*, *ends with*, *contains*, etc) for each field. The numbering system corresponds to the form parameters that map to the fields with `op` appended. The default for search operators is *begins with*. Each form element is detailed below:

- `s1op` - The search operator for the **ProductName** field in the **Product** table.
- `s2op` - The search operator for the **Category** field in the **Product** table.
- `s3op` - The search operator for the **SubCategory** field in the **Product** table.
- `s4op` - The search operator for the **SubCategory2** field in the **Product** table.
- `s5op` - The search operator for the **SubCategory3** field in the **Product** table.
- `s6op` - The search operator for the **Department** field in the **Product** table.
- `s7op` - The search operator for the **Manufacturer** field in the **Product** table.
- `s8op` - The search operator for the **Description** field in the **Product** table.
- `s9op` - The search operator for the **OnSale** field in the **Product** table.
- `s10op` - The search operator for the **Price** field in the **Product** table.
- `s11op` - The search operator for the **Price** field in the **Product** table.
- `s12op` - The search operator for the **RestrictClass** field in the **Product** table.

A third set of form parameters is provided to allow for the creation of forms that use checkboxes instead of a text form element. The numbering system corresponds to the form parameters that map to the fields with `cb` appended. Simply assign a value, `Y` for instance, to denote that the field will have data submitted from checkboxes. The corresponding field will then be searched with all values using a contains operator. Each form element is detailed below:

- `s1cb` - Checkboxes for the **ProductName** field in the **Product** table.
- `s2cb` - Checkboxes for the **Category** field in the **Product** table.
- `s3cb` - Checkboxes for the **SubCategory** field in the **Product** table.
- `s4cb` - Checkboxes for the **SubCategory2** field in the **Product** table.
- `s5cb` - Checkboxes for the **SubCategory3** field in the **Product** table.
- `s6cb` - Checkboxes for the **Department** field in the **Product** table.

- `s7cb` - Checkboxes for the **Manufacturer** field in the **Product** table.
- `s8cb` - Checkboxes for the **Description** field in the **Product** table.

A fourth set of form parameters is provided to allow for the creation of forms that allow multiple criteria to be specified in a text or textarea form element. The numbering system corresponds to the form parameters that map to the fields with `d` appended. Assign the delimiter used to the parameter. For example, if a comma will be used to separate multiple criteria for the **Description** field in the **Product** table, then the `s8d` parameter should be assigned a value of `,`. The corresponding field will then be searched with all values using a `contains` operator. Each form element is detailed below:

- `s1d` - Multiple criteria delimiter for the **ProductName** field in the **Product** table.
- `s2d` - Multiple criteria delimiter for the **Category** field in the **Product** table.
- `s3d` - Multiple criteria delimiter for the **SubCategory** field in the **Product** table.
- `s4d` - Multiple criteria delimiter for the **SubCategory2** field in the **Product** table.
- `s5d` - Multiple criteria delimiter for the **SubCategory3** field in the **Product** table.
- `s6d` - Multiple criteria delimiter for the **Department** field in the **Product** table.
- `s7d` - Multiple criteria delimiter for the **Manufacturer** field in the **Product** table.
- `s8d` - Multiple criteria delimiter for the **Description** field in the **Product** table.

A fifth set of form parameters is provided to control the logical operator used for fields that use checkbox selections or multiple criteria. If all of the checked values or criteria must be true, then the form parameter should have a value of `AND`. If any of the checked values or criteria can be true, then the form parameter should have a value of `OR`. The numbering system corresponds to the form parameters that map to the fields with `lop` appended. Each form element is detailed below:

- `s1lop` - Logical operator for the **ProductName** field in the **Product** table.
- `s2lop` - Logical operator for the **Category** field in the **Product** table.
- `s3lop` - Logical operator for the **SubCategory** field in the **Product** table.
- `s4lop` - Logical operator for the **SubCategory2** field in the **Product** table.
- `s5lop` - Logical operator for the **SubCategory3** field in the **Product** table.
- `s6lop` - Logical operator for the **Department** field in the **Product** table.
- `s7lop` - Logical operator for the **Manufacturer** field in the **Product** table.
- `s8lop` - Logical operator for the **Description** field in the **Product** table.

Additional form parameters control how the search is conducted and how to order the results:

- `sop` - The logical search operator as `AND` or `OR`. If not specified, the logical search operator defaults to `AND`.
- `sf` - The sort field for the search results.
- `so` - The sort order for the search results.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=search.lasso` and should not be modified if the default product search capabilities are to be used. If a third party module is used to perform product searches, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Search` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform product searches, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Many modifications can be made to this form to suit the needs of the store and/or to give the customer more options. For example, select lists could be added to allow for the selection of search operators for some or all of the fields. Or, select lists could be populated with all of the categories, subcategories, etc so that the customer does not waste time typing in a category that does not exist. Examples of some search form modifications may be provided in articles in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Search List

/hostedstore/storefront/html/search_list.lasso

`HSP_ProductList` is an array of arrays. The number of arrays is determined by the number of products that met the search criteria. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`. Hyperlinks to the shopping cart display content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`.

Important: An add to cart hyperlink should only be used for products without attributes. Failure to follow this advice will result in improper functionality.

2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. The price for the item (may be the **Price**, **SalePrice**, or **MemberPrice** field from the **Product** table depending upon which applies) formatted according to currency formatting preferences specified in *Store Configuration*.
6. The **Attributes** field for the product from the **Product** table.
7. An array of additional fields for the product record that are not used in the default product detail content page. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **Category** field for the item from the **Product** table.
 2. The **SubCategory** field for the item from the **Product** table.
 3. The **SubCategory2** field for the item from the **Product** table.
 4. The **SubCategory3** field for the item from the **Product** table.
 5. The **Department** field for the item from the **Product** table.
 6. The **Manufacturer** field for the item from the **Product** table.
 7. The **OnHand** field for the item from the **Product** table.
 8. The **WebSellQty** field for the item from the **Product** table.
 9. The age of the product record (the number of days between the current date and date in the **DateCreated** field for the item from the **Product** table). This value can be compared against an arbitrary value to display a **New!** message or image.

For example, if products that are less than 30 days old are to be displayed with an image named `new.gif` from the `/hostedstore/images/` directory, then the following code could be used:

```
[ If: ($HSP_ProductList->Get:Loop_Count->Get:7->Get:9) < 30 ]
  
[/If]
```

Or, for shared SSL certificate stores:

```
[ If: ($HSP_ProductList->Get:Loop_Count->Get:7->Get:9) < 30 ]
  
[/If]
```

8. The **OutOfStockDisplay** field for the item from the **Product** table if the **OnHand** field from the **Product** table has a quantity less than or equal to zero. A value of `Y` indicates that the out of stock product should be displayed, but not sold.
9. The **OutOfStockMessage** field for the item from the **Product** table if the **OnHand** field from the **Product** table has a quantity less than or equal to zero. If the **OutOfStockMessage** field is empty,

then the message defaults to the message set with the `HS_OutOfStockMessage` preference variable in *Store Configuration*.

`HSP_SearchCriteria` is a string containing all of the search criteria for the current product search result set. Add this variable to any hyperlinks that should preserve the search criteria. Examples of this include hyperlinks for navigating the found set and the product detail page. The product detail page and cart page are search criteria "aware" so passing the variable in hyperlinks to those pages will allow for the creation of a hyperlink back.

`HSP_Search` is an array that contains information about the current search result set. The elements of the array are detailed below (the number corresponds to the position in the array):

- The number of records in the found set. This is equivalent to the value returned by the **Found_Count** LDML tag. Note that this value may not accurately reflect the actual number of products in `HSP_ProductList`. The discrepancy will occur if member only items are in the found set, the customer is not a member (or is a member and has not logged in), and member only items are not to be displayed due to preferences set in *Store Configuration*.
- The position of the first record shown in the found set. This is equivalent to the value returned by the **Shown_First** LDML tag.
- The position of the last record shown in the found set. This is equivalent to the value returned by the **Shown_Last** LDML tag.
- The value that should be passed as a form parameter named `skip` with `HSP_SearchCriteria` in a navigation hyperlink for the previous product search result set.
- The value that should be passed as a form parameter named `skip` with `HSP_SearchCriteria` in a navigation hyperlink for the next product search result set.

Additional Form Parameters

An optional form parameter, `CartAddQty`, can be added to the form or URL that adds items to the shopping cart. The quantity specified by the parameter value will be the quantity of the item to add to the shopping cart. This parameter can be used in a number of ways:

- **As a select list** - By using a select list, the customer will be able to choose a quantity to add to the shopping cart. The following example allows for the selection of between 1 and 25 of an item:

```
<select name="CartAddQty">
  [Loop:25]
  <option value="[Loop_Count]">
[/Loop]
</select>
```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=cart.lasso" method="post">
  <input type="hidden" name="add" value="[$HSP_ProductList->Get:Loop_Count-
>Get:1]">
  <select name="CartAddQty">
    [Loop:25]
    <option value="[Loop_Count]">
    [/Loop]
  </select>
  <input type="submit" value="Add to Cart">
</form>
```

- **As a text box** - By using a text box, the customer will be able to specify any quantity to add to the

shopping cart.

```
<input type="text" name="CartAddQty" size="5" maxlength="5">
```

For non-configurable items, the hyperlinks for adding an item to the shopping cart will need to be converted to forms to allow for the choice of a quantity:

```
<form action="[$HS_PathToStore]/HostedStore.LassoApp?-  
ResponseLassoApp=cart.lasso" method="post">  
  <input type="hidden" name="add" value="[$HSP_ProductList->Get:Loop_Count-  
>Get:1]">  
  <input type="text" name="CartAddQty" size="5" maxlength="5">  
  <input type="submit" value="Add to Cart">  
</form>
```

- **As a URL** - By using a URL, images or hyperlinked text can be used to add a fixed quantity of an item to the shopping cart. The following example adds 10 of a non-configurable item to the shopping cart:

```
<a href="[$HS_PathToStore]/HostedStore.LassoApp?-  
ResponseLassoApp=cart.lasso&add=[$HSP_ProductList->Get:Loop_Count-  
>Get:1]&AddCartQty=10">Buy 10 Now!</a>
```

Intermediate Search Page

/hostedstore/storefront/html/search_searchpage.lasso

This content page is returned when a link to `HostedStore.LassoApp?-ResponseLassoApp=searchpage.lasso` is used without a `page` parameter. If the `page` parameter is passed with a value, then the intermediate search page will be */hostedstore/storefront/html/search_page Value*. All of the parameter information below will apply to the intermediate search page specified by the `page` parameter.

This content page can be used to create a form, a list of static links, or other methods to link to the search script. Several parameters are available for use. Each parameter is detailed below:

- `s1` - The **ProductName** field in the **Product** table.
- `s2` - The **Category** field in the **Product** table.
- `s3` - The **SubCategory** field in the **Product** table.
- `s4` - The **SubCategory2** field in the **Product** table.
- `s5` - The **SubCategory3** field in the **Product** table.
- `s6` - The **Manufacturer** field in the **Product** table.
- `s7` - The **Department** field in the **Product** table.
- `s8` - The **Description** field in the **Product** table.
- `s9` - The **OnSale** field in the **Product** table.
- `s10` - Verifies that the submitted price is greater than the value in the **Price** field in the **Product** table.
- `s11` - Verifies that the submitted price is less than the value in the **Price** field in the **Product** table.
- `s12` - The **RestrictClass** field in the **Product** table.

A second set of parameters is provided to allow the search operator (*begins with, ends with, contains,* etc) for each field to be specified. The numbering system corresponds to the parameters that map to the fields with `op` appended. The default for search operators is *begins with*. Each parameter is detailed below:

- `s1op` - The search operator for the **ProductName** field in the **Product** table.
- `s2op` - The search operator for the **Category** field in the **Product** table.
- `s3op` - The search operator for the **SubCategory** field in the **Product** table.
- `s4op` - The search operator for the **SubCategory2** field in the **Product** table.
- `s5op` - The search operator for the **SubCategory3** field in the **Product** table.
- `s6op` - The search operator for the **Department** field in the **Product** table.
- `s7op` - The search operator for the **Manufacturer** field in the **Product** table.
- `s8op` - The search operator for the **Description** field in the **Product** table.
- `s9op` - The search operator for the **OnSale** field in the **Product** table.
- `s10op` - The search operator for the **Price** field in the **Product** table.
- `s11op` - The search operator for the **Price** field in the **Product** table.
- `s12op` - The search operator for the **RestrictClass** field in the **Product** table.

A third set of parameters is provided to allow for the creation of forms that use checkboxes instead of a text form element. The numbering system corresponds to the parameters that map to the fields with `cb` appended. Simply assign a value, `y` for instance, to denote that the field will have data submitted from checkboxes. The corresponding field in the **Product** table will then be searched with all values using a *contains* operator. Each parameter is detailed below:

- `s1cb` - Checkboxes for the **ProductName** field in the **Product** table.
- `s2cb` - Checkboxes for the **Category** field in the **Product** table.
- `s3cb` - Checkboxes for the **SubCategory** field in the **Product** table.
- `s4cb` - Checkboxes for the **SubCategory2** field in the **Product** table.
- `s5cb` - Checkboxes for the **SubCategory3** field in the **Product** table.
- `s6cb` - Checkboxes for the **Department** field in the **Product** table.

- `s7cb` - Checkboxes for the **Manufacturer** field in the **Product** table.
- `s8cb` - Checkboxes for the **Description** field in the **Product** table.

A fourth set of parameters is provided to allow for the creation of forms that allow multiple criteria to be specified in a text or textarea form element. The numbering system corresponds to the parameters that map to the fields with `d` appended. Assign the delimiter used to the parameter. For example, if a comma will be used to separate multiple criteria for the **Description** field, then the `s8d` parameter should be assigned a value of `,`. The corresponding field in the **Product** table will then be searched with all values using a `contains` operator. Each parameter is detailed below:

- `s1d` - Multiple criteria delimiter for the **ProductName** field in the **Product** table.
- `s2d` - Multiple criteria delimiter for the **Category** field in the **Product** table.
- `s3d` - Multiple criteria delimiter for the **SubCategory** field in the **Product** table.
- `s4d` - Multiple criteria delimiter for the **SubCategory2** field in the **Product** table.
- `s5d` - Multiple criteria delimiter for the **SubCategory3** field in the **Product** table.
- `s6d` - Multiple criteria delimiter for the **Department** field in the **Product** table.
- `s7d` - Multiple criteria delimiter for the **Manufacturer** field in the **Product** table.
- `s8d` - Multiple criteria delimiter for the **Description** field in the **Product** table.

A fifth set of parameters is provided to control the logical operator used for fields that use checkbox selections or multiple criteria. If all of the checked values or criteria must be true, then the parameter should have a value of `AND`. If any of the checked values or criteria can be true, then the parameter should have a value of `OR`. The numbering system corresponds to the parameters that map to the fields with `lop` appended. Each parameter is detailed below:

- `s1lop` - Logical operator for the **ProductName** field in the **Product** table.
- `s2lop` - Logical operator for the **Category** field in the **Product** table.
- `s3lop` - Logical operator for the **SubCategory** field in the **Product** table.
- `s4lop` - Logical operator for the **SubCategory2** field in the **Product** table.
- `s5lop` - Logical operator for the **SubCategory3** field in the **Product** table.
- `s6lop` - Logical operator for the **Department** field in the **Product** table.
- `s7lop` - Logical operator for the **Manufacturer** field in the **Product** table.
- `s8lop` - Logical operator for the **Description** field in the **Product** table.

Additional parameters control how the search is conducted and how to order the results:

- `sop` - The logical search operator as `AND` or `OR`. If not specified, the logical search operator defaults to `AND`.
- `sf` - The sort field for the search results.
- `so` - The sort order for the search results.

The form action parameter or link for this page is `HostedStore.LassoApp?-ResponseLassoApp=search.lasso` and should not be modified if the default search capabilities are to be used. If a third party module is used to perform product searches, then consult the module documentation to modify the form action parameter as needed.

If a form is used, the name parameter of the submit input element is `Submitted_Search` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform product searches, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Search SubCategories

/hostedstore/storefront/html/search_subcategories.lasso

This content page is used to generate a list of all subcategories in a product category. When a subcategory is selected, then a search for products matching the subcategory and parent category is performed. Access the page by creating a hyperlink to `HostedStore.LassoApp?-ResponseLassoApp=search.lasso` with a form parameter of `show` that has a value from the **Category** field in the **Product** table. This hyperlink may be in the guise of a form or as a standard hyperlink. An example of a place to use this capability is in the *Browse Products* menu. Details about the *Browse Products* menu can be found in the [Modifying HostedStore > Content Pages > Navigation Pages > Browse Products](#) section.

`HSP_Subcategories` is an array of arrays. The number of arrays is determined by the number of subcategories in the product category. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **SubCategory** field from the **Product** table.
2. The number of products in the subcategory and category that have a value of `Y` in the **WebSell** field in the **Product** table (the products have been flagged for publication in the store catalog).

How is this feature useful? If an unusually large product category is selected by the customer, this page can be used to help the customer further refine their search by selecting a subcategory.

`HSP_Subcategories` can also be used to provide a further breakdown of product categories. Suppose that a store is selling products with a category of *Computers*. There may be multiple subcategories (*Cases, Hard Drives, Keyboards, Monitors*, etc) that each have multiple manufacturers. Breaking down the *Computers* category into its subcategories and manufacturers may allow the customer to find the exact brand of monitor they want quicker, for example.

Example code of how to break out a large category into its subcategories and each subcategory's manufacturers may be provided in the support area of the HostedStore Web site (www.hostedstore.com/support/). To use the code, simply use it to replace the code in the default *search_subcategories.lasso* file.

Shipping Info

/hostedstore/storefront/html/shipping_info.lasso

`HSP_AddressBook` is an array of the names of address books that belong to the customer. Each customer automatically has an address book entry named `Billing` that contains the billing address for the customer.

The display of address books on the page occurs if the `HS_UseAddressBooks` preference variable set in *Store Configuration* has a value of `Y`.

This also contains a form for submitting the information necessary to modify the shipping information for an order. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - First name (*REQUIRED*)
- `f2` - Last name (*REQUIRED*)
- `f3` - Company
- `f4` - Address (*REQUIRED*)
- `f5` - Additional address line
- `f6` - City (*REQUIRED*)
- `f7` - State/Province (*REQUIRED*)
- `f8` - Postal Code (*REQUIRED*)
- `f9` - Country (*REQUIRED*)
- `f10` - Email address (*REQUIRED*)
- `f11` - Phone number (*REQUIRED*)
- `f12` - Fax number
- `f13` - Customer comments
- `f14` - The credit card type. (*REQUIRED* if `f21` has a value of `Credit Card`)
- `f15` - The credit card number. (*REQUIRED* if `f21` has a value of `Credit Card`)
- `f16` - The credit card verification code. (*REQUIRED* if the `HS_UseSecurityCode` preference variable is set to `Y` in *Store Configuration* and `f21` has a value of `Credit Card`).
- `f17` - The month of credit card expiration. (*REQUIRED* if `f21` has a value of `Credit Card`)
- `f18` - The year of credit card expiration. (*REQUIRED* if `f21` has a value of `Credit Card`)
- `f19` - The address book entry name (*REQUIRED* if the `HS_UseAddressBooks` preference variable is set to `Y` in *Store Configuration*).
- `f20` - The address book entry name that should be assigned to the address. (only meaningful if the `HS_UseAddressBooks` preference variable is set to `Y` in *Store Configuration*).
- `f21` - The payment method. (*REQUIRED*) Supported values are `Check`, `Check by Mail/Phone`, `Credit Card`, `PayPal`, and `Purchase Order`.
- `f22` - The checking account number. (*REQUIRED* if `f21` has a value of `Check`)
- `f23` - The checking account routing number. (*REQUIRED* if `f21` has a value of `Check`)
- `f24` - The check number. (*REQUIRED* if `f21` has a value of `Check`)
- `f25` - The purchase order number. (*REQUIRED* if `f21` has a value of `Purchase Order`)
- `PC` - The promotion code for the order. (only meaningful if the `HS_UsePromotions` preference variable is set to `Y` in *Store Configuration*).

The form action parameter for this content page is `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=checkout.lasso` and should not be modified if the default shipping and payment information capture capabilities are to be used. If a third party module is used to perform capture of shipping and payment information, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_ContinueToOrderReview` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform capture of shipping and payment information, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Modify Shipping Info

/hostedstore/storefront/html/shipping_modify.lasso

This content page contains a form for submitting the information necessary to modify the shipping information for an order. Like other HostedStore forms, the form input elements are named numerically and prepended with an **f**. Each form element is detailed below:

- **f1** - First name (*REQUIRED*)
- **f2** - Last name (*REQUIRED*)
- **f3** - Company
- **f4** - Address (*REQUIRED*)
- **f5** - Additional address line
- **f6** - City (*REQUIRED*)
- **f7** - State/Province (*REQUIRED*)
- **f8** - Postal Code (*REQUIRED*)
- **f9** - Country (*REQUIRED*)
- **f10** - Email address (*REQUIRED*)
- **f11** - Phone number (*REQUIRED*)
- **f12** - Fax number

The form action parameter for this content page is `[$HS_SecurePathToStore]/HostedStore.LassoApp?-ResponseLassoApp=modifyshipping.lasso` and should not be changed if the default shipping information update capabilities are to be used. If a third party module is used to perform shipping information updates, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Update` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform shipping information updates, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Store Closed

/hostedstore/storefront/html/store_closed.lasso

Content pages do not get any simpler than this one! This is the page that is displayed if the store is set to a closed status in *Store Configuration*.

The default page just has a text message with a hyperlink to email customer service using the `HS_CustomerServiceEmail` preference variable specified in *Store Configuration*.

Wish List Detail

/hostedstore/storefront/html/wishlist_detail.lasso

`HSP_Wishlist` is an array of arrays. The number of arrays is determined by the number of items in the wish list. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The date the item was added to the wishlist. This is the **DateCreated** field for the wish list from the **WishList** table formatted according to date formatting preferences specified in *Store Configuration*.
3. The **ProductName** field for the wish list item from the **WishList** table.
4. The names of product attributes if the item has attributes.
5. The chosen values for the product attributes if the item has attributes.
6. The value that should be passed as a form parameter named `add` to the wish list detail content page. This value should be coupled with the form parameter named `view` as detailed below.
7. The value that should be passed as a form parameter named `view` to the wish list detail content page. This value should be coupled with the form parameter named `add` as detailed above.

Links to the wish list detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=wishlist.lasso&add=VALUE&view=VALUE`

A simple hyperlink is provided on the page to allow the user to go back to the wish list search results content page if necessary: `javascript:window.history.go(-1);`

Wish List List

/hostedstore/storefront/html/wishlist_list.lasso

`HSP_Wishlist` is an array of arrays. The number of arrays is determined by the number of wish lists that match the search criteria submitted from the wish list search form (*/hostedstore/storefront/html/wishlist_searchform.lasso*). The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `view` to the wish list detail content page. Hyperlinks to the wish list detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=wishlist.lasso&view=VALUE`
2. First and last name of the wish list owner in the format: **LastName, FirstName** (fields from the **Customer** table)
3. The **City** field for the wish list owner from the **Customer** table.
4. The **State** field for the wish list owner from the **Customer** table.
5. The **Country** field for the wish list owner from the **Customer** table.

Wish List Login Form

/hostedstore/storefront/html/wishlist_loginform.lasso

This content page contains a form for submitting the information necessary to login to add items to a wish list. Login is necessary to ensure that the item is added to the correct wish list. Unlike other HostedStore forms, the form input elements are named alphabetically. Each form element is detailed below:

- **a** - The email address of the wish list owner. (*REQUIRED*)
- **b** - The password of the wish list owner. (*REQUIRED*)
- **c** - A checkbox that enables the wish list owner to have their forgotten password emailed to the email address given.

The checkbox can be eliminated from the form. However, it is a convenient feature for users that may have forgotten their password so consider carefully whether or not to remove it. Having the checkbox can also reduce the number of customer service requests regarding forgotten passwords.

Three hidden input elements are at the bottom of the form. These should not be modified or removed!

- **add** is the form parameter of the same name that specifies which product is to be added to the wish list.
- **options** is the number of attributes the product has.
- **[Loop_Count]** and the code surrounding the hidden form element specifies the attribute choices for the product.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=wishlist.lasso` and should not be modified if the default wish list login capabilities are to be used. If a third party module is used to perform wish list login, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Login` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform wish list login, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Wish List Search Form

/hostedstore/storefront/html/wishlist_searchform.lasso

This content page contains a form for submitting the information necessary to find publicly searchable wish lists. Like all HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each numerical form element is detailed below:

- `f1` - First name of the wish list owner.
- `f2` - Last name of the wish list owner.
- `f3` - Email address of the wish list owner.
- `f4` - City of the wish list owner.
- `f5` - State of the wish list owner.
- `f6` - Country of the wish list owner.

None of these form input elements are required, but at least one must be present on the form.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=wishlist.lasso` and should not be modified if the default wish list search capabilities are to be used. If a third party module is used to perform wish list searches, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Search` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform wish list searches, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Customer Administration

Each of the content pages located in the */hostedstore/admincustomer/html/* directory are detailed below.

Sections

- [Address Book Entry](#)
- [Address Book List](#)
- [Modify Billing Address](#)
- [EAD Files List](#)
- [Login Form](#)
- [Main Menu](#)
- [Newsletter Subscription](#)
- [Order Detail](#)
- [Order List](#)
- [Change Password](#)
- [Shopping List Detail](#)
- [Shopping List List](#)
- [Email Wish List](#)
- [Wish List Menu](#)
- [Wish List Preferences](#)
- [Wish List Purchases](#)
- [Wish List Items](#)

Address Book Entry

/hostedstore/admincustomer/html/address.lasso

This content page contains a form for submitting the information necessary to add or modify an address book entry. Each form element is detailed below:

- **EntryName** - The name to associate with the entry. (*REQUIRED*)
- **f1** - First name (*REQUIRED*)
- **f2** - Last name (*REQUIRED*)
- **f3** - Company
- **f4** - Address (*REQUIRED*)
- **f5** - Additional address line
- **f6** - City (*REQUIRED*)
- **f7** - State/Province (*REQUIRED*)
- **f8** - Postal Code (*REQUIRED*)
- **f9** - Country (*REQUIRED*)
- **f10** - Email address (*REQUIRED*)
- **f11** - Phone number
- **f12** - Fax number

The form may also contain a hidden form parameter **ID** that contains the value from a variable of the same name. This form parameter identifies the address book and should not be modified or removed.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/address.lasso` and should not be modified if the default address book function capabilities are to be used. If a third party module is used to perform address book functions, then consult the module documentation to modify the form action parameter as needed.

The name parameters of the submit input elements are `Submitted_Update`, `Submitted_Delete`, `Submitted_Add` for update, delete, and add actions, respectively. The name parameters should not be changed. The value parameters can be changed to suit the needs of the store. If a third party module is used to perform address book functions, then consult the module documentation to modify the name and value parameter of the submit input elements as needed.

Address Book List

/hostedstore/admincustomer/html/addressbook.lasso

`HSP_AddressBook` is an array of arrays. The number of arrays is determined by the number of address books for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the address book entry detail content page. Hyperlinks to the address book entry detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/address.lasso&ID=VALUE`
2. The **EntryName** field for the address book entry from the **Customer** table.
3. The **FirstName** field for the address book entry from the **Customer** table.
4. The **LastName** field for the address book entry from the **Customer** table.
5. The **Company** field for the address book entry from the **Customer** table.
6. The **Address1** field for the address book entry from the **Customer** table.
7. The **Address2** field for the address book entry from the **Customer** table.
8. The **City** field for the address book entry from the **Customer** table.
9. The **State** field for the address book entry from the **Customer** table.
10. The **Postal** field for the address book entry from the **Customer** table.
11. The **Country** field for the address book entry from the **Customer** table.
12. The **Phone** field for the address book entry from the **Customer** table.
13. The **Fax** field for the address book entry from the **Customer** table.
14. The **Email** field for the address book entry from the **Customer** table.

To add a new address book entry, pass a form parameter named `do` with a value of `add` to the add address book entry content page. Hyperlinks to the add address book entry content page should use:

`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/address.lasso&do=add`

Modify Billing Address

/hostedstore/admincustomer/html/billingaddress.lasso

This content page contains a form for submitting the information necessary to add or modify an address book entry. Each form element is detailed below:

- **f1** - First name (*REQUIRED*)
- **f2** - Last name (*REQUIRED*)
- **f3** - Company
- **f4** - Address (*REQUIRED*)
- **f5** - Additional address line
- **f6** - City (*REQUIRED*)
- **f7** - State/Province (*REQUIRED*)
- **f8** - Postal Code (*REQUIRED*)
- **f9** - Country (*REQUIRED*)
- **f10** - Email address (*REQUIRED*)
- **f11** - Phone number
- **f12** - Fax number

The form may also contain a hidden form parameter **ID** that contains the value from a variable of the same name. This form parameter identifies the address book and should not be modified or removed.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/address.lasso` and should not be modified if the default address book function capabilities are to be used. If a third party module is used to perform address book functions, then consult the module documentation to modify the form action parameter as needed.

The name parameters of the submit input elements are `Submitted_Update`, `Submitted_Delete`, `Submitted_Add` for update, delete, and add actions, respectively. The name parameters should not be changed. The value parameters can be changed to suit the needs of the store. If a third party module is used to perform address book functions, then consult the module documentation to modify the name and value parameter of the submit input elements as needed.

EAD Files List

/hostedstore/admincustomer/html/files_list.lasso

`HSP_FilesList` is an array of arrays. The number of arrays is determined by the number of downloadable files purchased by the customer. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `download` to the download page. Hyperlinks to the download page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=admincustomer/files.lasso&download=VALUE`
2. The **EADFileNames** field for the file from the **Item** table.
3. The date portion of the **DateShipped** field from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
4. The expiration date for the download link.
5. The **EADDownloadLimit** field for the file from the **Item** table.
6. The **EADTimesDownloaded** field for the file from the **Item** table.
7. A value of `Y` if the download link has expired due to reaching the maximum downloads threshold or by passing the expiration date for download.

Note that when downloading EAD files, the customer must use the **Save As** capability of the browser in order to have it function properly. A note about this is included on the default EAD files list content page.

Login Form

/hostedstore/admincustomer/html/loginform.lasso

This content page contains a form for submitting the information necessary to login to *Customer Account Administration*. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - The email address of the customer. (*REQUIRED*)
- `f2` - The password of the customer. (*REQUIRED*)
- `f3` - A checkbox that enables the customer to have their forgotten password emailed to the email address given.

The checkbox can be eliminated from the form. However, it is a convenient feature for customers that may have forgotten their password so consider carefully whether or not to remove it. Having the checkbox can also reduce the number of customer service requests regarding forgotten passwords.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/index.lasso` and should not be modified if the default customer login capabilities are to be used. If a third party module is used to perform customer login, then consult the module documentation to modify the form action parameter as needed.

The name parameter of the submit input element is `Submitted_Login` and should not be changed. The value parameter can be changed to suit the needs of the store. If a third party module is used to perform customer login, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Main Menu

/hostedstore/admincustomer/html/menu.lasso

This content page contains a list of hard coded hyperlinks for navigating *Customer Account Administration*. Each hyperlink is provided below just in case one is deleted during modification:

- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&choose=password to modify the customer account password.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&choose=newsletter to subscribe/unsubscribe to the newsletter.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/address.lasso to access the address book administration menu.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&choose=billing to modify customer billing information.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&choose=orders to access the order administration menu.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/wishlist.lasso to access the wishlist administration menu.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/shoppinglist.lasso to access the shopping list administration menu.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&choose=files to access the purchased files administration menu.

Newsletter Subscription

/hostedstore/admincustomer/html/newsletter.lasso

This content page has a variable named `f2` which contains the current value of the customer's newsletter subscription status.

This content page also contains a form for submitting the information necessary to subscribe to or unsubscribe from the newsletter. There is one hidden form parameter named `f1` that contains a variable of the same name.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/admin.lasso` and should not be modified if the default newsletter handling capabilities are to be used. If a third party module is used to perform newsletter handling, then consult the module documentation to modify the form action parameter as needed.

The name parameters of the submit input elements are `Submitted_Subscribe` and `Submitted_Unsubscribe`. The name parameters of the submit input elements should not be changed. The value parameters can be changed to suit the needs of the store. If a third party module is used to perform newsletter handling, then consult the module documentation to modify the name and value parameter of the submit input element as needed.

Order Detail

/hostedstore/admincustomer/html/order_detail.lasso

`HSP_Order` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `view` to the create shopping list content page.
2. An array that contains the action dates for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **DateCanceled** field for the order from the **Invoice** table.
 2. The **DateShipped** field for the order from the **Invoice** table.
 3. The **DateBackOrdered** field for the order from the **Invoice** table.
 4. The **DateProcessed** field for the order from the **Invoice** table.
 5. The **DateApproved** field for the order from the **Invoice** table.
 6. The **DateCanceledPending** field for the order from the **Invoice** table.
 7. The **DatePosted** field for the order from the **Invoice** table.
3. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
4. An array that contains the global shipping vendor information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ShippingVendor** field for the order from the **Invoice** table.
 2. The **TrackingNumber** field for the order from the **Invoice** table.
5. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
 2. The **ProductName** field for the item from the **Item** table.
 3. Has a value of `y` if the item was purchased from a wishlist.
 4. An array that contains the names of the item attributes if any exist.
 5. An array that contains the chosen value for item attributes if any exist.
 6. The **Quantity** field for the item from the **Item** table.
 7. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **DateShipped** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
 10. The **ShippingVendor** field for the item from the **Item** table.
 11. The **TrackingNumber** field for the item from the **Item** table.
 12. The **DateBackOrdered** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
 13. The **DateCanceled** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
6. The **ItemCount** field for the order from the **Invoice** table.
7. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting

preferences specified in *Store Configuration*.

8. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.

The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*. This content page contains a form for submitting the information necessary to cancel an order. Each form element is detailed below:

- o **f1** - A checkbox to indicate the order should be canceled. (*REQUIRED*)
- o **f2** - The reason for cancelation. (*REQUIRED*)

The form also has a hidden form parameter `view` that contains the value from a variable of the same name. This form parameter identifies the order and should not be modified or removed.

This content page also contains a form for submitting the information necessary to create a shopping list from the order. Each form element is detailed below:

- o **f3** - The name of the shopping list. (*REQUIRED*)

The form also has a hidden form parameter `view` that contains the value from a variable of the same name. This form parameter identifies the order and should not be modified or removed.

The form action parameter for the cancel order form is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/cancelorder.lasso` and should not be modified if the default order cancelation capabilities are to be used. The form action parameter for the create shopping list form is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/shoppinglist.lasso` and should not be modified if the default shopping list creation capabilities are to be used.

The name parameter of the two submit input elements should not be changed, but the value parameter can be changed to suit the needs of the store.

Order List

/hostedstore/admincustomer/html/order_list.lasso

`HSP_OrderList` is an array of arrays. The number of arrays is determined by the number of orders for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `view` to the order detail content page. Hyperlinks to the order detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/admin.lasso&view=VALUE`
2. The **DateCreated** field for the order from the **Invoice** table formatted according to date formatting preferences specified in *Store Configuration*.
3. The **ItemCount** field for the order from the **Invoice** table.
4. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. The **Status** field for the order from the **Invoice** table.

Change Password

/hostedstore/admincustomer/html/password.lasso

This content page contains a form for submitting the information necessary to change the password used to login into *Customer Account Administration*. Like other HostedStore forms, the form input elements are named numerically and prepended with an **f**. Each form element is detailed below:

- **f1** - The password. (*REQUIRED*)
- **f2** - Verification of the password. (*REQUIRED*)

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/admin.lasso` and should not be modified if the default customer account administration login capabilities are to be used.

The name parameter of the submit input element should not be changed, but the value parameter can be changed to suit the needs of the store.

Shopping List Detail

/hostedstore/admincustomer/html/shoppinglist_detail.lasso

`HSP_ShoppingList` is an array of arrays. The number of arrays is determined by the number of items in the shopping list. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ListName** field for the item from the **ShoppingList** table.
2. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
3. The **ProductName** field for the item from the **ShoppingList** table.
4. The **Quantity** field for the item from the **ShoppingList** table.
5. The value to be used for the `name` attribute of the form parameter used to display and modify the quantity of the item.
6. An array that contains the names of the item attributes if any exist.
7. An array that contains the chosen value for item attributes if any exist.

This content page contains a form for submitting the information necessary to add the shopping list to the shopping cart, update the quantities of items in the shopping list, and to delete the shopping list. The form should have text form elements with a `name` of the fifth array element and `value` of the fourth array element.

A hidden form parameter named `ID` has the value for a variable of the same name. This form parameter should not be removed or modified.

The form action parameter for this page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/shoppinglist.lasso` and should not be modified if the default shopping list function capabilities are to be used.

The name parameter of the submit input elements should not be changed, but the value parameters can be changed to suit the needs of the store.

Shopping List List

/hostedstore/admincustomer/html/shoppinglist_list.lasso

`HSP_ShoppingLists` is an array of arrays. The number of arrays is determined by the number of shopping lists for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `view` to the shopping list detail content page. Hyperlinks to the shopping list detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/shoppinglist.lasso&view=VALUE`
2. The **ListName** field for the shopping list from the **ShoppingList** table.
3. The **DateCreated** field for the shopping list from the **ShoppingList** table formatted according to date formatting preferences specified in *Store Configuration*.
4. The **DateModified** field for the shopping list from the **ShoppingList** table formatted according to date formatting preferences specified in *Store Configuration*.

Email Wish List

/hostedstore/admincustomer/html/wishlist_email.lasso

This content page contains a form for submitting the information necessary to login to email a hyperlink to the customer's wish list. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - The subject of the email. (*REQUIRED*)
- `f2` - The email address of the recipient. (*REQUIRED*)
- `f3` - A personal message.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/wishlist.lasso` and should not be modified if the default wish list email capabilities are to be used.

The name parameter of the submit input element should not be changed, but the value parameter can be changed to suit the needs of the store.

Wish List Menu

/hostedstore/admincustomer/html/wishlist_menu.lasso

This content page contains a list of hard coded hyperlinks for navigating the wish list options. Each hyperlink is provided below just in case one is deleted during modification:

- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/wishlist.lasso&choose=preferences to modify public search status.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/wishlist.lasso&choose=email to email a a direct hyperlink to the wish list.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/wishlist.lasso&choose=view to view items currently in the wish list, remove items from the wish list, and add wish list items to the shopping cart.
- [`$_HS_PathToStore`]/HostedStore.LassoApp?-
ResponseLassoApp=admincustomer/wishlist.lasso&choose=purchases to view wish list items that have been purchased.

Wish List Preferences

/hostedstore/admincustomer/html/wishlist_preferences.lasso

This content page contains a variable named `f1` which holds the current value of the customer's wish list public search preference.

This content page also contains a form for submitting the information necessary to modify the preference. There is one hidden form parameter named `f1` that contains a variable of the same name.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/wishlist.lasso` and should not be modified if the default wish list function capabilities are to be used.

The name parameter of the submit input element should not be changed, but the value parameter can be changed to suit the needs of the store.

Wish List Purchases

/hostedstore/admincustomer/html/wishlist_purchases.lasso

`HSP_WishList` is an array of arrays. The number of arrays is determined by the number of items that have been purchased from the customer's wish list. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **DateModified** field for the item formatted according to date formatting preferences specified in *Store Configuration*.
3. The **ProductName** field for the item from the **WishList** table.
4. An array that contains the names of the item attributes if any exist.
5. An array that contains the chosen value for item attributes if any exist.

Wish List Items

/hostedstore/admindcustomer/html/wishlist_view.lasso

`HSP_WishList` is an array of arrays. The number of arrays is determined by the number of items that are in the customer's wish list. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
2. The **DateModified** field for the item from the **WishList** table formatted according to date formatting preferences specified in *Store Configuration*.
3. The **ProductName** field for the item from the **WishList** table.
4. An array that contains the names of the item attributes if any exist.
5. An array that contains the chosen value for item attributes if any exist.
6. The value that should be passed as a form parameter named `add` or `remove` to the wish list processing script. Hyperlinks to the wish list processing script should use the following hyperlink to add the item to the shopping cart: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=admincustomer/wishlist.lasso&add=VALUE`
Hyperlinks to the wish list processing script should use the following hyperlink to remove the item from the wish list: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=admincustomer/wishlist.lasso&remove=VALUE`

Store Administration

At this time, most of the content pages in *Store Administration* are not available for modification and, honestly, that is not expected to change in the immediate future. If different features are needed in *Store Administration* then a modification request for a future version can be made. Or, alternatively, a third party module that addresses the need may exist or a module may be created. Modification requests can be made in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Sections

- [Module Menu](#)
- [Mailing Invoice](#)
- [Packing Invoice](#)
- [Pick List](#)

Module Menu

/hostedstore/adminstore/html/module_menu.lasso

Modules that require administration can be listed on this page. Consult the documentation provided with the module for instructions on what to place on this page.

Mailing Invoice

/hostedstore/adminstore/html/order_mailinginvoice.lasso

`HSP_Order` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. *Not used for this page.*
2. The **ID** field from the **Invoice** table.
3. The **OrderID** field from the **Invoice** table.
4. The **DateCreated** field from the **Invoice** table formatted according to date and time formatting preferences specified in *Store Configuration*.
5. The **Status** field from the **Invoice** table.
6. A summary of the dates for each status change (posted, approved, back ordered, shipped, etc).
7. An array that contains the billing address information for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Customer** table.
 2. The **Company** field from the **Customer** table.
 3. The **Address1** field from the **Customer** table.
 4. The **Address2** field from the **Customer** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Customer** table.
 6. The **Country** field from the **Customer** table.
 7. The **Email** field from the **Customer** table.
 8. The **Phone** field from the **Customer** table.
 9. The **Fax** field from the **Customer** table.
8. An array that contains the shipping address information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Invoice** table.
 2. The **Company** field from the **Invoice** table.
 3. The **Address1** field from the **Invoice** table.
 4. The **Address2** field from the **Invoice** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Invoice** table.
 6. The **Country** field from the **Invoice** table.
 7. The **Email** field from the **Invoice** table.
 8. The **Phone** field from the **Invoice** table.
 9. The **Fax** field from the **Invoice** table.
9. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field from the **Invoice** table.
 2. The **Account** field from the **Invoice** table.
 3. The **Expiration** field from the **Invoice** table.
 4. The **Routing** field from the **Invoice** table.
 5. The **CheckNumber** field from the **Invoice** table.
 6. The **PONumber** field from the **Invoice** table.
 7. The **TransactionID** field from the **Invoice** table.
 8. The **CustomerComments** field from the **Invoice** table.
 9. The **Source** field from the **Invoice** table.
10. The **AdminComments** field from the **Invoice** table.
11. An array that contains the totals for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 2. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 4. The **PromotionCode** field.
 5. The **Discount** field for the order from the **Invoice** table formatted according to currency

- formatting preferences specified in *Store Configuration*.
6. The **SubTotalAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **ShippingAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TaxAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **DiscountAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
12. *Not used for this page.*
 13. The **ShippingVendor** field from the **Invoice** table.
 14. The **TrackingNumber** field from the **Invoice** table.
 15. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. *Not used for this page.*
 2. The **OrderStatus** field from the **Item** table.
 3. *Not used for this page.*
 4. The **ProductID** field from the **Item** table.
 5. The **ProductName** field from the **Item** table.
 6. The **WishListOwnerID** field from the **Item** table.
 7. The **Quantity** field from the **Item** table.
 8. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **ShippingVendor** field from the **Item** table.
 11. The **TrackingNumber** field from the **Item** table.
 12. The **DateShipped** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 13. The **DateBackOrdered** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 14. The **DateCanceled** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 15. An array that contains the names and chosen values of the item attributes if any exist formatted in the form **ATTRIBUTE NAME ATTRIBUTE VALUE**.
 16. An array of arrays that contains the part quantities and part product IDs (if any) associated with the attributes for ordered items. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. Array of part quantities arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.
 2. Array of part product ID arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.

Packing Invoice

/hostedstore/adminstore/html/order_packinginvoice.lasso

HSP_Order is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. *Not used for this page.*
2. The **ID** field from the **Invoice** table.
3. The **OrderID** field from the **Invoice** table.
4. The **DateCreated** field from the **Invoice** table formatted according to date and time formatting preferences specified in *Store Configuration*.
5. The **Status** field from the **Invoice** table.
6. A summary of the dates for each status change (posted, approved, back ordered, shipped, etc).
7. An array that contains the billing address information for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Customer** table.
 2. The **Company** field from the **Customer** table.
 3. The **Address1** field from the **Customer** table.
 4. The **Address2** field from the **Customer** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Customer** table.
 6. The **Country** field from the **Customer** table.
 7. The **Email** field from the **Customer** table.
 8. The **Phone** field from the **Customer** table.
 9. The **Fax** field from the **Customer** table.
8. An array that contains the shipping address information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Invoice** table.
 2. The **Company** field from the **Invoice** table.
 3. The **Address1** field from the **Invoice** table.
 4. The **Address2** field from the **Invoice** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Invoice** table.
 6. The **Country** field from the **Invoice** table.
 7. The **Email** field from the **Invoice** table.
 8. The **Phone** field from the **Invoice** table.
 9. The **Fax** field from the **Invoice** table.
9. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field from the **Invoice** table.
 2. The **Account** field from the **Invoice** table.
 3. The **Expiration** field from the **Invoice** table.
 4. The **Routing** field from the **Invoice** table.
 5. The **CheckNumber** field from the **Invoice** table.
 6. The **PONumber** field from the **Invoice** table.
 7. The **TransactionID** field from the **Invoice** table.
 8. The **CustomerComments** field from the **Invoice** table.
 9. The **Source** field from the **Invoice** table.
10. The **AdminComments** field from the **Invoice** table.
11. An array that contains the totals for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 2. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 4. The **PromotionCode** field.
 5. The **Discount** field for the order from the **Invoice** table formatted according to currency

- formatting preferences specified in *Store Configuration*.
6. The **SubTotalAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **ShippingAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TaxAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **DiscountAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
12. *Not used for this page.*
 13. The **ShippingVendor** field from the **Invoice** table.
 14. The **TrackingNumber** field from the **Invoice** table.
 15. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. *Not used for this page.*
 2. The **OrderStatus** field from the **Item** table.
 3. *Not used for this page.*
 4. The **ProductID** field from the **Item** table.
 5. The **ProductName** field from the **Item** table.
 6. The **WishListOwnerID** field from the **Item** table.
 7. The **Quantity** field from the **Item** table.
 8. The **Price** field from the **Item** table for the item formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **ShippingVendor** field from the **Item** table.
 11. The **TrackingNumber** field from the **Item** table.
 12. The **DateShipped** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 13. The **DateBackOrdered** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 14. The **DateCanceled** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 15. An array that contains the names and chosen values of the item attributes if any exist formatted in the form **ATTRIBUTE NAME ATTRIBUTE NAME**.
 16. An array of arrays that contains the part quantities and part product IDs (if any) associated with the attributes for ordered items. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. Array of part quantities arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.
 2. Array of part product ID arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.

Pick List

`/hostedstore/adminstore/html/order_picklist.lasso`

`HSP_Order` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. *Not used for this page.*
2. The **ID** field from the **Invoice** table.
3. The **OrderID** field from the **Invoice** table.
4. The **DateCreated** field from the **Invoice** table formatted according to date and time formatting preferences specified in *Store Configuration*.
5. The **Status** field from the **Invoice** table.
6. A summary of the dates for each status change (posted, approved, back ordered, shipped, etc).
7. An array that contains the billing address information for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Customer** table.
 2. The **Company** field from the **Customer** table.
 3. The **Address1** field from the **Customer** table.
 4. The **Address2** field from the **Customer** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Customer** table.
 6. The **Country** field from the **Customer** table.
 7. The **Email** field from the **Customer** table.
 8. The **Phone** field from the **Customer** table.
 9. The **Fax** field from the **Customer** table.
8. An array that contains the shipping address information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The concatenated **FirstName** and **LastName** fields from the **Invoice** table.
 2. The **Company** field from the **Invoice** table.
 3. The **Address1** field from the **Invoice** table.
 4. The **Address2** field from the **Invoice** table.
 5. The concatenated **City**, **State**, and **Postal** fields from the **Invoice** table.
 6. The **Country** field from the **Invoice** table.
 7. The **Email** field from the **Invoice** table.
 8. The **Phone** field from the **Invoice** table.
 9. The **Fax** field from the **Invoice** table.
9. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field from the **Invoice** table.
 2. The **Account** field from the **Invoice** table.
 3. The **Expiration** field from the **Invoice** table.
 4. The **Routing** field from the **Invoice** table.
 5. The **CheckNumber** field from the **Invoice** table.
 6. The **PONumber** field from the **Invoice** table.
 7. The **TransactionID** field from the **Invoice** table.
 8. The **CustomerComments** field from the **Invoice** table.
 9. The **Source** field from the **Invoice** table.
10. The **AdminComments** field.
11. An array that contains the totals for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 2. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 4. The **PromotionCode** field.
 5. The **Discount** field for the order from the **Invoice** table formatted according to currency

- formatting preferences specified in *Store Configuration*.
6. The **SubTotalAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **ShippingAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TaxAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **DiscountAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
12. *Not used for this page.*
 13. The **ShippingVendor** field from the **Invoice** table.
 14. The **TrackingNumber** field from the **Invoice** table.
 15. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. *Not used for this page.*
 2. The **OrderStatus** field from the **Item** table.
 3. *Not used for this page.*
 4. The **ProductID** field from the **Item** table.
 5. The **ProductName** field from the **Item** table.
 6. The **WishListOwnerID** field from the **Item** table.
 7. The **Quantity** field from the **Item** table.
 8. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **ShippingVendor** field from the **Item** table.
 11. The **TrackingNumber** field from the **Item** table.
 12. The **DateShipped** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 13. The **DateBackOrdered** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 14. The **DateCanceled** field from the **Item** table formatted according to date and time formatting preferences specified in *Store Configuration*.
 15. An array that contains the names and chosen values of the item attributes if any exist formatted in the form **ATTRIBUTE NAME ATTRIBUTE NAME**.
 16. An array of arrays that contains the part quantities and part product IDs (if any) associated with the attributes for ordered items. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. Array of part quantities arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.
 2. Array of part product ID arrays, where each element is an attribute. The order matches the order of the attributes accessed with `$HSP_Order->Get:15->Get:15`.

Affiliate Administration

Each of the content pages located in the */hostedstore/affiliate/html/* directory are detailed below.

Sections

- [Login Form](#)
- [Orders List](#)

Login Form

/hostedstore/affiliate/html/loginform.lasso

This content page contains a form for submitting the information necessary to login into *Affiliate Account Administration*. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - The email address of the affiliate. (*REQUIRED*)
- `f2` - The password of the affiliate. (*REQUIRED*)
- `f3` - A checkbox that enables the affiliate to have their forgotten password emailed to the email address given.

The checkbox can be eliminated from the form. However, it is a convenient feature for affiliates that may have forgotten their password so consider carefully whether or not to remove it. Having the checkbox can also reduce the number of customer service requests regarding forgotten passwords.

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=affiliate/index.lasso` and should not be modified if the default affiliate login capabilities are to be used.

The name parameter of the submit input element should not be changed, but the value parameter can be changed to suit the needs of the store.

Orders List

/hostedstore/affiliate/html/orders_list.lasso

HSP_AffiliateOrders is an array of arrays. The number of arrays is determined by the number of orders attributed to the affiliate. Orders are attributed to an affiliate when the affiliate's ID is associated with the order and the order has a status of *Shipped*. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **DateShipped** field for the order from the **Invoice** table formatted according to date formatting preferences specified in *Store Configuration*.
2. The **DateCreated** field for the order from the **Invoice** table formatted according to date formatting preferences specified in *Store Configuration*.
3. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
4. The affiliate payout due for the order formatted according to currency formatting preferences specified in *Store Configuration*.
5. The date portion of the **AffiliatePayout** field for the order from the **Invoice** table formatted according to date formatting preferences specified in *Store Configuration*.
6. The check number portion of the **AffiliatePayout** field for the order from the **Invoice** table.
7. The check monetary value portion of the **AffiliatePayout** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.

HSP_AffiliatePlan is a string that states what payout in percent or value the affiliate receives for each order.

HSP_HistoricSubtotal is the cumulative subtotal of all orders attributed to the affiliate formatted according to currency formatting preferences specified in *Store Configuration*.

HSP_HistoricPayout is the cumulative payout of all orders attributed to the affiliate formatted according to currency formatting preferences specified in *Store Configuration*.

NEED TO UPDATE WITH NEW VARIABLES.

Supplier Administration

A template can be made for each supplier (or shared among a group of suppliers).

Sections

- [Template](#)

Template

Supplier Account Administration templates are created by determining where the content pages will be inserted into the template and taking the upper half of the HTML document and saving it with the name *top.lasso* and saving the lower half of the template as *bottom.lasso*. The template files should then be placed in their own directory in the `/hostedstore/adminsupplier/templates/` directory. The name of the directory is the name of the template to assign to the supplier in *Supplier Administration*. Details about assigning a template to a supplier can be found in the [Store Administration](#) > [Supplier Administration](#) > section.

The default template is named *default* and can be found in the `/hostedstore/adminsupplier/templates/` directory. The default template is very basic, but may provide some insight into how to split a template into two files (*top.lasso* and *bottom.lasso*).

Help Center

Each of the content pages located in the */hostedstore/helpcenter/html/* directory are detailed below.

Sections

- [Customer Service Inquiry Form](#)
- [Help Center Menu](#)
- [Other Pages](#)

Customer Service Inquiry Form

`/hostedstore/helpcenter/html/customerserviceform.lasso`

This content page contains a form for submitting the information necessary for a customer service inquiry. Like other HostedStore forms, the form input elements are named numerically and prepended with an `f`. Each form element is detailed below:

- `f1` - The customer's name. (*REQUIRED*)
- `f2` - The customer's email address. (*REQUIRED*)
- `f3` - The subject of the inquiry. (*REQUIRED*)
- `f4` - The message. (*REQUIRED*)

The form action parameter for this content page is `HostedStore.LassoApp?-ResponseLassoApp=helpcenter/customerservice.lasso` and should not be modified if the default customer service inquiry capabilities are to be used.

The name parameter of the submit input element, `Submitted_SubmitInquiry`, should not be changed, but the value parameter can be changed to suit the needs of the store.

Help Center Menu

/hostedstore/helpcenter/html/menu.lasso

This is the default page that is loaded when a hyperlink to the *Help Center* is clicked of the form `HostedStore.LassoApp?-ResponseLassoApp=helpcenter.lasso`.

The hyperlink to the customer service inquiry form is of the form `HostedStore.LassoApp?-ResponseLassoApp=helpcenter/customerservice.lasso`.

All other hyperlinks to *Help Center* content pages are of the form

`[$HS_PathToStore]/HostedStore.LassoApp?-`

`ResponseLassoApp=helpcenter.lasso&show=PAGE NAME HERE`, where *PAGE NAME HERE* is the name of the format file (without the .lasso extension) to load from the */hostedstore/helpcenter/html/* directory. For example, a format file named *securitypolicy.lasso* would have a corresponding hyperlink of

`[$HS_PathToStore]/HostedStore.LassoApp?-`

`ResponseLassoApp=helpcenter.lasso&show=securitypolicy`.

Other Pages

Several pages are provided in the *Help Center* as suggestions of the type of content that may be provided for customers. Each of the pages has a corresponding hyperlink on the `/hostedstore/helpcenter/html/menu.lasso` page. If any of the content pages are not wanted, be sure to remove the appropriate hyperlink from the menu. Most of the content pages do not have any content because the policies, descriptions, etc will vary from store to store. Each content page is listed below along with a suggestion of the type of content it could contain:

- *about.lasso* - History of the company, the store, etc.
- *browsersupport.lasso* - What browsers and browser versions are supported and which browsers may not work as expected.
- *cancelorder.lasso* - Policy and procedure for canceling orders.
- *contactus.lasso* - Mailing address and email addresses.
- *errors.lasso* - Errors and omissions policy.
- *faq.lasso* - A list of frequently asked questions.
- *firsttime.lasso* - Guidance for first time customers on how to shop successfully in the store.
- *manageaccount.lasso* - Procedure for customers to manage their account, options available, etc.
- *orderproblems.lasso* - Procedure for what to do if a customer has a problem placing an order or problems with an order that has already been placed.
- *orderstatus.lasso* - How to check the status of an order.
- *ordertracking.lasso* - How to track a shipped order.
- *paymentoptions.lasso* - Description of payment methods and policies.
- *placingorders.lasso* - Steps for placing an order in the store.
- *preorderpolicy.lasso* - Policy on preordering out of stock items.
- *pricing.lasso* - Pricing and availability policy.
- *privacypolicy.lasso* - Store privacy policy.
- *promotions.lasso* - Types of promotions, how to use promotion codes, how to obtain promotion codes, how promotion discounts are calculated, etc.
- *returnpolicy.lasso* - Policy for returning shipped items.
- *review.lasso* - Policy for reviewing items in the store.
- *salestax.lasso* - How sales tax is calculated, what jurisdictions are subject to taxation, etc.
- *searchtips.lasso* - Guidance for how to successfully use the product search form to efficiently find items.
- *securitypolicy.lasso* - Store security policy.
- *shipping.lasso* - How shipping is calculated, shipping options available, etc.

Email Messages

All of the email messages sent by HostedStore can be modified. Each default email format file is detailed below. Note that the main intent is to document the information that is available to each default email format file. In general, there will not be a discussion of the actual code and code structure.

Sections

- [Storefront](#)
- [Customer Administration](#)
- [Store Administration](#)
- [Affiliate Administration](#)
- [Supplier Administration](#)
- [Help Center](#)

Storefront

Each of the email files located in the */hostedstore/storefront/email/* directory are detailed below.

Sections

- [Customer Order Approval](#)
- [Email a Friend](#)
- [Sales Membership Purchase](#)
- [Customer New Account](#)
- [Forgotten Password](#)
- [Shopping Cart Quote](#)
- [Sales Order Approval](#)
- [Newsletter Subscription](#)

Customer Order Approval

/hostedstore/storefront/email/email_customerapproval.lasso

HSP_Order is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ID** field for the order from the **Invoice** table.
2. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table.
 3. The **Expiration** field for the order from the **Invoice** table.
 4. The **PONumber** field for the order from the **Invoice** table.
3. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
4. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ProductName** field for the item from the **Item** table.
 2. Has a value of **y** if the item was purchased from a wishlist.
 3. An array that contains the names of the item attributes if any exist.
 4. An array that contains the chosen value for item attributes if any exist.
 5. The **Quantity** field for the item from the **Item** table.
 6. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. The **ItemCount** field for the order from the **Invoice** table.
6. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
7. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
8. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
9. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.

Email a Friend

/hostedstore/storefront/email/email_friend.lasso

`HSP_Email` is an array that has information about the product that is to be emailed. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The value that should be passed as a form parameter named `ID` to the product detail content page or as a form parameter named `add` to the shopping cart display content page. Hyperlinks to the product detail content page should use: `[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
Hyperlinks to the shopping cart display content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=cart.lasso&add=VALUE`
2. The price for the product formatted according to currency formatting preferences specified in *Store Configuration*.
3. The **ProductName** field for the product from the **Product** table.
4. The **BriefDescription** field for the product from the **Product** table formatted according to whether or not it contains HTML.
5. The personal message submitted by the user.

Sales Membership Purchase

/hostedstore/storefront/email/email_membershipsales.lasso

Three variables are made available for inclusion in this email. Each variable is detailed below:

- **HSP_Message** - A message that explains if the membership is new or a renewal.
- **HSP_InvoiceNumber** - The invoice number of the transaction (**ID** field from the **Invoice** table).
- **HSP_CustomerID** - The ID of the customer that purchased a membership (**ID** field from the **Customer** table).

Customer New Account

/hostedstore/storefront/email/email_newaccount.lasso

Two variables are made available for inclusion in this email. Each variable is detailed below:

- **HSP_EmailAddress** - The email address associated with the new customer account (**Email** field from the **Customer** table).
- **HSP_Password** - The password associated with the new customer account (**Password** field from the **Customer** table).

Forgotten Password

/hostedstore/storefront/email/email_password.lasso

One variable is made available for inclusion in this email. **HSP_Password** is the password associated with the new customer account (**Password** field from the **Customer** table).

Shopping Cart Quote

/hostedstore/storefront/email/email_quote.lasso

The information available to this email format file is identical to the information made available to the shopping cart display content page. However, it is likely that, like the default quote email, some of the information will not be needed.

`HSP_Cart` is an array of values and arrays that contains information about the shopping cart items. The elements of the array are detailed below (the number corresponds to the position in the array):

1. An array of arrays that contains the information for the shopping cart items. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The value that should be passed as a form parameter named `modify` to the product modification content page. Hyperlinks to the product modification content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-
ResponseLassoApp=modifyitem.lasso&modify=VALUE`
 2. The **ProductName** field for the item from the **Item** table.
 3. Has a value of `y` if the item was purchased from a wishlist.
 4. An array that contains the names of the item attributes if any exist.
 5. An array that contains the chosen value for item attributes if any exist.
 6. The **Quantity** field for the item from the **Item** table.
 7. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
2. The calculated total item count for the order.
3. The calculated subtotal for the order formatted according to currency formatting preferences specified in *Store Configuration*.
4. The calculated shipping for the order formatted according to currency formatting preferences specified in *Store Configuration*.
5. The calculated total for the order (based on no tax) formatted according to currency formatting preferences specified in *Store Configuration*.
6. An array that contains tax and order totals for all taxable states/provinces specified in preferences specified in *Store Configuration*. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The state/province to which the tax and order total correspond.
 2. The calculated tax for the order (based on the state/province) formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The calculated total for the order (based on the state/province) formatted according to currency formatting preferences specified in *Store Configuration*.

Sales Order Approval

/hostedstore/storefront/email/email_salesapproval.lasso

Two variables are made available for inclusion in this email. Each variable is detailed below:

- **HSP_InvoiceNumber** - The invoice number of the transaction (**ID** field from the **Invoice** table).
- **HSP_CustomerID** - The ID of the customer that purchased a membership (**ID** field from the **Customer** table).

Newsletter Subscription

/hostedstore/storefront/email/email_subscribe.lasso

This email should contain the subscribe message for subscribing to the newsletter. Use the `HS_SubscribeMessage` preference variable if it has been properly set in *Store Configuration*.

Customer Administration

Each of the email files located in the */hostedstore/admincustomer/email/* directory are detailed below.

Sections

- [Cancel Order](#)
- [Forgotten Password](#)
- [Newsletter Subscription](#)
- [Newsletter Unsubscribe](#)
- [Email Wish List](#)

Cancel Order

/hostedstore/admincustomer/email/email_cancelorder.lasso

Two variables are made available for inclusion in this email. Each variable is detailed below:

- **HSP_CustomerID** - The **ID** for the customer canceling the order (**ID** field from the **Customer** table).
- **HSP_OrderID** - The **ID** for the order to cancel (**ID** field from the **Invoice** table).
- **HSP_Reason** - The reason for the cancelation (**CancelReason** field from the **Invoice** table).

Forgotten Password

/hostedstore/admincustomer/email/email_password.lasso

One variable is made available for inclusion in this email. `HSP_Password` is the password associated with the new customer account (**Password** field from the **Customer** table).

Newsletter Subscription

/hostedstore/admincustomer/email/email_subscribe.lasso

This email should contain the subscribe message for subscribing to the newsletter. Use the `HS_SubscribeMessage` preference variable if it has been properly set in *Store Configuration*.

Newsletter Unsubscribe

/hostedstore/admincustomer/email/email_unsubscribe.lasso

This email should contain the unsubscribe message for unsubscribing to the newsletter. Use the `HS_UnsubscribeMessage` preference variable if it has been properly set in *Store Configuration*.

Email Wish List

/hostedstore/admindcustomer/email/email_wishlist.lasso

Two variables are made available for inclusion in this email. Each variable is detailed below:

- `HSP_Message` - The personal message provided by the customer.
- `HSP_WishlistLink` - The hyperlink value for the wish list.

Store Administration

Each of the email files located in the */hostedstore/adminstore/email/* directory are detailed below.

Sections

- [Email Status Change](#)
- [Out of Stock Notification](#)
- [Newsletter Subscription](#)
- [Newsletter Unsubscribe](#)
- [Product In Stock Notification](#)

Email Status Change

/hostedstore/adminstore/email/email_customerstatus.lasso

HSP_Order is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ID** field for the order from the **Invoice** table.
2. The **OrderID** field for the order from the **Invoice** table.
3. The **DateCreated** field for the order from the **Invoice** table formatted according to date and time formatting preferences specified in *Store Configuration*.
4. The **Status** field for the order from the **Invoice** table.
5. A summary of the dates associated with an order. Include the **DateCanceled**, **DateShipped**, **DateBackOrdered**, **DateProcessed**, **DateApproved**, **DatePosted** and **DateCreated** fields for the order from the **Invoice** table formatted according to date and time formatting preferences specified in *Store Configuration*.
6. An array that contains the billing address for the customer. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** and **LastName** fields for the customer from the **Customer** table.
 2. The **Company** field for the customer from the **Customer** table.
 3. The **Address1** field for the customer from the **Customer** table.
 4. The **Address2** field for the customer from the **Customer** table.
 5. The **City**, **State**, and **Postal** fields for the customer from the **Customer** table.
 6. The **Country** field for the customer from the **Customer** table.
 7. The **Email** field for the customer from the **Customer** table.
 8. The **Phone** field for the customer from the **Customer** table.
 9. The **Fax** field for the customer from the **Customer** table.
7. An array that contains the shipping address for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
8. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table with all but the first two characters masked with x characters (x).
 3. The **Expiration** field for the order from the **Invoice** table with a slash character (/) inserted between the month and year.
 4. The **Routing** field for the order from the **Invoice** table.
 5. The **CheckNumber** field for the order from the **Invoice** table.
 6. The **PONumber** field for the order from the **Invoice** table.
 7. The **TransactionID** field for the order from the **Invoice** table.
 8. The **CustomerComments** field for the order from the **Invoice** table.
9. The **AdminComments** field for the order from the **Invoice** table.
10. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 2. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 3. The **Tax** field for the order from the **Invoice** table formatted according to currency

- formatting preferences specified in *Store Configuration*.
4. The **PromotionCode** field for the order.
 5. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 6. The **SubTotalAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **ShippingAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 8. The **TaxAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **DiscountAdjustment** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
 11. A value of **Y** if the order has been line item shipped.
 12. The **ShippingVendor** field for the order from the **Invoice** table.
 13. The **TrackingNumber** field for the order from the **Invoice** table.
 14. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. A value for internal use.
 2. The **OrderStatus** field for the item from the **Item** table.
 3. The value that should be passed as a form parameter named **ID** to the product detail content page. Hyperlinks to the product detail content page should use:
`[$HS_PathToStore]/HostedStore.LassoApp?-ResponseLassoApp=detail.lasso&ID=VALUE`
 4. The **ProductID** field for the item from the **Item** table.
 5. The **ProductName** field for the item from the **Item** table.
 6. Has a value of **Y** if the item was purchased from a wishlist.
 7. The **Quantity** field for the item from the **Item** table.
 8. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 9. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 10. The **ShippingVendor** field for the item from the **Item** table.
 11. The **TrackingNumber** field for the item from the **Item** table.
 12. The **DateShipped** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
 13. The **DateBackOrdered** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
 14. The **DateCanceled** field for the item from the **Item** table formatted according to date formatting preferences specified in *Store Configuration*.
 15. An array that contains the names of the item attributes and the chosen value for the attributes if any exist.

Out of Stock Notification

/hostedstore/adminstore/email/email_outofstock.lasso

`HSP_OutOfStock` is an array of values that contains information about the out of stock product. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ProductID** field for the product from the **Product** table.
2. The **ProductName** field for the product from the **Product** table.
3. The **WebSellQty** field for the product from the **Product** table.
4. The **OnHand** field for the product from the **Product** table (updated with the new value).
5. The **OutOfStockDisplay** field for the product from the **Product** table.

Newsletter Subscription

/hostedstore/adminstore/email/email_subscribe.lasso

This email should contain the subscribe message for subscribing to the newsletter. Use the `HS_SubscribeMessage` preference variable if it has been properly set in *Store Configuration*.

Newsletter Unsubscribe

/hostedstore/adminstore/email/email_unsubscribe.lasso

This email should contain the unsubscribe message for unsubscribing to the newsletter. Use the `HS_UnsubscribeMessage` preference variable if it has been properly set in *Store Configuration*.

Product In Stock Notification

/hostedstore/adminstore/email/email_waitlist.lasso

`HSP_WaitListProduct` is an array of values that contains information about the product. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ProductID** field for the product from the **Product** table encrypted for passing as the `ID` parameter to the product detail page.
2. The **ProductID** field for the product from the **Product** table.
3. The **ProductName** field for the product from the **Product** table.

Affiliate Administration

Each of the email files located in the */hostedstore/affiliate/email/* directory are detailed below.

Sections

- [Forgotten Password](#)

Forgotten Password

/hostedstore/affiliate/email/email_password.lasso

One variable is made available for inclusion in this email. `HSP_Password` is the password associated with the affiliate account (**Password** field from the **Affiliate** table).

Supplier Administration

Sections

- [Customer Order Status Notification](#)
- [Store Order Status Notification](#)

Customer Order Status Notification

`/hostedstore/adminsupplier/email/email_customerstatus.lasso`

`HSP_Order` is an array of values and arrays that contains information about the order. The elements of the array are detailed below (the number corresponds to the position in the array):

1. The **ID** field for the order from the **Invoice** table.
2. An array that contains the payment information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **PaymentType** field for the order from the **Invoice** table.
 2. The **Account** field for the order from the **Invoice** table.
 3. The **Expiration** field for the order from the **Invoice** table.
3. An array that contains the shipping information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **FirstName** field for the order from the **Invoice** table.
 2. The **LastName** field for the order from the **Invoice** table.
 3. The **Company** field for the order from the **Invoice** table.
 4. The **Address1** field for the order from the **Invoice** table.
 5. The **Address2** field for the order from the **Invoice** table.
 6. The **City** field for the order from the **Invoice** table.
 7. The **State** field for the order from the **Invoice** table.
 8. The **Postal** field for the order from the **Invoice** table.
 9. The **Country** field for the order from the **Invoice** table.
4. An array of arrays that contains the ordered items information for the order. The elements of the array are detailed below (the number corresponds to the position in the array):
 1. The **ProductName** field for the item from the **Item** table.
 2. Has a value of `y` if the item was purchased from a wishlist.
 3. An array that contains the names of the item attributes if any exist.
 4. An array that contains the chosen value for item attributes if any exist.
 5. The **Quantity** field for the item from the **Item** table.
 6. The **Price** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
 7. The **TotalPrice** field for the item from the **Item** table formatted according to currency formatting preferences specified in *Store Configuration*.
5. The **ItemCount** field for the order from the **Invoice** table.
6. The **SubTotal** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
7. The **Shipping** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
8. The **Tax** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
9. The **Discount** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.
10. The **Total** field for the order from the **Invoice** table formatted according to currency formatting preferences specified in *Store Configuration*.

Store Order Status Notification

(Future Feature)

Help Center

Each of the email files located in the */hostedstore/helpcenter/email/* directory are detailed below.

Sections

- [Customer Service Inquiry](#)

Customer Service Inquiry

/hostedstore/helpcenter/email/email_customerservice.lasso

Three variables are made available for inclusion in this email. Each variable is detailed below:

- **HSP_Name** - The name of the person submitting the inquiry.
- **HSP_Subject** - The nature of the inquiry.
- **HSP_Message** - The message.

HostedStore Hooks

The features in HostedStore will not be all things to all people. Some users will want more functionality. Some feature requests may not be of general interest so the user will want to customize HostedStore to their own needs. Or, the feature may be of general interest, but the user needs the feature sooner than it can be implemented as a feature in HostedStore. A series of "hooks" and the ability to bypass core code have been made available for this specific purpose. In fact, it is hoped that users will make customizations to HostedStore and make the modifications available to other HostedStore users--either for free or as a fee-based module. To get the process started, some modules have been created and made available for download from the HostedStore Web site.

Sections

- [How Hook Points Work](#)
- [Affiliate Administration Hooks](#)
- [Customer Administration Hooks](#)
- [Store Administration Hooks](#)
- [Storefront Hooks](#)
- [Cookies](#)

How Hook Points Work

Each hook point includes a specific format file from the */hostedstore/thirdparty/hooks/* directory. Code can be placed directly in the format files or the `Include` LDML tag can be used to include code from a different format file in another directory installed specifically for the third party module. The latter method is suggested in order to make it easy for several third party modules to be used in the same hook. Note that if several third party modules are used in the same hook, then compatibility may be an issue. It is hoped that all third party vendors will "play nice" and publish how their module affects the code in HostedStore so maximum compatibility and interoperability can be ensured.

Four types of variables may be available in a hook point format file:

- **Hook point variables** - Variables that indicate another hook point format file should be included. The entry hook point always includes its format file. All other hook points include their format file if an appropriate hook point variable is set. Hook point variables can be set at any point before the actual hook point call. It is recommended for consistency that all variables that execute hook points are set in the entry hook point.
- **Code bypass variables** - Variables that a disable a block of HostedStore core code. The included format file for each hook point can set a variable, or possibly several variables, that will disable the appropriate sections of internal HostedStore code that follow. This feature can be used to override HostedStore code and insert custom code to perform a variation of the original code or a completely different function.
- **Email variables** - Variables that set the subject for email messages sent from the store. This feature can be used to override the default email subjects used by HostedStore.
- **Error variables** - Variables that set messages reported when an error is encountered. This feature can be used to override the default error messages returned by HostedStore.

An attempt has been made to make the naming of variables and hook point format files as intuitive as possible. However, due to operating system restrictions on file name length some hook point format files have less than ideal names. Be sure to carefully read the documentation to ensure that the proper file is modified.

If a hook point or hook variable is needed, but not currently available, then be sure to request that it is added in a future release. Modification requests can be submitted in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Affiliate Administration Hooks

The details for each hook point, the location in the code, how inserted code may affect HostedStore, and the variable(s) to override HostedStore code are detailed below for each internal format file. All of the *Affiliate Account Administration* hook point format files are located in the `/hostedstore/thirdparty/hooks/affiliate/` directory.

Each affiliate format file includes two hook format files:

- *affiliate_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`), but before the entry hook specific to the format file.
- *affiliate_exit.lasso* - Included prior to setting and returning the store template and content page, but after the exit hook specific to the format file.

Sections

- [affiliate/index.lasso](#)

affiliate/index.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=affiliate/index.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/affiliate/index_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

- `HSH_Email_LostPassword` - Assign a value that will be used as the subject for the email sent to affiliates when a lost password request is made. The value defaults to *Your password has arrived.*

Error variables:

- `HSH_Error_LoginFailure` - Assign a value that will be used as the error message returned when an attempt has been made to login with an invalid email address and password combination. The value defaults to *The email address and password provided do not match any of our affiliates.*
- `HSH_Error_LostPasswordFailure` - Assign a value that will be used as the error message returned when an attempt has been made to email a lost password with an invalid email address. The value defaults to *The email address and password provided do not match any of our affiliates.*
- `HSH_Error_LostPasswordSuccess` - Assign a value that will be used as the message returned when the password has been successfully retrieved and emailed. The value defaults to *Your password has been mailed to you.*
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields.*

Hook Points

Each file name is prefixed with `index_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `affiliate_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `index_entry.lasso` - Included immediately after `affiliate_entry.lasso`.
- `index_exit.lasso` - Included immediately prior to `affiliate_exit.lasso` if `HSH_Exit` has a value.
- `affiliate_exit.lasso` - Included prior to setting and returning the store template and content page.

Customer Administration Hooks

The details for each hook point, the location in the code, how inserted code may affect HostedStore, and the variable(s) to override HostedStore code are detailed below for each internal format file. All of the *Customer Account Administration* hook point format files are located in the `/hostedstore/thirdparty/hooks/admincustomer/` directory.

Each customer administration format file includes two hook format files:

- *admincustomer_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`), but before the entry hook specific to the format file.
- *admincustomer_exit.lasso* - Included prior to setting and returning the store template and content page, but after the exit hook specific to the format file.

Sections

- [admincustomer/address.lasso](#)
- [admincustomer/admin.lasso](#)
- [admincustomer/cancelorder.lasso](#)
- [admincustomer/files.lasso](#)
- [admincustomer/index.lasso](#)
- [admincustomer/logout.lasso](#)
- [admincustomer/shoppinglist.lasso](#)
- [admincustomer/wishlist.lasso](#)

admincustomer/address.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/address.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

- `HSH_Error_BillingReserved` - Assign a value that will be used as the error message returned when an attempt is made to save an address book entry named *Billing*. The value defaults to *Billing cannot be used as an entry name*.
- `HSH_Error_DuplicateEntry` - Assign a value that will be used as the error message returned when an attempt is made to save an address book entry with a name for an address book already exists. The value defaults to *\$EntryName has already been used as an entry name*.
- `HSH_Error_EntryAddedFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to add an address book entry. The value defaults to *An error occurred adding the \$EntryName entry. Please try again*.
- `HSH_Error_EntryAddedSuccess` - Assign a value that will be used as the message returned when an address book entry is successfully added. The value defaults to *\$EntryName entry added*.
- `HSH_Error_EntryDeleted` - Assign a value that will be used as the message returned when an address book entry is deleted. The value defaults to *\$EntryName entry removed*.
- `HSH_Error_EntryUpdatedFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to update an address book entry. The value defaults to *An error occurred updating the \$EntryName entry. Please try again*.
- `HSH_Error_EntryUpdatedSuccess` - Assign a value that will be used as the message returned when an address book entry is successfully updated. The value defaults to *\$EntryName entry updated*.
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields*.
- `HSH_Error_SearchFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while searching for address book entries. The value defaults to *An error occurred accessing while your file. Please try again*.

Hook Points

Each file name is prefixed with `address_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admincustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).

- *address_entry.lasso* - Included immediately after *admincustomer_entry.lasso*.
- *address_exit.lasso* - Included immediately prior to *admincustomer_exit.lasso*.
- *admincustomer_exit.lasso* - Included prior to setting and returning the store template and content page.

admincustomer/admin.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/admin.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

- `HSH_Email_SubscribeNewsletter` - Assign a value that will be used as the subject for the email sent to the mailing list manager when a customer subscribes to the store newsletter. The value defaults to the `HS_SubscribeSubject` variable set in store administration.
- `HSH_Email_UnsubscribeNewsletter` - Assign a value that will be used as the subject for the email sent to the mailing list manager when a customer unsubscribes from the store newsletter. The value defaults to the `HS_UnsubscribeSubject` variable set in store administration.

Error variables:

- `HSH_Error_OrderSearchFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while searching for an order. The value defaults to *That order could not be found. Please try again.*
- `HSH_Error_PasswordMismatch` - Assign a value that will be used as the error message returned when an attempt is made to change the account password with passwords that do not match. The value defaults to *The new password and password verification did not match. Please try again.*
- `HSH_Error_PasswordUpdateFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to update the account password. The value defaults to *An error occurred updating your password. Please try again.*
- `HSH_Error_PasswordUpdateSuccess` - Assign a value that will be used as the message returned when the account password is successfully updated. The value defaults to *Password updated.*
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields.*
- `HSH_Error_SearchFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while searching for the account. The value defaults to *An error occurred accessing your file. Please try again.*
- `HSH_Error_UpdateFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while updating the account. The value defaults to *An error occurred updating your file. Please try again.*
- `HSH_Error_BillingUpdateFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while updating the account billing address. The value defaults to *An error occurred updating the billing address. Please try again.*
- `HSH_Error_BillingUpdateSuccess` - Assign a value that will be used as the message returned when the account billing address is successfully updated. The value defaults to *Billing address updated.*

Hook Points

Each file name is prefixed with `admin_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- *admindcustomer_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- *admin_entry.lasso* - Included immediately after *admindcustomer_entry.lasso*.
- *admin_exit.lasso* - Included immediately prior to *admindcustomer_exit.lasso*.
- *admindcustomer_exit.lasso* - Included prior to setting and returning the store template and content page.

admincustomer/cancelorder.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/cancelorder.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

- `HSH_Email_CancelOrder` - Assign a value that will be used as the subject for the emailed cancellation request sent to `HS_SalesEmail` when a customer cancels an order. The value defaults to *Order HSP_OrderID has been canceled*.

Error variables:

- `HSH_Error_CancelFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to cancel the order. The value defaults to *There was a problem canceling the order. Please try again*.
- `HSH_Error_CancelSuccess` - Assign a value that will be used as the message returned when the order cancellation request is successfully added. The value defaults to *Order cancelation request sent*.
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields*.

Hook Points

Each file name is prefixed with `cancelorder_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admincustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `cancelorder_entry.lasso` - Included immediately after `admincustomer_entry.lasso`.
- `cancelorder_exit.lasso` - Included immediately prior to `admincustomer_exit.lasso`.
- `admincustomer_exit.lasso` - Included prior to setting and returning the store template and content page.

admincustomer/files.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/files.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

- `HSH_Error_DownloadLimit` - Assign a value that will be used as the error message returned when an attempt is made to download a file that has reached its download limit. The value defaults to `(Action_Param:'limit')` has reached its download limit.

Hook Points

Each file name is prefixed with `files_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admincustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `files_entry.lasso` - Included immediately after `admincustomer_entry.lasso`.
- `files_exit.lasso` - Included immediately prior to `admincustomer_exit.lasso`.
- `admincustomer_exit.lasso` - Included prior to setting and returning the store template and content page.

admincustomer/index.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/index.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

- `HSH_Email_LostPassword` - Assign a value that will be used as the subject for the email sent to customers when a lost password request is made. The value defaults to *Your HS_StoreName password has arrived*.

Error variables:

- `HSH_Error_LoginFailure` - Assign a value that will be used as the error message returned when an attempt has been made to login with an invalid email address and password combination. The value defaults to *Our records do not reflect a customer with that email address and password*.
- `HSH_Error_LostPasswordFailure` - Assign a value that will be used as the error message returned when an attempt has been made to email a lost password with an invalid email address. The value defaults to *Our records do not reflect a customer with that email address*.
- `HSH_Error_LostPasswordSuccess` - Assign a value that will be used as the message returned when the password has been successfully retrieved and emailed. The value defaults to *Your password has been emailed to \$f1*.
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields*.

Hook Points

Each file name is prefixed with `index_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admincustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `index_entry.lasso` - Included immediately after `admincustomer_entry.lasso`.
- `index_exit.lasso` - Included immediately prior to `admincustomer_exit.lasso`.
- `admincustomer_exit.lasso` - Included prior to setting and returning the store template and content page.

admincustomer/logout.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/logout.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `logout_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- *admincustomer_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- *logout_entry.lasso* - Included immediately after *admincustomer_entry.lasso*.
- *logout_exit.lasso* - Included immediately prior to *admincustomer_exit.lasso*.
- *admincustomer_exit.lasso* - Included prior to setting and returning the store template and content page.

admincustomer/shoppinglist.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/shoppinglist.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

- `HSH_Error_AddListSuccess` - Assign a value that will be used as the message returned when a shopping list is successfully created. The value defaults to `$f3 added to your shopping lists`.
- `HSH_Error_DuplicateListName` - Assign a value that will be used as the error message returned when an attempt is made to save a shopping list with a name for a shopping list that already exists. The value defaults to `A shopping list with that name already exists. Please choose a different name`.
- `HSH_Error_ListAddFailure` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a shopping list to the cart. The value defaults to `The entire $HSH_ListName shopping list could not be added to your cart. Problems encountered are listed above`.
- `HSH_Error_ListAddSuccess` - Assign a value that will be used as the message returned when an entire shopping list is successfully added to the cart. The value defaults to `The entire $HSH_ListName shopping list was added to your cart`.
- `HSH_Error_ListDeleted` - Assign a value that will be used as the message returned when a shopping list is successfully deleted. The value defaults to `$HSH_ListName deleted`.
- `HSH_Error_NoListName` - Assign a value that will be used as the error message returned when the shopping list name field is left blank. The value defaults to `Please assign a name to the shopping list`.
- `HSH_Error_ProductAddFailure` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a shopping list item to the cart. The value defaults to `An error occurred while adding $HSH_ProductName to your cart`.
- `HSH_Error_ProductAddSuccess` - Assign a value that will be used as the message returned when a shopping list item is successfully added to the cart. The value defaults to `$HSH_ProductName added to your cart`.
- `HSH_Error_ProductAttributesChanged` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a shopping list item to the cart for which the attributes have changed. The value defaults to `$HSH_ProductName not added: Product attributes have changed. Please purchase the product through the storefront`.
- `HSH_Error_ProductNotAvailable` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a shopping list item to the cart that is not currently offered. The value defaults to `$HSH_ProductName not added: The product is currently not for sale or is out of stock`.
- `HSH_Error_UpdateListFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to update a shopping list. The value defaults to `There was a problem updating $HSH_ListName. Please try again`.

- `HSH_Error_UpdateListSuccess` - Assign a value that will be used as the message returned when a shopping list is successfully updated. The value defaults to `$HSI_ListName updated`.

Hook Points

Each file name is prefixed with `shoppinglist_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admindcustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `shoppinglist_entry.lasso` - Included immediately after `admindcustomer_entry.lasso`.
- `shoppinglist_exit.lasso` - Included immediately prior to `admindcustomer_exit.lasso`.
- `admindcustomer_exit.lasso` - Included prior to setting and returning the store template and content page.

admincustomer/wishlist.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=admincustomer/wishlist.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

- `HSH_Error_EmailWishlist` - Assign a value that will be used as the message returned when a wish list has been successfully emailed. The value defaults to *Your wish list has been sent to \$f2.*
- `HSH_Error_ItemDeleted` - Assign a value that will be used as the message returned when a wish list item is successfully deleted. The value defaults to *Item removed.*
- `HSH_Error_ProductAddFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to add a wish list item to the cart. The value defaults to *There was a problem adding the item to your cart. Please try again.*
- `HSH_Error_ProductAddSuccess` - Assign a value that will be used as the message returned when a wish list item is successfully added to the cart. The value defaults to *\$HSI_ProductName has been added to your cart.*
- `HSH_Error_ProductAttributesChanged` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart for which the attributes have changed. The value defaults to *Product attributes have changed since the item was added to the wishlist. Please select the item below and choose a new configuration.*
- `HSH_Error_ProductNotAvailable` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart that is not currently offered. The value defaults to *That product is currently not for sale.*
- `HSH_Error_ProductNotFound` - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart that is not in the product catalog. The value defaults to *That product could not be found in our catalog or is currently not for sale.*
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields.*

Hook Points

Each file name is prefixed with `wishlist_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `admincustomer_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables

prefixed with `HSH_`).

- *wishlist_entry.lasso* - Included immediately after *admincustomer_entry.lasso*.
- *wishlist_exit.lasso* - Included immediately prior to *admincustomer_exit.lasso*.
- *admincustomer_exit.lasso* - Included prior to setting and returning the store template and content page.

Store Administration Hooks

The details for each hook point, the location in the code, how inserted code may affect HostedStore, and the variable(s) to override HostedStore code are detailed below for each internal format file. All of the Store Administration hook point format files are located in the `/hostedstore/thirdparty/hooks/adminstore/` directory.

Each Store Administration format file includes two hook format files:

- *adminstore_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`), but before the entry hook specific to the format file.
- *adminstore_exit.lasso* - Included prior to setting and returning the store template and content page, but after the exit hook specific to the format file.

Sections

- [adminstore/adminuser.lasso](#)
- [adminstore/affiliate.lasso](#)
- [adminstore/affiliatepayout.lasso](#)
- [adminstore/affiliatepayouthistory.lasso](#)
- [adminstore/application.lasso](#)
- [adminstore/autoship.lasso](#)
- [adminstore/autoshipsweep.lasso](#)
- [adminstore/batch.lasso](#)
- [adminstore/customer.lasso](#)
- [adminstore/image.lasso](#)
- [adminstore/indexpage.lasso](#)
- [adminstore/inventory.lasso](#)
- [adminstore/maintenance.lasso](#)
- [adminstore/manualorder.lasso](#)
- [adminstore/manualreturn_checkout.lasso](#)
- [adminstore/manualreturn.lasso](#)
- [adminstore/menu.lasso](#)
- [adminstore/order.lasso](#)
- [adminstore/ordermanager.lasso](#)
- [adminstore/price.lasso](#)
- [adminstore/priceglobaladjust.lasso](#)
- [adminstore/pricegroup.lasso](#)
- [adminstore/product.lasso](#)
- [adminstore/promotion.lasso](#)
- [adminstore/review.lasso](#)
- [adminstore/supplier.lasso](#)
- [adminstore/upload.lasso](#)

adminstore/adminuser.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/adminuser.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/adminuser_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `adminuser_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `adminuser_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `adminuser_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/affiliate.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/affiliate.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/affiliate_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `affiliate_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `affiliate_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `affiliate_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/affiliatepayout.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/affiliatepayout.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/affiliatepayout_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `affiliatepayout_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `affiliatepayout_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `affiliatepayout_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/affiliatepayouthistory.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/affiliatepayouthistory.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/affiliatepayouthistory_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `affiliatepayouthistory_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `affiliatepayouthistory_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `affiliatepayouthistory_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/application.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/application.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/application_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `application_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `application_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `application_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/autoship.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/autoship.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/autoship_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `autoship_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `autoship_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `autoship_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/autoshipsweep.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/autoshipsweep.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/autoshipsweep_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `autoshipsweep_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `autoshipsweep_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `autoshipsweep_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/batch.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/batch.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/batch_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `batch_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `batch_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `batch_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/customer.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/customer.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/customer_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `customer_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `customer_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `customer_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/image.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/image.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/image_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `image_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HST_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `image_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `image_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/index.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/index.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/index_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `index_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `index_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `index_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/inventory.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/inventory.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/inventory_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `inventory_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `inventory_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `inventory_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/maintenance.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/maintenance.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/maintenance_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `maintenance_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `maintenance_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `maintenance_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/manualcheckout.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/manualcheckout.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/manualcheckout_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `manualcheckout_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `manualcheckout_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `manualcheckout_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/manualorder.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/manualorder.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/manualorder_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `manualorder_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `manualorder_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `manualorder_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/manualreturn_checkout.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/manualreturn_checkout.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/manualreturn_checkout_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `manualreturn_checkout_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `manualreturn_checkout_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `manualreturn_checkout_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/manualreturn.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/manualreturn.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/manualreturn_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `manualreturn_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `manualreturn_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `manualreturn_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/menu.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/menu.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/menu_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `menu_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HST_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `menu_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `menu_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/order.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/order.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/order_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `order_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `order_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `order_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/ordermanager.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/ordermanager.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/ordermanager_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `ordermanager_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `ordermanager_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `ordermanager_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/price.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/price.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/price_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `price_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `price_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `price_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/priceglobaladjust.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/priceglobaladjust.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/priceglobaladjust_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `priceglobaladjust_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `priceglobaladjust_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `priceglobaladjust_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/pricegroup.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/pricegroup.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/pricegroup_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `pricegroup_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `pricegroup_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `pricegroup_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/product.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/product.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/product_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `product_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `product_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `product_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/promotion.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/promotion.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/promotion_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `promotion_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `promotion_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `promotion_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/review.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/review.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/review_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `review_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HST_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `review_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `review_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/supplier.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/supplier.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/supplier_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `supplier_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `supplier_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `supplier_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

adminstore/upload.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=adminstore/upload.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/adminstore/upload_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `upload_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `adminstore_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HST_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `upload_entry.lasso` - Included immediately after `adminstore_entry.lasso`.
- `upload_exit.lasso` - Included immediately prior to `adminstore_exit.lasso`.
- `adminstore_exit.lasso` - Included prior to setting and returning the store template and content page.

Storefront Hooks

The details for each hook point, the location in the code, how inserted code may affect HostedStore, and the variable(s) to override HostedStore code are detailed below for each internal format file. All of the storefront hook point format files are located in the */hostedstore/thirdparty/hooks/storefront/* directory.

Each storefront format file includes two hook format files:

- *storefront_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with **HS_**), initialization of internal variables (variables prefixed with **HSI_**) and initialization of hook variables (variables prefixed with **HSH_**), but before the entry hook specific to the format file.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page, but after the exit hook specific to the format file.

Sections

- [cart.lasso](#)
- [cartrecover.lasso](#)
- [cartsave.lasso](#)
- [checkout.lasso](#)
- [closed.lasso](#)
- [detail.lasso](#)
- [emailfriend.lasso](#)
- [helpcenter.lasso](#)
- [index.lasso](#)
- [modifyitem.lasso](#)
- [modifypayment.lasso](#)
- [modifyshipping.lasso](#)
- [review.lasso](#)
- [search.lasso](#)
- [wishlist.lasso](#)

cart.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=cart.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/cart_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

- `HSH_Email_Quote` - Assign a value that will be used as the subject for the email sent to customers when a quote request is made. The value defaults to *Your requested HS_StoreName quote*.

Error variables:

- `HSH_Error_EmailQuote` - Assign a value that will be used as the message returned when a cart quotes has been successfully emailed. The value defaults to *A quote was emailed to \$f2*.
- `HSH_Error_NoEmailAddress` - Assign a value that will be used as the error message returned when an email address is not specified. The value defaults to *Please provide an email address*.
- `HSH_Error_ProductNotAvailable` - Assign a value that will be used as the error message returned when an error occurred while attempting to add an item to the cart that is not currently offered. The value defaults to *That product is currently not for sale*.
- `HSH_Error_ProductNotFound` - Assign a value that will be used as the error message returned when an error occurred while attempting to add an item to the cart that is not in the product catalog. The value defaults to *That product could not be found in our catalog or is currently not for sale*.
- `HSH_Error_RequiredAttributes` - Assign a value that will be used as the error message returned when a required attribute is left blank. The value defaults to *Please make a selection for each option below before adding the item to your cart*.

Hook Points

Each file name is prefixed with `cart_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `cart_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `cart_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

cartrecover.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=cartrecover.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/cartrecover_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `cartrecover_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `cartrecover_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `cartrecover_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

cartsave.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=cartsave.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/cartsave_exit.lasso` format file.

Code bypass variables:

None at this time.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

Each file name is prefixed with `cartsave_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `cartsave_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `cartsave_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

checkout.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=checkout.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/checkout_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

- `HSH_Email_NewAccount` - Assign a value that will be used as the subject for the email sent to customers when a new account is created. The value defaults to *Your HS_StoreName account information*.
- `HSH_Email_SubscribeNewsletter` - Assign a value that will be used as the subject for the email sent to the mailing list manager when a customer subscribes to the store newsletter. The value defaults to the `HS_SubscribeSubject` variable set in store administration.
- `HSH_Email_LostPassword` - Assign a value that will be used as the subject for the email sent to customers when a lost password request is made. The value defaults to *Your HS_StoreName password*.

Error variables:

- `HSH_Error_AccountAddFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to create a customer account. The value defaults to *There was a problem creating your account. Please try again*.
- `HSH_Error_AddressBookAddFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to create an address book entry. The value defaults to *There was a problem creating the address book entry. Please try again*.
- `HSH_Error_AddressBookDuplicate` - Assign a value that will be used as the error message returned when an attempt is made to create an address book entry with a name already used for an address book entry. The value defaults to *An address book entry with that name already exists*.
- `HSH_Error_AddressBookFailure` - Assign a value that will be used as the error message returned when an attempt is made to retrieve an address book entry. The value defaults to *There was a problem loading your address book. Please try again*.
- `HSH_Error_AddressBookInvalid` - Assign a value that will be used as the error message returned when an attempt is made to use an address book entry that has a country to which shipping is not offered. The value defaults to *The \$f19 address book entry does not contain a valid shipping country. Please see the country select list below for valid shipping countries*.
- `HSH_Error_CCNumberExpired` - Assign a value that will be used as the error message returned when a provided credit card expiration date is expired. The value defaults to *The credit card provided has expired*.
- `HSH_Error_CCNumberInvalid` - Assign a value that will be used as the error message returned when

a provided credit card number is invalid. The value defaults to *The credit card number provided is invalid.*

- **HSH_Error_CCNumberMismatch** - Assign a value that will be used as the error message returned when a provided credit card number does not match the provided credit card type. The value defaults to *Credit card type and number do not match.*
- **HSH_Error_EmailAddressMismatch** - Assign a value that will be used as the error message returned when provided email addresses do not match. The value defaults to *Please make sure that the email address and verified email address match.*
- **HSH_Error_LoginFailure** - Assign a value that will be used as the error message returned when an attempt has been made to login with an invalid email address and password combination. The value defaults to *The email address and password provided do not match any of our customers.*
- **HSH_Error_LostPasswordFailure** - Assign a value that will be used as the error message returned when an attempt has been made to email a lost password with an invalid email address. The value defaults to *The email address provided does not match any of our customers.*
- **HSH_Error_LostPasswordSuccess** - Assign a value that will be used as the message returned when the password has been successfully retrieved and emailed. The value defaults to *Your password has been mailed to you.*
- **HSH_Error_NoCCNumber** - Assign a value that will be used as the error message returned when a credit card number is not provided. The value defaults to *Please provide a credit card number.*
- **HSH_Error_NoCCType** - Assign a value that will be used as the error message returned when a credit card type is not selected. The value defaults to *Please select a credit card type.*
- **HSH_Error_NoCheckingAccountNumber** - Assign a value that will be used as the error message returned when a checking account number is not provided. The value defaults to *Please provide a checking account number.*
- **HSH_Error_NoCheckNumber** - Assign a value that will be used as the error message returned when a check number is not provided. The value defaults to *Please provide a check number.*
- **HSH_Error_NoCustomerType** - Assign a value that will be used as the error message returned when a customer type is not selected. The value defaults to *Please specify if you are a new or returning customer.*
- **HSH_Error_NoExpirationMonth** - Assign a value that will be used as the error message returned when a credit card expiration month is not provided. The value defaults to *Please provide an expiration month.*
- **HSH_Error_NoExpirationYear** - Assign a value that will be used as the error message returned when a credit card expiration year is not provided. The value defaults to *Please provide an expiration year.*
- **HSH_Error_NoPaymentMethod** - Assign a value that will be used as the error message returned when a payment method is not selected. The value defaults to *Please select a payment method.*
- **HSH_Error_NoPONumber** - Assign a value that will be used as the error message returned when a purchase order number is not provided. The value defaults to *Please provide a purchase order number.*
- **HSH_Error_NoRoutingNumber** - Assign a value that will be used as the error message returned when a checking account routing number is not provided. The value defaults to *Please provide a routing number.*
- **HSH_Error_NoSignature** - Assign a value that will be used as the error message returned when a signature is not provided. The value defaults to *Please provide a signature.*
- **HSH_Error_OrderAddFailure** - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to create an order. The value defaults to *There was a problem creating your order. Please try again.*
- **HSH_Error_OrderNotFound** - Assign a value that will be used as the error message returned when an error occurred while attempting to retrieve an existing order. The value defaults to *There was a problem retrieving your order. Please try again.*
- **HSH_Error_PasswordMismatch** - Assign a value that will be used as the error message returned when provided passwords do not match. The value defaults to *Please make sure that the chosen password and verified password match.*
- **HSH_Error_PromotionCodeInvalid** - Assign a value that will be used as the error message returned when a provided promotion code is invalid. The value defaults to *The promotion code entered is either invalid or expired.*
- **HSH_Error_RequiredFields** - Assign a value that will be used as the error message returned when a

required field is left blank. The value defaults to *Please complete all required fields.*

- `HSH_Error_NoTermsAndConditions` - Assign a value that will be used as the error message returned when the terms and conditions checkbox has not been checked. The value defaults to *Please read and agree to the terms and conditions.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `checkout_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `checkout_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `checkout_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

closed.lasso

Returned when the `HS_StoreStatus` variable is set to `Y` in the `/hostedstore/includes/HS_datafile.lasso` store configuration file.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/closed_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `closed_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `closed_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `closed_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

detail.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=detail.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/detail_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassProductDetail` - Assign a value to bypass the main code block that gets the product detail information and sets the `HSP_Product` array variable for the product detail content page (`/hostedstore/storefront/html/product_detail.lasso`).
- `HSH_BypassPreserveCriteria` - Assign a value to bypass the code that sets the `HSP_SearchCriteria` variable used to preserve search criteria and return to the search results list from the product detail content page (`/hostedstore/storefront/html/product_detail.lasso`) and from the shopping cart content page (`/hostedstore/storefront/html/cart_display.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

- `HSH_Error_ProductNotFound` - Assign a value that will be used as the error message returned when an error occurred while attempting to view an item that is not in the product catalog. The value defaults to *Sorry, that product could not be found. Please try again.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `detail_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `detail_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `detail_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

emailfriend.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=emailfriend.lasso`.

Variables

Hook point variables:

- `HSH_PreEmail` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/emailfriend_preemail.lasso` format file.
- `HSH_PostEmail` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/emailfriend_postemail.lasso` format file.
- `HSH_PreDesc` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/emailfriend_predesc.lasso` format file.
- `HSH_PostDesc` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/emailfriend_postdesc.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/emailfriend_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassEmail` - Assign a value to bypass the code that queries the database for product information and emails the product info and customer message to the recipient email address.
- `HSH_BypassDesc` - Assign a value to bypass the code that queries the database for product information to populate the `HSP_Product` array variable.
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `emailfriend_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `emailfriend_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `emailfriend_preemail.lasso` - Included before the code that queries the database for product information and emails the product info and customer message to the recipient email address.
- `emailfriend_postemail.lasso` - Included after the code that queries the database for product information and emails the product info and customer message to the recipient email address.
- `emailfriend_predesc.lasso` - Included before the code that queries the database for product

information to populate the `HSP_Product` array variable.

- *emailfriend_postdesc.lasso* - Included after the code that queries the database for product information to populate the `HSP_Product` array variable.
- *emailfriend_exit.lasso* - Included immediately prior to *storefront_exit.lasso* if `HSH_Exit` has a value.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page.

helpcenter.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=helpcenter.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/helpcenter_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassShow` - Assign a value to bypass the code that sets the default content page from the `show` parameter.
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `helpcenter_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `helpcenter_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `helpcenter_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

index.lasso

Accessed via a URL ending with `HostedStore.LassoApp` or `HostedStore.LassoApp?-ResponseLassoApp=index.lasso`.

Variables

Hook point variables:

- `HSH_PreFeatured` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/index_prefeatured.lasso` format file.
- `HSH_PostFeatured` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/index_postfeatured.lasso` format file.
- `HSH_PreFavorites` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/index_prefavorites.lasso` format file.
- `HSH_PostFavorites` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/index_postfavorites.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/index_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassFeatured` - Assign a value to bypass the code that sets the `HSP_Featured` variable.
- `HSH_BypassFavorites` - Assign a value to bypass the code that sets the `HSP_Favorites` variable.
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

None at this time.

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `index_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `index_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `index_prefeatured.lasso` - Included before the code that sets the `HSP_Featured` variable if `HSH_PreFeatured` has a value.
- `index_postfeatured.lasso` - Included after the code that sets the `HSP_Featured` variable if `HSH_PostFeatured` has a value.
- `index_prefavorites.lasso` - Included before the code that sets the `HSP_Favorites` variable if `HSH_PreFavorites` has a value.

- *index_postfavorites.lasso* - Included after the code that sets the `HSP_Favorites` variable if `HSH_PostFavorites` has a value.
- *index_exit.lasso* - Included immediately prior to *storefront_exit.lasso* if `HSH_Exit` has a value.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page.

modifyitem.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=modifyitem.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyitem_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassApplyModification` - Assign a value to bypass the code that validates and applies changes made to the product configuration.
- `HSH_BypassModifyItem` - Assign a value to bypass the code that sets the `HSP_Product` array variable for the default configurable product modification content page (`/hostedstore/storefront/html/cart_modifyitem.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

- `HSH_Error_RequiredAttributes` - Assign a value that will be used as the error message returned when a required attribute is left blank. The value defaults to *One of the options was left blank. Your previous selections have been restored.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `modifyitem_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `modifyitem_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `modifyitem_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

modifypayment.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=modifypayment.lasso`.

Variables

Hook point variables:

- `HSH_PreValidate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_prevalid.lasso` format file.
- `HSH_PostValidate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_postvalid.lasso` format file.
- `HSH_PreApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_preapply.lasso` format file.
- `HSH_PostApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_postapply.lasso` format file.
- `HSH_PreForm` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_preform.lasso` format file.
- `HSH_PostForm` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_postform.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifypayment_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassPaymentValidation` - Assign a value to bypass the code that validates the submitted payment modification information.
- `HSH_BypassApplyModification` - Assign a value to bypass the code that applies changes made to the payment information.
- `HSH_BypassModifyPayment` - Assign a value to bypass the code that sets the variables for the default payment modification form content page (`/hostedstore/storefront/html/payment_modify.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

- `HSH_Error_CCNumberExpired` - Assign a value that will be used as the error message returned when a provided credit card expiration date is expired. The value defaults to *The credit card provided has expired.*
- `HSH_Error_CCNumberInvalid` - Assign a value that will be used as the error message returned when a provided credit card number is invalid. The value defaults to *The credit card number provided is invalid.*
- `HSH_Error_CCNumberMismatch` - Assign a value that will be used as the error message returned when a provided credit card number does not match the provided credit card type. The value defaults to *Credit card type and number do not match.*
- `HSH_Error_NoCCNumber` - Assign a value that will be used as the error message returned when a credit card number is not provided. The value defaults to *Please provide a credit card number.*
- `HSH_Error_NoCCType` - Assign a value that will be used as the error message returned when a credit card type is not selected. The value defaults to *Please select a credit card type.*

- `HSH_Error_NoCheckingAccountNumber` - Assign a value that will be used as the error message returned when a checking account number is not provided. The value defaults to *Please provide a checking account number.*
- `HSH_Error_NoCheckNumber` - Assign a value that will be used as the error message returned when a check number is not provided. The value defaults to *Please provide a check number.*
- `HSH_Error_NoExpirationMonth` - Assign a value that will be used as the error message returned when a credit card expiration month is not provided. The value defaults to *Please provide an expiration month.*
- `HSH_Error_NoExpirationYear` - Assign a value that will be used as the error message returned when a credit card expiration year is not provided. The value defaults to *Please provide an expiration year.*
- `HSH_Error_NoPaymentMethod` - Assign a value that will be used as the error message returned when a payment method is not selected. The value defaults to *Please select a payment method.*
- `HSH_Error_NoPONumber` - Assign a value that will be used as the error message returned when a purchase order number is not provided. The value defaults to *Please provide a purchase order number.*
- `HSH_Error_NoRoutingNumber` - Assign a value that will be used as the error message returned when a checking account routing number is not provided. The value defaults to *Please provide a routing number.*
- `HSH_Error_NoSignature` - Assign a value that will be used as the error message returned when a signature is not provided. The value defaults to *Please provide a signature.*
- `HSH_Error_PaymentUpdateFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to update the order. The value defaults to *There was a problem updating your billing information. Please try again.*
- `HSH_Error_NoTermsAndConditions` - Assign a value that will be used as the error message returned when the terms and conditions checkbox has not been checked. The value defaults to *Please read and agree to the terms and conditions.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `modifypayment_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `modifypayment_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `modifypayment_prevalid.lasso` - Included before the code that validates submitted payment information modifications if `HSH_PreValidate` has a value.
- `modifypayment_postvalid.lasso` - Included after the code that validates submitted payment information modifications if `HSH_PostValidate` has a value.
- `modifypayment_preapply.lasso` - Included before the code that posts submitted payment information modifications to the database if `HSH_PreApply` has a value.
- `modifypayment_postapply.lasso` - Included after the code that posts submitted payment information modifications to the database if `HSH_PostApply` has a value.
- `modifypayment_preform.lasso` - Included before the code that sets the variables for the default payment modification form content page (`/hostedstore/storefront/html/payment_modify.lasso`) if `HSH_PreForm` has a value.
- `modifypayment_postform.lasso` - Included after the code that sets the variables for the default payment modification form content page (`/hostedstore/storefront/html/payment_modify.lasso`) if `HSH_PostForm` has a value.
- `modifypayment_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

modifyshipping.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=modifyshipping.lasso`.

Variables

Hook point variables:

- `HSH_PreValidate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_prevalid.lasso` format file.
- `HSH_PostValidate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_postvalid.lasso` format file.
- `HSH_PreApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_preapply.lasso` format file.
- `HSH_PostApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_postapply.lasso` format file.
- `HSH_PreForm` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_preform.lasso` format file.
- `HSH_PostForm` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_postform.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/modifyship_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassShippingValidation` - Assign a value to bypass the code that validates the submitted shipping modification information.
- `HSH_BypassApplyModification` - Assign a value to bypass the code that validates and applies changes made to the shipping information.
- `HSH_BypassModifyShipping` - Assign a value to bypass the code that sets the numeric variables for the default shipping modification form content page (`/hostedstore/storefront/html/shipping_modify.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields.*
- `HSH_Error_ShippingUpdateFailure` - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to update an order. The value defaults to *There was a problem updating your shipping information. Please try again.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `modifyship_`. The hooks are listed below in the order in which the hook is called

by HostedStore.

- *storefront_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- *modifyship_entry.lasso* - Included immediately after *storefront_entry.lasso*.
- *modifyship_prevalid.lasso* - Included before the code that validates submitted shipping information modifications if `HSH_PreValidate` has a value.
- *modifyship_postvalid.lasso* - Included after the code that validates submitted shipping information modifications if `HSH_PostValidate` has a value.
- *modifyship_preapply.lasso* - Included before the code that posts submitted shipping information modifications to the database if `HSH_PreApply` has a value.
- *modifyship_postapply.lasso* - Included after the code that posts submitted shipping information modifications to the database if `HSH_PostApply` has a value.
- *modifyship_preform.lasso* - Included before the code that sets the variables for the default shipping modification form content page (*/hostedstore/storefront/html/shipping_modify.lasso*) if `HSH_PreForm` has a value.
- *modifyship_postform.lasso* - Included after the code that sets the variables for the default shipping modification form content page (*/hostedstore/storefront/html/shipping_modify.lasso*) if `HSH_PostForm` has a value.
- *modifyship_exit.lasso* - Included immediately prior to *storefront_exit.lasso* if `HSH_Exit` has a value.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page.

review.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=review.lasso`.

Variables

Hook point variables:

- `HSH_PreApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_preapply.lasso` format file.
- `HSH_PostApply` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_postapply.lasso` format file.
- `HSH_PreList` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_prelist.lasso` format file.
- `HSH_PostList` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_postlist.lasso` format file.
- `HSH_PreRead` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_preread.lasso` format file.
- `HSH_PostRead` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_postread.lasso` format file.
- `HSH_PreRate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_prerate.lasso` format file.
- `HSH_PostRate` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_postrate.lasso` format file.
- `HSH_PreLogin` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_prelogin.lasso` format file.
- `HSH_PostLogin` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_postlogin.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/review_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassPostReview` - Assign a value to bypass the code that verifies submitted reviews are properly structured and adds the review to the review database.
- `HSH_BypassReviewList` - Assign a value to bypass the code that creates a list of reviews and gets product detail information when `view` has a value.
- `HSH_BypassReadReview` - Assign a value to bypass the code that gets review detail information, updates read count, creates a list of reviews, and gets product detail information when `read` has a value.
- `HSH_BypassRateReview` - Assign a value to bypass the code that rates a review, gets review detail information, updates read count, creates a list of reviews, and gets product detail information when `rate` has a value.
- `HSH_BypassLogin` - Assign a value to bypass the code that handles customer login, login verification, and emailing of forgotten passwords.
- `HSH_BypassLoginVerify` - Assign a value to bypass the code that verifies the customer is logged in when `HS_LoginForProductReviews` has been assigned a value of `Y` in the `/hostedstore/includes/HS_datafile.lasso` store configuration file.
- `HSH_BypassReviewFormInfo` - Assign a value to bypass the code that populates the `HSP_Product` variable with product information for the review form content page (`/hostedstore/storefront/html/review_form.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

- `HSH_Email_ReviewNotify` - Assign a value that will be used as the subject for the emailed posted review notification sent to `HS_StoreManagerEmail` when a customer posts a review and `HS_EmailNotifyReviews` has been set to `Y`. The value defaults to *A product review has been posted*
- `HSH_Email_LostPassword` - Assign a value that will be used as the subject for the email sent to customers when a lost password request is made. The value defaults to *Your password has arrived*.

Error variables:

- `HSH_Error_LoginFailure` - Assign a value that will be used as the error message returned when an attempt has been made to login with an invalid email address and password combination. The value defaults to *The email address and password provided do not match any of our customers*.
- `HSH_Error_LoginRequired` - Assign a value that will be used as the error message returned when login is required to post a review and the customer is not logged in the store. The value defaults to *Please login prior to posting a review*.
- `HSH_Error_LostPasswordFailure` - Assign a value that will be used as the error message returned when an attempt has been made to email a lost password with an invalid email address. The value defaults to *The email address provided does not match any of our customers*.
- `HSH_Error_LostPasswordSuccess` - Assign a value that will be used as the message returned when the password has been successfully retrieved and emailed. The value defaults to *Your password has been mailed to you*.
- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields*.
- `HSH_Error_ReviewAddSuccess` - Assign a value that will be used as the message returned when a review is successfully added. The value defaults to *Your review has been added*.

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `review_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `review_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `review_preapply.lasso` - Included before the code that posts the review information to the database if `HSH_PreApply` has a value.
- `review_postapply.lasso` - Included after the code that posts the review information to the database if `HSH_PostApply` has a value.
- `review_prelist.lasso` - Included before the code that populates the `HSP_ReviewList` array variable and the `HSP_Product` variable if `HSH_PreList` has a value.
- `review_postlist.lasso` - Included after the code that populates the `HSP_ReviewList` array variable and the `HSP_Product` variable if `HSH_PostList` has a value.
- `review_preread.lasso` - Included before the code that gets review detail information, updates read count, creates a list of reviews, and gets product detail information if `HSH_PreRead` has a value.
- `review_postread.lasso` - Included after the code that gets review detail information, updates read count, creates a list of reviews, and gets product detail information if `HSH_PostRead` has a value.
- `review_prerate.lasso` - Included before the code that rates a review, gets review detail information, updates read count, creates a list of reviews, and gets product detail information if `HSH_PreRate` has a value.
- `review_postrate.lasso` - Included after the code that rates a review, gets review detail information, updates read count, creates a list of reviews, and gets product detail information if `HSH_PostRate` has a value.

a value.

- *review_prelogin.lasso* - Included before the code that handles customer login, login verification, and emailing of forgotten passwords if `HSH_PreLogin` has a value.
- *review_postlogin.lasso* - Included after the code that handles customer login, login verification, and emailing of forgotten passwords if `HSH_PostLogin` has a value.
- *review_exit.lasso* - Included immediately prior to *storefront_exit.lasso* if `HSH_Exit` has a value.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page.

search.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=search.lasso`.

Variables

Hook point variables:

- `HSH_PreSearch` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/search_presearch.lasso` format file.
- `HSH_SearchFilter` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/search_filter.lasso` format file.
- `HSH_PostSearch` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/search_postsearch.lasso` format file.
- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/search_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassLinkBasedSearch` - Assign a value to bypass the code that converts the parameters passed through links generated from the `HSP_SearchCriteria` variable to parameters recognized by the search code block.
- `HSH_BypassBrowseProducts` - Assign a value to bypass the code that converts the parameters passed from the browse products navigation file (`/hostedstore/includes/navigation/browseproducts.lasso`) to parameters recognized by the search code block.
- `HSH_BypassSubcategorySearch` - Assign a value to bypass the code that converts the product category passed in the `show` parameter to a list of subcategories (the `HSP_Subcategories` array variable) for the `/hostedstore/storefront/html/search_subcategories.lasso` content page.
- `HSH_BypassQuickSearch` - Assign a value to bypass the code that converts the parameters passed from the quick search form to parameters recognized by the search code block.
- `HSH_BypassSearch` - Assign a value to bypass the code that searches for products based on search criteria from the advanced search form (`/hostedstore/storefront/html/search_form.lasso`), from the browse products navigation file (`/hostedstore/includes/navigation/browseproducts.lasso`), from the quick search form, and from link based searching.
- `HSH_BypassSearchFilter` - Assign a value to bypass the code that validates records returned from the search and populates the `HSP_ProductList` array variable.
- `HSH_BypassPreserveCriteria` - Assign a value to bypass the code that sets the `HSP_SearchCriteria` variable used to preserve search criteria and return to the search results list from the product detail content page (`/hostedstore/storefront/html/product_detail.lasso`) and from the shopping cart content page (`/hostedstore/storefront/html/cart_display.lasso`).
- `HSH_BypassCatalogSize` - Assign a value to bypass the code that sets the `HSP_CatalogSize` variable used on the default search form content page (`/hostedstore/storefront/html/search_form.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

None at this time.

Error variables:

- `HSH_Error_NoMatch` - Assign a value that will be used as the error message returned when a no

products met the search criteria. The value defaults to *Sorry, no products met the search criteria. Please try again.*

- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please specify at least one criteria for your search.*

Hook Points

All of the hook point format files are located in the `/hostedstore/thirdparty/hooks/storefront/` directory. Each file name is prefixed with `search_`. The hooks are listed below in the order in which the hook is called by HostedStore.

- `storefront_entry.lasso` - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with `HS_`), initialization of internal variables (variables prefixed with `HSI_`) and initialization of hook variables (variables prefixed with `HSH_`).
- `search_entry.lasso` - Included immediately after `storefront_entry.lasso`.
- `search_presearch.lasso` - Included immediately before the code block containing the product search inline if `HSH_PreSearch` has a value.
- `search_filter.lasso` - Included between the `Records` and `/Records` tags for the product search inline if `HSH_SearchFilter` has a value.
- `search_postsearch.lasso` - Included immediately after the code block containing the product search inline if `HSH_PostSearch` has a value.
- `search_exit.lasso` - Included immediately prior to `storefront_exit.lasso` if `HSH_Exit` has a value.
- `storefront_exit.lasso` - Included prior to setting and returning the store template and content page.

wishlist.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=wishlist.lasso`.

Variables

Hook point variables:

- `HSH_Exit` - Assign a value to include the `/hostedstore/thirdparty/hooks/storefront/wishlist_exit.lasso` format file.

Code bypass variables:

- `HSH_BypassLogin` - Assign a value to bypass the code that handles customer login, login verification, and emailing of forgotten passwords.
- `HSH_BypassWishlistList` - Assign a value to bypass the code that sets the `HSP_WishList` array variable for the default wishlist search results content page (`/hostedstore/storefront/html/wishlist_list.lasso`).
- `HSH_BypassAddToCart` - Assign a value to bypass the code that adds a wishlist item to the cart.
- `HSH_BypassAddToWishlist` - Assign a value to bypass the code that adds an item to the wishlist.
- `HSH_BypassWishlistDetail` - Assign a value to bypass the code that sets the `HSP_WishList` array variable for the default wishlist detail content page (`/hostedstore/storefront/html/wishlist_detail.lasso`).
- `HSH_BypassStaticCart` - Assign a value to bypass the code that sets the `HSP_CartItemCount` and `HSP_CartSubTotal` variables used for display of the static cart.

Email variables:

- `HSH_Email_LostPassword` - Assign a value that will be used as the subject for the email sent to customers when a lost password request is made. The value defaults to *Your password has arrived*.

Error variables:

- `HSH_Error_LoginFailure` - Assign a value that will be used as the error message returned when an attempt has been made to login with an invalid email address and password combination. The value defaults to *The email address and password provided do not match any of our customers*.
- `HSH_Error_LoginRequired` - Assign a value that will be used as the error message returned when a customer is not logged in the store and attempts to add an item to their wish list. The value defaults to *Please login prior to adding items to your wishlist*.
- `HSH_Error_LoginRequiredFields` - Assign a value that will be used as the error message returned when a required field for login is left blank. The value defaults to *Email address and password are necessary to login*.
- `HSH_Error_LostPasswordFailure` - Assign a value that will be used as the error message returned when an attempt has been made to email a lost password with an invalid email address. The value defaults to *The email address and password provided do not match any of our customers*.
- `HSH_Error_LostPasswordSuccess` - Assign a value that will be used as the message returned when the password has been successfully retrieved and emailed. The value defaults to *Your password has been mailed to you*.
- `HSH_Error_ProductAddFailure` - Assign a value that will be used as the error message returned

when a database or permissions error occurs while attempting to add a wish list item to the cart. The value defaults to *There was a problem adding the item to your cart. Please try again.*

- **HSH_Error_ProductAddSuccess** - Assign a value that will be used as the message returned when a wish list item is successfully added to the cart. The value defaults to *\$HSI_ProductName has been added to your cart.*
- **HSH_Error_ProductAttributesChanged** - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart for which the attributes have changed. The value defaults to *Product attributes have changed since the item was added to the wishlist. Please make modifications below.*
- **HSH_Error_ProductNotAvailable** - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart that is not currently offered. The value defaults to *That product is currently not for sale.*
- **HSH_Error_ProductNotFound** - Assign a value that will be used as the error message returned when an error occurred while attempting to add a wish list item to the cart that is not in the product catalog. The value defaults to *That product could not be found in our catalog or is currently not for sale.*
- **HSH_Error_RequiredAttributes** - Assign a value that will be used as the error message returned when a required attribute is left blank. The value defaults to *Please make a selection for each option below before adding the item to your wish list.*
- **HSH_Error_SearchNoMatch** - Assign a value that will be used as the error message returned when no wish lists met the search criteria. The value defaults to *Sorry, there are no wish lists that meet your criteria. The wish list you are searching for may not have been marked for public searching by the list owner. You may want to contact that individual for a direct link to their wish list.*
- **HSH_Error_SearchRequiredFields** - Assign a value that will be used as the error message returned when a required field for wish list search is left blank. The value defaults to *Please specify at least one criteria for your search.*
- **HSH_Error_WishlistAddFailure** - Assign a value that will be used as the error message returned when a database or permissions error occurs while attempting to add an item to the wish list. The value defaults to *There was a problem adding \$HSI_ProductName to your wish list. Please try again.*
- **HSH_Error_WishlistAddSuccess** - Assign a value that will be used as the message returned when an item is successfully added to the wish list. The value defaults to *\$HSI_ProductName has been added to your wish list.*

Hook Points

All of the hook point format files are located in the */hostedstore/thirdparty/hooks/storefront/* directory. Each file name is prefixed with *wishlist_*. The hooks are listed below in the order in which the hook is called by HostedStore.

- *storefront_entry.lasso* - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with **HS_**), initialization of internal variables (variables prefixed with **HSI_**) and initialization of hook variables (variables prefixed with **HS_**).
- *wishlist_entry.lasso* - Included immediately after *storefront_entry.lasso*.
- *wishlist_exit.lasso* - Included immediately prior to *storefront_exit.lasso* if **HSH_Exit** has a value.
- *storefront_exit.lasso* - Included prior to setting and returning the store template and content page.

Help Center Hooks

The details for each hook point, the location in the code, how inserted code may affect HostedStore, and the variable(s) to override HostedStore code are detailed below for each internal format file. All of the *Help Center* hook point format files are located in the */hostedstore/thirdparty/hooks/helpcenter/* directory.

Sections

- [customerservice.lasso](#)

customerservice.lasso

Accessed via a URL ending with `HostedStore.LassoApp?-ResponseLassoApp=helpcenter/customerservice.lasso`.

Variables

Hook point variables:

None at this time.

Code bypass variables:

None at this time.

Email variables:

- `HSH_Email_Inquiry` - Assign a value that will be used as the subject for the email sent to `HS_CustomerServiceEmail` when a customer submits a customer service inquiry. The value defaults to the *Customer Service Inquiry*.

Error variables:

- `HSH_Error_RequiredFields` - Assign a value that will be used as the error message returned when a required field is left blank. The value defaults to *Please complete all required fields*.

Cookies

Modification or use of the cookies used by HostedStore may be necessary. Each cookie is encrypted with the `HS_Encrypt` variable used as the key. To decrypt a cookie use the `Decrypt_Blowfish` LDML tag with the `HS_Encrypt` variable as the value for the `-Seed` parameter. To encrypt a value to reset a cookie use the `Encrypt_Blowfish` LDML tag with the `HS_Encrypt` variable as the value for the `-Seed` parameter. Example uses of each of these tags:

- Decrypting a value: `[Decrypt_Blowfish:VALUE,-Seed=$HS_Encrypt]`
- Encrypting a value: `[Encrypt_Blowfish:VALUE,-Seed=$HS_Encrypt]`

Cookies that hold multiple values will have each values separated by two adjacent pipe characters (`|` `|`). Each cookie is detailed below:

- `HS_Account` - Identifies a customer who has successfully logged in to the customer administration area or logged in during the checkout process. Two values are contained in the cookie. The position of each value is numbered below along with an explanation of the value:
 1. The key value for the customer record.
 2. The customer ID.
 3. The customer price group ID.
 4. The price group type.
 5. The price group value.
 6. `Y` if the **ApprovedForPO** field for the customer record has a value of `Y`.
- `HS_Admin` - Matches the value of the `HS_AdminLogin` variable set in the store preferences file the store administrator has successfully logged into the store administration area.
- `HS_AdminCustomerID` - The customer ID for the customer that is being used for the store administration order creation process. Note that this value is not encrypted like other cookies.
- `HS_AdminOrderID` - The order ID for the order that is being created in the store administration order creation process. Note that this value is not encrypted like other cookies.
- `HS_Affiliate` - Identifies an affiliate who has successfully logged in to the affiliate area. The cookie has one value, the affiliate ID.
- `HS_AID` - Tracks the affiliate ID that was passed in a referral hyperlink. The cookie has one value, the affiliate ID.
- `HS_Cart` - Tracks the order ID for the order and the values used to display the static shopping cart for the default store templates. Four values are contained in the cookie. The position of each value is numbered below along with an explanation of the value:
 1. The order ID for cart items and the order if an order has been created.
 2. The number of items in the cart.
 3. The subtotal of items in the cart formatted according to currency formatting preferences.
 4. The key value for the order record if an order has been created.
- `HS_Member` - Has a value of `Y` if a customer that has a current membership has logged in to the customer administration area or logged in during the checkout process.
- `HS_PID` - This cookie is used to track the promotion code that was passed in a hyperlink. The cookie has one value.

Creating HostedStore Modules

A list of things to keep in mind when creating HostedStore modules is provided below. The list is by no means exhaustive. If something is missing please let us know.

Store Configuration Variables

Some modules, especially those that generate content that is displayed within the store template, will require access to some of the variables set by the HS_datafile.lasso store configuration file. A module, HS_DecryptDataFile, is available that will decrypt the data file and make the variables available for use in third party modules. The module is available in the [download area](#) of the HostedStore Web site.

Keep in mind that this will require that the end user installs this module so it is recommended that proper documentation for the download and installation process is provided.

Displaying Content in the Storefront Template

If a module requires displaying content in the storefront template chosen by the end user, then the module will need to add the template files to the content page. Storefront templates are set in the HS_datafile.lasso store configuration file so the HS_DecryptDataFile module will be necessary to decrypt the file and set the store configuration variables.

The top store template file can be included with:

```
[Include:($HS_IncludeFilePathToStore+'/templates/'+$HS_TemplateName+'/top.lasso')]
```

The bottom store template file can be included with:

```
[Include:($HS_IncludeFilePathToStore+'/templates/'+$HS_TemplateName+'/bottom.lasso')]
```

Variable Naming Conventions

It is recommended that variables to be used on content pages are prefixed with a string value that identifies the variables as belonging to the module. This will help avoid the possibility of a conflict with variable names chosen by the end user or another module developer for use on the same content page.

It is highly recommended that variable names prefixed with **HS_**, **HSI_**, and **HSP_** are avoided so that there are not any complications in this or future versions of HostedStore.

Database Alterations

Fields can be added to the HostedStore database tables to support module functionality. It is recommended that a prefix is applied to the field name so that there is no chance that a future feature of HostedStore uses the same field name.

Important: Do not change any of the existing field or table names even if the module overrides or bypasses core HostedStore functionality.

Keep in mind that altering database tables will require the end user to add the new fields and update Lasso Administration. This process should be clearly documented to prevent any problems.

Module Installation Location

Two directories have been provided with a default installation of HostedStore.

- */hostedstore/thirdparty/modules/* - Intended for third party modules that add functionality that supplements or overrides HostedStore functionality or adds new functionality.
- */hostedstore/thirdparty/tags/* - Intended for third party modules that are not a suite of functionality, but provide simple added functionality. Despite the name, the module does not necessarily have to be a tag.

Feel free to use the existing directories as installation locations for modules. It is by no means necessary that modules are installed in the provided directories, but it will be a common convention recommended by us. It is recommended to use a prefix on all new directory names so that it does not conflict with possible directory names created by other third parties, end users, or future HostedStore directories.

Module Administration

A module administration menu is provided so that end users can access the module administration page from store administration. If a module has an administration interface, then document the modifications (text and appropriate link) that should be made to the module administration menu.

The menu content page is named *module_menu.lasso* and is located in the */hostedstore/adminstore/html/* directory.

Publishing Hook Modifications

If a module makes use of hooks, we recommend publishing a separate piece of documentation that details the hooks modifications for other developers. It is hoped that by having this documentation available that conflicts between third party modules can be avoided.

Register the Module

Be sure to make use of the module directory provided on the HostedStore Web site so that end users know about the module. Modules can be added in the [support area](#) of the HostedStore Web site.

Database Schema

The schema for the database tables is provided below. A few things may be readily apparent to experienced database administrators:

1. The database is not normalized. This is due to HostedStore initially being designed to support FileMaker Pro as a backend database. Hosting FileMaker Pro databases is expensive because only 50 databases can be hosted per license. Design decisions were made to limit the number of tables needed to accommodate FileMaker Pro users, including the ability to turn on/off certain HostedStore functions hence limiting the number of tables used. In the future a more normalized structure will be used. Meanwhile, the current structure should not be a concern because HostedStore still performs quickly and efficiently.
2. The choice of field types may not be optimal for how the user sets up their store. For example, passwords use a `TEXT` field. If passwords are limited to 15 characters by a `maxlength` value in the form input element, then that field could be a `VARCHAR(15)` type instead. There are indeed many difficulties in providing a generic off-the-shelf solution. If someone with an in-depth knowledge of databases is available, then by all means use that knowledge to tailor the databases to the needs of the store. Just make sure not to change the names of the fields because the field names provided below are expected by HostedStore. A mapping function to use other field names is being discussed as a future feature.
3. The fields may not be indexed in a manner that is optimized for how an individual store is used. As with the field types, if a knowledgeable person is available, then use that knowledge to fine tune the indexing of the databases. This will be of more importance for a large table containing thousands of products, customers, and/or orders.

Many of the fields repeat within a table. In cases like that, ellipses (...) have been placed between the first and last field in a series. The purpose of the field is detailed in the **Description** when it is not self evident.

Important: Do not change any of the field or table names. HostedStore expects the field and table names listed below to function properly. It is okay to add new fields to the tables below, but it is recommended that a prefix is applied to the field name so that there is no chance that a future feature of HostedStore uses the same field name.

Four tables are required for HostedStore to operate:

- [Customer](#) - Stores customer billing address and address book entries.
- [Invoice](#) - Stores shipping information, transaction processing information, and dates for orders.
- [Item](#) - Stores product information for the shopping cart and orders.
- [Product](#) - Stores product information.

Ten tables are used to support additional features:

- [AdminUser](#) - Stores store administration user information. **(Future Feature)**
- [Affiliate](#) - Stores affiliate information.
- [AutoShipOrderQueue](#) - Stores information for auto-generated orders.
- [EmailArchive](#) - Stores emails sent through *Store Administration*
- [GiftCertificate](#) - Stores purchased gift certificate balances. **(Future Feature)**
- [PriceGroup](#) - Stores price group information.
- [Promotion](#) - Stores promotion/coupon information.
- [Review](#) - Stores product reviews submitted by customers.
- [ShoppingList](#) - Stores product information for saved carts.
- [Supplier](#) - Stores supplier information for drop shipped products.
- [WishList](#) - Stores product information for active and purchased wish list items.

AdminUser

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential admin user ID.
Active	CHAR(1)	Current status of the user.
AdminGroup	TEXT	Group(s) to which the user belongs.
FirstName	TEXT	
LastName	TEXT	
Username	VARCHAR(80)	
Password	VARCHAR(80)	
Notes	TEXT	
Permissions	TEXT	
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.

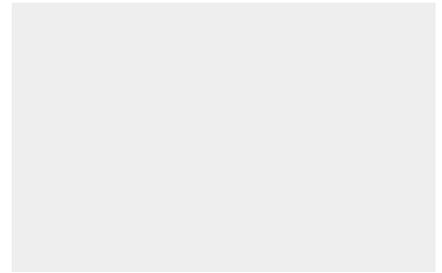
Affiliate

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential affiliate ID.
AffiliateID	TEXT	Affiliate ID used in referral hyperlinks.
Status	VARCHAR(9)	Current status of the affiliate.
FirstName	TEXT	Contact information.
LastName	TEXT	
Company	TEXT	
Address1	TEXT	
Address2	TEXT	
City	TEXT	
State	TEXT	
Postal	TEXT	
Country	TEXT	
Phone	TEXT	
Fax	TEXT	
Email	TEXT	
URL	TEXT	
Password	TEXT	
CompensationPlan	TEXT	The type of compensation received for each attributed order.
CompensationAmount	TEXT	The value for the type of compensation.
OrderQuantity	INT(11)	The number of orders attributed to the affiliate.
OrderTotal	DECIMAL(10,2) UNSIGNED	The monetary value of orders attributed to the affiliate.
AdminComments	TEXT	Comments about the affiliate made in store administration. Not visible by the affiliate.
Referrals	BIGINT(20)	The number of store visits attributed to affiliate hyperlinks.
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.

AutoShipOrderQueue

Field Name	MySQL	Description	
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential auto ship ID.	
CustomerID	BIGINT(20) UNSIGNED	The customer ID for the customer that placed the original order.	
OrderID	BIGINT(20) UNSIGNED	The order ID for the original order.	
DateOrdered	DATETIME	Date/time for the original order.	
Active	CHAR(1)	Y if the auto ship will be used to generate orders.	
DateActivated	DATETIME	Date/time the auto ship was last activated.	
DateDeactivated	DATETIME	Date/time the auto ship was last deactivated.	
Frequency	TINYINT(3)	Number of days to wait to generate an order.	
TimesExecuted	TINYINT(3)	Number of times the auto ship has been used to generate an order.	
RestrictedDuration	CHAR(1)	Y if the auto ship will be executed a fixed number of times.	
Duration	TINYINT(3)	The number of times to execute the auto ship order.	
OrderAction	TEXT	Signals what to do with the generated order.	
NextOrderDate	DATE	Date to execute the auto ship.	
OrderHistory	TEXT	List of all generated order invoice numbers and dates.	
Notes	TEXT		
AuditTrail	TEXT	List of changes made to the auto ship order.	
Item	TEXT	List of items to be auto shipped.	
FirstName	TEXT	Billing address information.	
LastName	TEXT		
Company	TEXT		
Address1	TEXT		
Address2	TEXT		
City	TEXT		
State	TEXT		
Postal	TEXT		
Country	TEXT		
Phone	TEXT		
Fax	TEXT		
Email	TEXT		
PaymentType	TEXT		Payment information.

PONumber	TEXT
Account	TEXT
Routing	TEXT
CheckNumber	INT(11)
CheckSignature	TEXT
Expiration	VARCHAR(7)
DateCreated	DATETIME
DateModified	DATETIME



Date/time the record was created.

Date/time the record was last modified.

Customer

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential customer ID.
OwnerID	BIGINT(20) UNSIGNED	The customer ID for the customer that owns the record. Used for address book entries.
TempCustomerID	VARCHAR(28)	Used to create a customer.
PriceGroupID	BIGINT(20) UNSIGNED	The price group ID to which the customer is assigned if <code>HS_UsePriceGroups</code> has a value of <code>Y</code> .
EntryName	TEXT	Name of the address book entry. Every customer has a master record for their billing address named <i>Billing</i> .
FirstName	TEXT	Billing address information.
LastName	TEXT	
Company	TEXT	
Address1	TEXT	
Address2	TEXT	
City	TEXT	
State	TEXT	
Postal	TEXT	
Country	TEXT	
Phone	TEXT	
Fax	TEXT	
Email	TEXT	
Password	TEXT	
Newsletter	CHAR(1)	
PublicWishList	CHAR(1)	<code>Y</code> if the wishlist is searchable by the public.
ApprovedForPO	CHAR(1)	<code>Y</code> if the customer can checkout with a purchase order.
Notes	TEXT	
EmailGroups	TEXT	Email group(s) to which the customer belongs.
DateCreated	DATETIME	Date the record was created.
DateModified	DATETIME	Date and time the record was last modified.
MembershipCreated	DATETIME	Membership creation date.
MembershipExpires	DATETIME	Membership expiration date.

EmailArchive

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential auto ship ID.
Type	CHAR(2)	
EmailGroup	TEXT	
CustomerID	BIGINT(20) UNSIGNED DEFAULT NULL	
HeaderTo	TEXT	
HeaderCC	TEXT	
HeaderBCC	TEXT	
HeaderFrom	TEXT	
HeaderSubject	TEXT	
Body	TEXT	
Message	TEXT	
ReadByIDs	TEXT	
SentByID	BIGINT(20) UNSIGNED	
DateCreated	DATETIME	Date/time the record was created.
DateModified	DATETIME	Date/time the record was last modified.

GiftCertificate

(Future Feature)

Invoice

Field Name	MySQL	Description	
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential order ID.	
TempID	VARCHAR(28)		
AutoShipOrderQueueID	BIGINT(20)	ID of the record in the AutoShipOrderQueue table that generated the order.	
OrderID	VARCHAR(28)	Unique value generated by the Lasso_UniqueID LDML tag.	
CustomerID	BIGINT(20) UNSIGNED	Customer ID for the customer purchasing the ordered items.	
AdminUserID	BIGINT(20) UNSIGNED	ID for the user that manually created the order in <i>Order Administration</i> .	
AffiliateID	TEXT	Affiliate ID of the affiliate referring the customer.	
Source	TEXT	The source of the order: Web, Retail, Mail Order, etc.	
FirstName	TEXT	Shipping address information.	
LastName	TEXT		
Company	TEXT		
Address1	TEXT		
Address2	TEXT		
City	TEXT		
State	TEXT		
Postal	TEXT		
Country	TEXT		
Phone	TEXT		
Fax	TEXT		
Email	TEXT		
ItemCount	INT(11)		Total number of items in the order.
SubTotal	DECIMAL(10,2) UNSIGNED		Subtotal for the order.
TaxLevel1	DECIMAL(10,2) UNSIGNED		
TaxLevel2	DECIMAL(10,2) UNSIGNED		
TaxLevel3	DECIMAL(10,2) UNSIGNED		
Tax	DECIMAL(10,2) UNSIGNED	Tax cost for the order.	
Shipping	DECIMAL(10,2) UNSIGNED	Shipping cost for the order.	
Total	DECIMAL(10,2) UNSIGNED	Total cost of the order (subtotal + tax + shipping - discount + adjustments).	
PaymentType	TEXT	Payment method used.	
PONumber	TEXT	Purchase order number used.	
Account	TEXT	Checking or credit card account number used. Encrypted.	
Routing	TEXT	Bank routing number for checks.	
CheckNumber	INT(11)	Check number of the check used.	

CheckSignature	Text	Typed signature for check.
Expiration	VARCHAR(7)	Credit card expiration date in the form MMYYYY.
Approved	CHAR(1)	Y if the transaction was approved by the payment gateway or manually in store administration.
ApprovalCode	TEXT	Approval code returned by the payment gateway.
ShippingVendor	TEXT	Shipping vendor used. Only applied when the entire order is shipped at once with the same vendor.
TrackingNumber	TEXT	Package tracking number assigned by the shipping vendor. Only applied when the entire order is shipped at once with the same vendor.
Status	TEXT	Current status of the order.
TransactionID	TEXT	Transaction ID returned by the payment gateway.
AVSResult	TEXT	AVS result code returned by the payment gateway.
ApprovalMessage	TEXT	Approval message returned by the payment gateway.
AuthorizationCode	TEXT	Authorization code returned by the payment gateway.
AdminPaymentComments	TEXT	<i>For future use.</i>
AdminComments	TEXT	Comments about the order made in store administration. Not visible by the customer. Encrypted.
CustomerComments	TEXT	Comments about the order made by the customer during the checkout process.
PhysicalPossession	VARCHAR(4)	<i>NOT USED</i>
CancelReason	TEXT	Reason for canceling an order provided by the customer.
PromotionCode	TEXT	Promotion code used during checkout.
Discount	DECIMAL(10,2)	Discount applied to the order by a valid promotion code.
SubTotalAdjustment	DECIMAL(10,2)	Adjustment to the subtotal made in store administration.
ShippingAdjustment	DECIMAL(10,2)	Adjustment to the shipping cost made in store administration.
TaxAdjustment	DECIMAL(10,2)	Adjustment to the tax cost made in store administration.
DiscountAdjustment	DECIMAL(10,2)	Adjustment to the discount made in store administration.
SettlementAdjustment	DECIMAL(10,2)	Adjustment to the total made in store administration when settling batches against an external source.
AffiliateAdjustment	DECIMAL(10,2)	Adjustment to the the theoretical affiliate payout amount.
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.
DatePosted	DATETIME	Date and time the order was completed by the customer.
DateProcessed	DATETIME	Date and time the order was processed.

DateBackOrdered	DATETIME	Date and time the order was globally backordered.
DateShipped	DATETIME	Date and time the order was completely shipped (either globally or by item).
DateCanceled	DATETIME	Date and time the order was canceled.
DateDeclined	DATETIME	Date and time the order was declined.
DateApproved	DATETIME	Date and time the order was approved.
DateCanceledPending	DATETIME	Date and time the order was canceled by the customer.
DateSettled	DATETIME	Date and time the order was batch settled.
SettlementBatch	TEXT	The batch in which the order was settled.
IPAddress	VARCHAR(15)	IP address of the customer placing the order.
AffiliateTheoreticalPayout	TEXT	Tracks the current affiliate plan when the order was shipped.
AffiliatePlan	DECIMAL(10,2)	Tracks the theoretical affiliate payout based on the current affiliate plan when the order was shipped.
AffiliatePayout	TEXT	Tracks if an affiliate payout was issued for the invoice.
AutoshipGenerated	CHAR(1)	Y if the order was created from an auto-ship.
TermsAndConditions	CHAR(1)	Y if the customer has agreed to the store terms and conditions.

Item

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential item ID.
OrderID	VARCHAR(28)	Order ID for the order to which the item belongs.
OrderStatus	VARCHAR(20)	Current status of the entire order.
CustomerID	BIGINT(20) UNSIGNED	Customer ID of the customer that ordered the item.
WishListOwnerID	BIGINT(20) UNSIGNED	Customer ID of the customer that had the item in their wishlist. Only applies if the item was purchased from a wishlist.
CartName	VARCHAR(80)	Name assigned by customer when saving a cart.
CartEmail	VARCHAR(80)	Email address assigned by customer when saving a cart.
CartExpires	Date	Date that the saved cart expires if <code>HS_SavedCartDays</code> has a value of <code>Y</code> .
ProductID	TEXT	Product ID for the item from the product table.
ShadowProducts	TEXT	Product IDs for items that should have their inventory deprecated along with the product.
ItemType	TEXT	
ProductName	TEXT	Product name for the item from the product table.
EnforceInventory	CHAR(1)	<code>Y</code> if inventory for the product should be verified during checkout.
Quantity	INT(11) UNSIGNED	Quantity ordered.
Price	DECIMAL(10,2)	Price of the item (adjusted by attributes).
TotalPrice	DECIMAL(10,2)	Quantity times the price.
Taxable	CHAR(1)	<code>Y</code> if the item is subject to taxation.
TaxClass	TEXT	
ShippingType	VARCHAR(20)	Type of shipping cost calculation applied to the item.
ShippingValue	DECIMAL(10,2)	The values to use in the shipping calculation.
Attributes	TINYINT(4)	Number of attributes.
Attribute1	TEXT	Selected configuration info for the nth attribute.
...		
Attribute30	TEXT	
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.
DateShipped	DATETIME	Date and time the item was shipped.
DateCanceled	DATETIME	Date and time the item was canceled.
Cost	DECIMAL(10,2)	Total merchant product cost at time of purchase.
CostDetail	TEXT	Itemized merchant costs for the product and component products.
ShippingVendor	TEXT	Shipping vendor used to ship the item.
TrackingNumber	TEXT	Package tracking number supplied by the shipping vendor.

DateBackOrdered	DATETIME	Date and time the item was backordered.
DateProcessed	DATETIME	Date and time the item was processed.
IsMembership	CHAR(1)	Y if the item is a membership.
MemberPlanDays	INT(11)	The number of days in the membership plan (only applies if the item is a membership).
MemberOnlyItem	CHAR(1)	Y if the item is available to members only.
UseEAD	CHAR(1)	Y if the item is distributed by download.
EADContentType	TEXT	The MIME type of the file that can be downloaded with item purchase.
EADFileNames	VARCHAR(120)	The name of the file that can be downloaded with item purchase.
EADDownloadLimit	INT(11)	The total number of times the files can be downloaded.
EADTimeLimit	INT(11)	The maximum number of days the customer has to download the files.
EADTimesDownloaded	INT(11)	The number of times the files have been downloaded.
Restricted	CHAR(1)	Y if the item can only be purchased if another item is also purchased.
SupplierID	VARCHAR(23)	Tracks the supplier responsible for drop shipping the item.

PriceGroup

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential admin user ID.
Active	CHAR(1)	Current status of the user.
Name	VARCHAR(80)	
Type	CHAR(1)	
Value	DECIMAL(8,5)	
TaxExempt	CHAR(1)	
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.

Product

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential product ID.
ProductID	TEXT	Product ID assigned in store administration.
ShadowProducts	TEXT	Product IDs for products that should have their inventory deprecated along with the product.
PageTitle	TEXT	
MetaKeywords	TEXT	
MetaDescription	TEXT	
ProductName	TEXT	Name of the product visible to customer.
WebSell	CHAR(1)	Y if the product is published in the store catalog.
Featured	CHAR(1)	Y if the product is featured on the store home page.
IsMembership	CHAR(1)	Y if the product is a membership.
MemberPlanDays	INT(11)	The number of days the membership is valid. Only applies if the item is a membership.
MemberOnlyItem	CHAR(1)	Y if the item is available to members only.
BaseCost	DECIMAL(10,2)	The base cost for the item from the supplier.
MSRP	DECIMAL(10,2)	Suggested retail price.
Price	DECIMAL(10,2)	Price at which the product is normally sold.
MemberPrice	DECIMAL(10,2)	Price at which the product is sold to members.
SalePrice	DECIMAL(10,2)	Price at which the product is sold if on sale.
OnSale	CHAR(1)	Y if the product is on sale.
VolumePrice	TEXT	Quantity range rules and prices for volume discounting.
OnHand	BIGINT(20)	Quantity in inventory.
BuildTo	BIGINT(20)	Quantity to which the inventory should be built.
WebSellQty	INT(11)	Minimum quantity that must be in inventory to sell the item in the store.
EnforceInventory	CHAR(1)	Y if inventory for the product should be verified during checkout.
OutOfStockNotify	CHAR(1)	Y if an email notification should be sent to the HS_OutOfStockEmail address when the OnHand value reaches the WebSellQty threshold value due to store sales activity.
OutOfStockDisplay	CHAR(1)	Y if the product should be displayed, but not sold, if out of stock.
OutOfStockMessage	TEXT	Message to display if the product is out of stock.
QuantitySold	BIGINT(20)	Total quantity sold historically.
Department	TEXT	Categorization for the product.
Category	TEXT	
SubCategory	TEXT	
SubCategory2	TEXT	
SubCategory3	TEXT	

Manufacturer	TEXT	
SearchRank	TINYINT(2)	
BriefHasHTML	CHAR(1)	Y if the brief description contains HTML tags.
BriefDescription	TEXT	Brief description displayed on product search results content page.
DescHasHTML	CHAR(1)	Y if the full description contains HTML tags.
Description	TEXT	The full description displayed on the product detail content page.
Complement1	TEXT	Product IDs for complimentary products.
...		
Complement6	TEXT	
Taxable	CHAR(1)	Y if the product is subject to taxation.
TaxClass	TEXT	
ShippingType	TEXT	Type of shipping cost calculation applied to the product.
ShippingValue	DECIMAL(10,2)	The value to use in the shipping calculation.
QuantitySearched	BIGINT(20) UNSIGNED	Number of times the product has been viewed on the search results content page.
QuantityViewed	BIGINT(20) UNSIGNED	Number of times the product has been viewed on the product detail content page.
QuantityInCart	BIGINT(20) UNSIGNED	Number of times the product has been added to a shopping cart.
Attributes	CHAR(2)	Number of attributes.
Attribute1Name	TEXT	Name of the nth attribute.
...		
Attribute30Name	TEXT	
Attribute1Values	TEXT	Values for the nth attribute.
...		
Attribute30Values	TEXT	
Attribute1Prices	TEXT	Price adjustments for the nth attribute.
...		
Attribute30Prices	TEXT	
Attribute1Description	TEXT	Description files and display type for the nth attribute.
...		
Attribute30Description	TEXT	
Attribute1PartNumbers	TEXT	Product IDs for the nth attribute.
...		
Attribute30PartNumbers	TEXT	
Attribute1PartQuantities	TEXT	Inventory adjustments for the nth attribute.
...		
Attribute30PartQuantities	TEXT	
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last created.
UseEAD	CHAR(1)	Y if the item is distributed by download.

EADContentType	TEXT	The MIME type of the file that can be downloaded with item purchase.
EADFileNames	VARCHAR(120)	The name of the file that can be downloaded with item purchase.
EADDownloadLimit	INT(11)	The total number of times the files can be downloaded.
EADTimeLimit	INT(11)	The maximum number of days the customer has to download the files.
PerOrderMin	INT(11)	The minimum quantity that must be purchased in an order.
PerOrderMax	INT(11)	The maximum quantity that can be purchased in an order.
PerCustomerMax	INT(11)	The maximum number of times a customer can order the product.
RestrictClass	TEXT	Restricted product class.
RestrictTo	TEXT	Class of product that must be in the cart before the product itself can be added to the cart.
PriceGroup	TEXT	Price group(s) to which a customer must belong for the product to show up in product search results.
IsGiftCertificate	INT(11)	Has a value of Y if the product is a gift certificate.
GiftCertificateUse	CHAR(3)	Stipulates how the gift certificate can be applied.
GiftCertificateValue	DECIMAL(10,2)	Face value of the gift certificate.
WaitList	TEXT	Store double pipe delimited email addresses of customers who would like to be notified when the product is back in stock.
ContentPage	TEXT	Specifies the content page to return instead of the default product detail content page.
SupplierID	BIGINT(20)	ID for the supplier from the Supplier that supplies the product.
SupplierNotify	CHAR(1)	Y if the supplier will drop ship the product.

Promotion

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential promotion ID.
PromotionCode	TEXT	Promotion code used by the customer.
Active	CHAR(1)	Y if the promotion is available for use.
BeginDate	DATETIME	Date when the promotion begins.
EndDate	DATETIME	Date when the promotion ends.
PromotionName	TEXT	Promotion identification fields for internal use. Not displayed to the customer.
Description	TEXT	
ApplyTo	TEXT	The order total to which the discount is applied.
MatchPaymentMethod	TEXT	Required payment tupe for the promotion to be applied.
MatchCustomerID	TEXT	Required customer ID for the promotion to be applied.
MatchProductID	TEXT	Required product ID for the promotion to be applied.
MinItems	INT(11)	Minimum number of ordered items for the promotion to be applied.
MinShipping	DECIMAL(10,2)	Minimum shipping cost for the promotion to be applied.
MinSubTotal	DECIMAL(10,2)	Minimum subtotal for the promotion to be applied.
Type	TEXT	Type of discount calculation to use.
Value	DECIMAL(10,2)	Value to use in the discount calculation.
PerCustomerLimit	INT(11)	Maximum number of times a customer can use the promotion code.
MaximumDiscount	DECIMAL(10,2)	Maximum discount that can be applied with teh promotion code.
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.

Review

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential review ID.
ProductID	TEXT	Product ID from the product table.
CustomerID	BIGINT(20) UNSIGNED	Customer ID of the customer reviewing the item.
Author	TEXT	Name of reviewer.
Rating	CHAR(2)	Rating given by reviewer.
ReviewText	TEXT	The review.
TimesRead	BIGINT(20)	Number of times the review has been viewed.
TimesHelpful	BIGINT(20)	Number of times the review has been rated as helpful.
DateCreated	DATETIME	Date and time the record was created.
Publish	CHAR(1)	Y if the review is to be published in the store.

ShoppingList

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential shopping list ID.
ListName	VARCHAR(80)	Name of the shopping list assigned by the customer.
CustomerID	BIGINT(20) UNSIGNED	Customer ID for the customer that owns the shopping list.
ProductID	TEXT	Product ID from the product table.
ProductName	TEXT	Product name from the product table.
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.
Quantity	INT(11)	Quantity of the item in the list.
Attributes	CHAR(2)	Number of attributes.
Attribute1	TEXT	Selected configuration info for the nth attribute.
...		
Attribute30	TEXT	

Supplier

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential supplier ID.
Status	CHAR(1)	Y if the supplier can drop ship orders.
FirstName	TEXT	Contact information.
LastName	TEXT	
Company	TEXT	
Address1	TEXT	
Address2	TEXT	
City	TEXT	
State	TEXT	
Postal	TEXT	
Country	TEXT	
Phone	TEXT	
Fax	TEXT	
Email	TEXT	
URL	TEXT	
Password	TEXT	
AllowLogin	CHAR(1)	Y if the supplier may login to <i>Supplier Account Administration</i> .
NotifyEmail	TEXT	
NotifyEmailTemplate	TEXT	
SupplierAdminTemplate	TEXT	
CustomerEmailTemplate	TEXT	
StoreEmailTemplate	TEXT	
StoreEmailAddress	TEXT	
OrderQuantity	INT(11)	
OrderTotal	DECIMAL(15,2)	
AdminComments	TEXT	
DateCreated	DATETIME	Date the record was created.
DateModified	DATETIME	Date and time the record was last modified.

Wishlist

Field Name	MySQL	Description
ID	BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY	Unique sequential wishlist ID.
CustomerID	BIGINT(20) UNSIGNED	Customer ID for the customer that owns the wishlist.
ProductID	TEXT	Product ID from the product table.
ProductName	TEXT	Product name from the product table.
PurchaserID	BIGINT(20) UNSIGNED	Customer ID for the customer that purchased the item.
DateCreated	DATETIME	Date and time the record was created.
DateModified	DATETIME	Date and time the record was last modified.
Attributes	CHAR(2)	Number of attributes.
Attribute1	TEXT	Selected configuration info for the nth attribute.
...		
Attribute30	TEXT	

HostedStore Change Log

Dates and details for changes to HostedStore are provided in reverse chronological order. The **Completed** column is the change first appeared. The **Type** column will be either *Bug*, *Comment*, *Documentation*, *New*, *Request*, or *Workaround*. If the change is the result of a modification request, then the type will appear as a hyperlink to the original modification request. The **Description** column provides details of the changes.

A complete change log (including changes that have been implemented, but not released yet) can be viewed in the support area of the HostedStore Web site (www.hostedstore.com/support/).

Sections

- [Version 2003.06.16](#)
- [Version 2002.12.01](#)
- [Version 2002.11.15](#)
- [Version 2002.07.15](#)
- [Version 2002.07.01](#)
- [Version 2002.06.17](#)
- [Version 2002.05.27](#)
- [Version 2002.05.20](#)
- [Version 2002.05.15](#)
- [Version 2002.04.15](#)

Version 2003.06.16

Completed	Version	Type	Description
2003.06.01	2003.06.16	Feature Request	All menus in Store Administration have been restructured to separate submit buttons into individual forms so that the return/enter key can be used to submit the form with the focus.
2003.06.01	2003.06.16	Feature Request	Report generated data can now be exported as a tab delimited file to the /hostedstore/adminstore/report/ directory by specifying a file (and path, if wanted) in the Export To field on the report menu.
2003.06.01	2003.06.16	Feature Request	Price groups can now be assigned to products. Customers that do not belong to an assigned price group will not have the product show up in their search results. This feature requires that a field, PriceGroup, is added to the Product table. The following SQL statement will add the field when submitted in a MySQL client: ALTER TABLE Product ADD COLUMN PriceGroup TEXT AFTER RestrictTo;
2003.06.01	2003.06.16	Feature Request	The Authorize.Net Advanced Integration Method (AIM) gateway is now supported.
2003.06.01	2003.06.16	Feature Request	The Verisign PayFlow Pro gateway is now supported on Lasso Professional 6.
2003.06.01	2003.06.16	Feature Request	Added support for Bil Corry's Get_Uploads tag for stores deployed on Lasso Professional 6 that use the Get_Uploads tag.
2003.06.01	2003.06.16	Feature Request	A very basic permissions system has been added to Store Administration that will allow for the creation of users that have access or are denied access to each area of Store Administration. The administration user created in Store Configuration will continue to have access to all areas of Store Administration. This feature lays the groundwork for a more comprehensive permissions system that will be implemented in the future.
2003.06.01	2003.06.16	New	The Store Administration template can now be modified. The Store Administration Template Name line under the Default Template heading on the Store Configuration form sets the the template to use. Templates should be placed in the /hostedstore/adminstore/templates/ directory in a directory that is named the same as the template name. Use HS_1 if the internal HostedStore Store Administration template should be used. Avoid naming custom templates with an HS_ prefix to prevent possible conflicts with future versions of HostedStore.
2003.05.31	2003.06.16	Comment	Removed a few instances of 5 pixel cell padding for tables in Store Administration and standardized on a 2 pixel cell padding.
2003.05.31	2003.06.16	Feature Request	Click/double-click sorting on column headers has been implemented in Store Administration search results lists for those users that think clicking twice to get to a descending sort order is more efficient.
2003.05.31	2003.06.16	Feature Request	If a customer is already logged in to the store (either through the Account Administration area or through the storefront prior to creating an order) it is no longer necessary to login again during checkout.

2003.05.31 2003.06.16	Feature Request	A Limit field has been added to the menus for Affiliate, Customer, Order, Product, and Promotion Administration. Set the limit to the number of records that are to be displayed per page. Next/Previous navigation has been added to the search results list to allow for paging through the found set.
2003.05.30 2003.06.16	Bug	<p>Rounding errors no longer occur when adding order subtotal, shipping, tax, and discount.</p> <p>This change will require changes to the tax calculation script (calculate_tax.lasso in the /hostedstore/includes/calculations/ directory). Because that is a user editable file the appropriate adjustments will need to be determined and applied by the user if the file has been altered.</p> <p>The necessary changes for the default tax calculation script provided with a HostedStore distribution are provided below:</p> <p>Modify line 101. From: \$HSP_CartTaxableTotal+=\$HSP_CartShippingCost; To: \$HSP_CartShippingCost->SetFormat:-Precision=\$HS_CurrencyPrecision; \$HSP_CartTaxableTotal+=(Decimal:(String:\$HSP_CartShippingCost));</p> <p>Insert at line 143: // Set display formatting. \$HSP_CartSubTotal->SetFormat:-Precision=\$HS_CurrencyPrecision; \$HSP_CartShippingCost->SetFormat:-Precision=\$HS_CurrencyPrecision; \$HSP_TempTax->SetFormat:-Precision=\$HS_CurrencyPrecision;</p> <p>Modify line 148. From: \$HSP_TempTotalWithTax=\$HSP_CartSubTotal +\$HSP_CartShippingCost +\$HSP_TempTax; To: \$HSP_TempTotalWithTax=(Decimal:(String:\$HSP_CartSubTotal)) +(Decimal:(String:\$HSP_CartShippingCost)) +(Decimal:(String:\$HSP_TempTax));</p>
2003.05.30 2003.06.16	Feature Request	Basic hook points have been added to Store Administration.
2003.05.30 2003.06.16	Feature Request	The Date Created field on the Order Administration search form has been replaced with a Begin Date and End Date field that will allow for date range searches.
2003.05.30 2003.06.16	Feature Request	<p>The default purchase order eligibility for a customer can now be set in Store Configuration.</p> <p>Set the HS_DefaultPOEligibility preference variable in Store Configuration with the Default Purchase Order Eligibility item under the Payment Methods Accepted heading. The value defaults to No ("N" for the HS_DefaultPOEligibility preference variable value).</p> <p>The HS_DefaultPOEligibility value will be used as the value for the ApprovedForPO field in the Customer table when a new customer account is created.</p>
2003.05.30 2003.06.16	Feature Request	Added "Mask Account Info" to the Maintenance Menu in Store Administration. Use of this feature will mask account and routing numbers by replacing all but the first and last four digits with Xs.
2003.05.29 2003.06.16	Bug	The HSP_SearchCriteria variable published to the product_detail.lasso content page in the /hostedstore/storefront/html/ directory is now URL encoded.

- 2003.05.29 2003.06.16 [Feature Request](#) Added attribute part quantities and part product IDs to the information published to the order detail page in Order Administration. This has a side effect of publishing the same information to the pick list, packing invoice and mailing invoice (the order_picklist.lasso, order_packinginvoice.lasso, order_mailinginvoice.lasso content pages in the /hostedstore/adminstore/html/ directory, respectively).
- \$HSP_Order->Get:15->Get:16 will access an array of arrays.
- 1: Array of part quantities arrays, where each element is an attribute. The order matches the order of the attributes accessed with \$HSP_Order->Get:15->Get:15.
 - 2: Array of part product ID arrays, where each element is an attribute. The order matches the order of the attributes accessed with \$HSP_Order->Get:15->Get:15.
- 2003.05.29 2003.06.16 [Feature Request](#) Four hook points were added to Order Administration in Store Administration.
- The following files should be added to the /hostedstore/thirdparty/hooks/adminstore/ directory:
- adminstore_entry.lasso
adminstore_exit.lasso
order_entry.lasso
order_exit.lasso
- Hook point variables:
- HSX_Exit - Assign a value to include the /hostedstore/thirdparty/hooks/adminstore/order_exit.lasso format file.
- Hook Points
- Each file name is prefixed with adminstore_. The hooks are listed below in the order in which the hook is called by HostedStore.
- adminstore_entry.lasso - Included after HostedStore runs its internal routines including serial number validation, setting of preference variables (variables prefixed with HS_), initialization of internal variables (variables prefixed with HSI_) and initialization of hook variables (variables prefixed with HSH_).
- order_entry.lasso - Included immediately after adminstore_entry.lasso.
- order_exit.lasso - Included immediately prior to adminstore_exit.lasso if HSH_Exit has a value.
- adminstore_exit.lasso - Included prior to setting and returning the store template and content page.
- 2003.05.29 2003.06.16 [Feature Request](#) The various prices and discounts used to determine the price(s) displayed on the product detail content page (product_detail.lasso in the /hostedstore/storefront/html/ directory) are now available as separate items in a map of arrays.
- The map of arrays can be accessed with \$HSP_Product->Get:5->Get:4.
- The array structure of the map values:
- 1: Price as a decimal type.
 - 2: Price formatted with HS_FormatCurrency according to currency formatting preferences set in Store Configuration.
 - 3: Price group type applied (if applicable). Possible values are: Discount (%), At Retail, Discount from Retail (%), At Cost, and Markup Cost (%).
 - 4: Discount (the difference between full price and the price) as a decimal type.
 - 5: Discount formatted with HS_FormatCurrency according to currency formatting preferences set in Store Configuration.
 - 6: Y/N flag that indicates whether or not the price was used as the final price that is displayed to the customer.
- Possible map keys: MSRP, Standard, Member, Sale, and Base Cost.
- This map can be used to create different price display text than what is provided in \$HSP_Product->Get:5->Get:3 or to display percentage discounts by using the decimal type

			prices in calculations.
2003.05.28	2003.06.16	Bug	The HSP_SearchCriteria variable published to the search_list.lasso content page in the /hostedstore/storefront/html/ directory is now URL encoded. The links created by the default browseproducts.lasso content page in the /hostedstore/includes/navigation/ directory are now URL encoded.
2003.05.28	2003.06.16	Bug	Duplicate billing accounts are no longer created if a customer refreshes their browser or resubmits the form during the account creation process. The check for an existing account is performed against the FirstName, LastName, Email, and Password fields of the Customer table.
2003.05.28	2003.06.16	Bug	All links to HostedStore_Reports.LassoApp now use /hostedstore/adminstore/HostedStore_Reports.LassoApp. This mainly affects help related functions of the LassoApp.
2003.05.28	2003.06.16	Bug	The public shipping destination variables (HSP_ShippingCity, HSP_ShippingState, HSP_ShippingPostal, and HSP_ShippingCountry) are now available to the shipping calculation script immediately after shipping destination information is provided by the customer.
2003.05.28	2003.06.16	Bug	An error is no longer generated when clicking the Packing Invoice hyperlink on the order detail page in Order Administration.
2003.05.28	2003.06.16	Bug	Shipping information is now submitted securely during the checkout process. Lines 24 and 26 of the shipping_info.lasso content page in the /hostedstore/storefront/html/ directory were changed to use \$HS_SecurePathToStore instead of \$HS_PathToStore in the form actions.
2003.05.28	2003.06.16	Bug	The Store Administration template no longer displays when generating a pick list, packing invoice, and mailing invoice in Order Administration.
2003.05.28	2003.06.16	Bug	An error no longer occurs when trying to email the customer from the order detail page in Order Administration if the customer_emailgroups.lasso file in the /hostedstore/adminstore/selectlists/ directory is blank.
2003.05.28	2003.06.16	Bug	It is now possible to clear an existing out of stock message for a product in Product Administration.
2003.05.28	2003.06.16	Bug	The proper price group for a customer is now selected on the customer edit form in the Customer Administration area.
2003.05.28	2003.06.16	Bug	Customers can now modify their billing address when a store does not have the address book feature enabled. Added a new content page: billingaddress.lasso in the /hostedstore/admincustomer/html/ directory to help prevent current users from having to heavily modify the address.lasso content page. Added two hook point error variables to admincustomer/admin.lasso: HSH_Error_BillingUpdateSuccess and HSH_Error_BillingUpdateFailure
2003.05.28	2003.06.16	Bug	Invalid syntax in the stylesheets for the HS_1, HS_2, HS_3, HS_4, HS_5, HS_6, and HS_7 templates has been fixed.
2003.05.28	2003.06.16	Bug	Invalid syntax in the Store Administration stylesheet has been fixed.
2003.05.28	2003.06.16	Bug	All hook point variables are now properly defined in the admincustomer/admin.lasso page.
2003.05.28	2003.06.16	Bug	Added preventcache=[Lasso_UniqueID] to the form actions on lines 61 and 63 of the /hostedstore/storefront/html/cart_display.lasso content page.
2003.05.28	2003.06.16	Bug	All hook variables are now properly defined in admincustomer/cancelorder.lasso.

2003.05.28 2003.06.16	Bug	An error is no longer displayed on the checkout page for returns in the order administration area when the customer belongs to a tax exempt price group.
2003.05.28 2003.06.16	Bug	Fixed internal link error so that adminstore/manualreturn_checkout.lasso would show up as the form action for the checkout button.
2003.05.28 2003.06.16	Feature Request	<p>Checkout can now be prevented until a customer has agreed to any terms and conditions that a store may provide.</p> <p>This feature requires multiple modifications:</p> <ol style="list-style-type: none"> 1) Set the HS_UseTermsAndConditions preference variable in Store Configuration with the Require Terms And Conditions item under the Cart/Checkout Page heading. The value defaults to No ("N" for the HS_UseTermsAndConditions preference variable value). 2) Add a TermsAndConditions column to the Invoice table by issuing the following SQL statement in a MySQL client: ALTER TABLE Invoice ADD COLUMN TermsAndConditions CHAR(1); 3) An HSH_Error_NoTermsAndConditions hook point error variable was added to checkout.lasso and modifypayment.lasso. Assign a value that will be used as the message returned when the terms and conditions checkbox has not been checked. The value defaults to "Please read and agree to the terms and conditions." 4) Modify the shipping_info.lasso and payment_modify.lasso content pages in the /hostedstore/storefront/html/ directory to take advantage of this feature. The variable used for terms and conditions is named f27. <p>The following code was inserted at line 314 of shipping_info.lasso and at line 160 of payment_modify.lasso:</p> <pre>[If:\$HS_UseTermsAndConditions=='Y'] <tr><td colspan="2" class="subtitle">Terms and Conditions</td></tr> <tr> <td>&nbsp;</td> <td align="left" class="med"> <input type="checkbox" name="f27" value="Y"[If:\$f27=='Y'] checked[/If]> I have read and agree to the terms and conditions. </td> </tr> <tr><td colspan="2"><hr></td></tr> [/If]</pre> <p>Variations of this code will likely be necessary to meet the specific needs of a store. For example, the "terms and conditions" text will probably need to be changed to a hyperlink that points to the page that details the terms and conditions of purchase.</p>
2003.05.28 2003.06.16	Feature Request	The Lasso password and store admin password fields on the Store Configuration form now use password form elements instead of text form elements.
2003.05.28 2003.06.16	Feature Request	<p>Added the PONumber field to the HSP_Order variable that is published to the email_customerapproval.lasso file in the /hostedstore/storefront/email directory.</p> <p>It can be accessed with \$HSP_Order->Get:2->Get:4.</p>
2003.05.28 2003.06.16	Feature Request	<p>Added an admincustomer/logout.lasso page that will clear the HS_Account and HS_Member cookies and then display the login form content page.</p> <p>A link was added to the default customer administration menu content page: the menu.lasso file in the /hostedstore/admincustomer/html/ directory.</p> <p>Also added two new hook points for this page: the logout_entry.lasso and logout_exit.lasso files in the /hostedstore/thirdparty/hooks/admincustomer/ directory.</p>
2003.05.17 2003.06.16	Bug	A MySQL sleeping connection is no longer created when adding an item to the shopping cart.

Version 2002.12.01

Completed	Version	Type	Description
2002.11.30	2002.12.01	New	<p>Tax classes have been implemented in Product Administration.</p> <p>In the simplest case, the field can be used to hold the name of a single tax class. Alternatively, more complex strings that contain multiple tax classes and tax jurisdictions can be stored for parsing in the tax calculation script (/hostedstore/includes/calculations/calculate_tax.lasso).</p>
2002.11.30	2002.12.01	Comment	<p>The process for obtaining past orders on the Store Administration entry page now processes quicker.</p>
2002.11.30	2002.12.01	Bug	<p>An error is no longer generated in Customer Administration when the email group file (/hostedstore/adminstore/selectlists/customer_emailgroups.lasso) is blank.</p>
2002.11.29	2002.12.01	New	<p>Shadow products can now be assigned to a product in Product Administration.</p>
2002.11.29	2002.12.01	New	<p>Product level shadow products have been implemented. This feature will allow for products to act as bundles of other products that may or may not be available for sale individually. The inventory for all shadow product IDs will be decremented by the specified amount.</p>
2002.11.28	2002.12.01	New	<p>Added a TaxClass field to the Item and Product tables to support taxing product types differently according to jurisdiction. The fields can be added with the following statements in the mysql client:</p> <pre>ALTER TABLE Item ADD COLUMN TaxClass TEXT AFTER Taxable; ALTER TABLE Product ADD COLUMN TaxClass TEXT AFTER Taxable;</pre>
2002.11.28	2002.12.01	New	<p>Added HSP_TaxInformation to the tax calculation script (/hostedstore/includes/calculations/calculate_tax.lasso). HSP_TaxInformation is an array of arrays that contains information about ordered items that may be needed for complex tax scenarios that require taxing specific classes of products in different jurisdictions.</p>

Version 2002.11.15

Completed	Version	Type	Description
2002.11.15	2002.11.15	New	The default browse products menu (/hostedstore/includes/navigation/browseproducts.lasso) has been changed to only display categories for which there are products flagged for selling in the store (WebSell field from the Product table has a value of Y). Note that the products may not actually be for sale (no inventory available).
2002.11.14	2002.11.15	Bug	Saved cart items currently in the shopping cart will no longer be deleted from the saved cart when the items are removed from the shopping cart.
2002.11.13	2002.11.15	New	The email a friend feature on the product detail content page and the wish list link email feature in Customer Account Administration can now be used in environments where the from or to email address must be for an account registered with the mail server.
2002.11.13	2002.11.15	Bug	The billing address address book entry can now be modified in Customer Account Administration.
2002.11.13	2002.11.15	Bug	Extraneous (Add \$) and (Subtract \$) text is no longer displayed for non-price adjusting attributes on the modify item content page.
2002.11.11	2002.11.15	Documentation	Comments can now be added to the online version of the documentation at <http://documentation.hostedstore.com>.
2002.11.09	2002.11.15	New	Product attributes can now have multiple product IDs and product quantities specified so that single attributes can adjust the inventory of multiple products. Each value should be delimited with a comma. This feature will allow for product packages to be specified as an attribute selection.
2002.11.09	2002.11.15	New	Columns have been added to the Invoice and Item tables to support a future shadow product feature. The columns can be added by executing the following statements in the mysql client: ALTER TABLE Item ADD COLUMN ShadowProducts TEXT AFTER ProductID; ALTER TABLE Product ADD COLUMN ShadowProducts TEXT AFTER ProductID;
2002.11.06	2002.11.15	New	Product complement fields now have support for multiple double pipe () delimited product IDs. The fields can now be used to rank or group product complements which in turn can be used to group accordingly on the product detail content page.
2002.11.06	2002.11.15	Bug	Membership dates now display properly in the edit customer select lists in Customer Administration.
2002.10.29	2002.11.15	Bug	The file path has been corrected for the customer service inquiry form in the help center so the form now loads properly.
2002.10.27	2002.11.15	Bug	The next order date for an autoship order is now calculated by adding the frequency to the current date instead of by adding the frequency to the next order date.
2002.10.26	2002.11.15	Bug	Non-price adjusting attributes that follow negative price adjusting attributes no longer display (Subtract \$).
2002.10.26	2002.11.15	Bug	An error is no longer reported during order checkout in store administration if the customer belongs to a tax exempt price group.
2002.10.26	2002.11.15	Bug	Attributes that adjust inventory of other products are now properly captured when creating an order in store administration.
2002.10.26	2002.11.15	Bug	Credit card expiration dates are now properly stored in mmyyyy format when creating orders in store administration and in orders created from autoship products.

2002.10.26	2002.11.15	Bug	Product description files now function properly in the manual order creation process of store administration.
2002.10.16	2002.11.15	Feature Request	<p>The ability to create autoship products is partially implemented. Autoship orders can be created as pending. A future version will allow for the automated processing of orders created from autoship products.</p> <p>Several settings have been added to store configuration to support autoship products. Autoship products are determined by attributes settings for the product.</p> <p>More details can be found in the documentation in Store Configuration and Product Administration sections.</p>
2002.08.28	2002.11.15	Bug	Windows drive file path is no longer included in the name of uploaded files.
2002.08.18	2002.11.15	New	<p>Added AdminUserID column to Invoice table for future use.</p> <p>The following text can be used in the mysql client to add the column to an existing Invoice table:</p> <pre>ALTER TABLE Invoice ADD COLUMN AdminUserID BIGINT(20) UNSIGNED DEFAULT NULL AFTER CustomerID;</pre>
2002.08.18	2002.11.15	New	<p>The template used for store administration is now open. The files for the template are located in the /hostedstore/adminstore/templates/ directory.</p> <p>NOTE: The template file has been opened due to what appear to be internal file count restrictions for LassoApps. If that restriction is lifted or if there is a suitable solution found, then the template may be closed in the future.</p>
2002.08.18	2002.11.15	New	<p>The reports in store administration have been built into an independent LassoApp named HostedStore_Reports.LassoApp. The recommended location for HostedStore_Reports.LassoApp is the /hostedstore/adminstore/ directory so it can be realm protected if that extra level of protection is wanted.</p> <p>NOTE: Reports have been built into their own LassoApp due to what appear to be internal file count restrictions for LassoApps. Splitting out reports was the most logical way to split the code while meeting the file limitations. If that restriction is lifted or if there is a suitable solution found, then the reports will be moved back into the main HostedStore LassoApp in the future.</p>
2002.08.18	2002.11.15	New	<p>Added column AutoShipOrderQueueID column to the Invoice table for future use.</p> <p>The column can be added with the following line via the mysql client:</p> <pre>ALTER TABLE Invoice ADD COLUMN AutoShipOrderQueueID BIGINT(20) UNSIGNED NOT NULL AFTER TempID;</pre>
2002.08.18	2002.11.15	New	<p>Added multiple columns to the AdminUser table to facilitate easier identification of users.</p> <p>The new columns can be added via the mysql client with the following lines:</p> <pre>ALTER TABLE AdminUser ADD COLUMN AdminGroup TEXT DEFAULT NULL AFTER Active; ALTER TABLE AdminUser ADD COLUMN FirstName TEXT DEFAULT NULL AFTER AdminGroup; ALTER TABLE AdminUser ADD COLUMN LastName TEXT DEFAULT NULL AFTER FirstName; ALTER TABLE AdminUser ADD COLUMN Notes TEXT DEFAULT NULL AFTER Password;</pre>
2002.08.17	2002.11.15	Bug	Edit product form in store administration no longer displays broken images when a thumbnail image of the product does not exist.

2002.08.17	2002.11.15	Bug	There is no longer a variable definition error reported after uploading files. (Note that 2002.08.17 was an internal release, not a public release.)
2002.08.08	2002.11.15	New	Added text column EmailGroups to Customer table to support future email groups feature. ALTER TABLE Customer ADD COLUMN EmailGroups TEXT AFTER ApprovedforPO;
2002.08.08	2002.11.15	Feature Request	Customers can now be assigned to email groups that can be used in the email customers area of store administration. Requires two changes: - Add EmailGroups text field to the Customer table. - List return delimited email groups in the /hostedstore/adminstore/selectlists/customer_emailgroups.lasso format file.
2002.08.08	2002.11.15	Feature Request	Customer emails from customer administration and order administration can now be archived. This change requires the addition of a new table. The table, EmailArchive, can be added via the mysql client with: DROP TABLE IF EXISTS EmailArchive; CREATE TABLE EmailArchive (ID bigint(20) unsigned NOT NULL auto_increment, Type char(2) default NULL, EmailGroup text, CustomerID bigint(20) unsigned default NULL, HeaderTo text, HeaderCC text, HeaderBCC text, HeaderFrom text, HeaderSubject text, Body text, Message text, ReadByIDs text, SentByID bigint(20) unsigned, DateCreated datetime default NULL, DateModified datetime default NULL, PRIMARY KEY (ID)) TYPE=MyISAM;
2002.08.05	2002.11.15	Comment	Product searching has been rewritten to work more efficiently with large found sets.
2002.08.05	2002.11.15	Comment	Cart code has been optimized to deal with configurable items more efficiently. Approximately 2 seconds of processing time was eliminated when all attributes are used and the product is misconfigured.
2002.08.04	2002.11.15	New	Added a tag HS_ArrayRandomSort to randomly sort an array.
2002.08.04	2002.11.15	New	Added a tag HS_GetFeaturedProducts that returns an array of featured products. The array can be randomly sorted.
2002.08.04	2002.11.15	New	Added a tag HS_GetFavoriteProducts that returns an array of top selling products.
2002.08.04	2002.11.15	Feature Request	Added a tag HS_GetRecommendedProducts that, given an array of products, will return an array of products that were ordered in conjunction with the given products. The array can be sorted randomly or by total quantity ordered.
2002.08.04	2002.11.15	Feature Request	Prices can now be adjusted by the same monetary amount or percentage. This option is available from the product administration menu.

2002.08.03	2002.11.15	Workaround	Changed default for the MemberOnlyItem field to "N" to workaround the LP5 bug (lack of recognition for NULL values).
2002.08.02	2002.11.15	Documentation	All documentation references to other sections in the documentation are now hyperlinked to aid in navigation.
2002.08.01	2002.11.15	New	Added fields TaxLevel1, TaxLevel2, and TaxLevel3 of type decimal(10,2) to the Invoice table for future use. The fields can be added with the following lines in the mysql client: alter table Invoice add column TaxLevel1 decimal(10,2) after SubTotal; alter table Invoice add column TaxLevel2 decimal(10,2) after TaxLevel1; alter table Invoice add column TaxLevel3 decimal(10,2) after TaxLevel2;
2002.07.31	2002.11.15	Documentation	Documentation has been restructured to allow for easier breaking out of content pages into separate pages. Also made changes that will allow constant width fonts used for code to appear properly in the PDF version of the documentation.
2002.07.29	2002.11.15	New	The theoretical affiliate payout is now calculated when an order is created. This change allows for the changing of an affiliate plan during a payout cycle and also provides a more accurate view of store liabilities when generating gross income reports. This change requires a field, AffiliateTheoreticalPayout, to be added to the Invoice table. The following line can be used in the mysql client: alter table Invoice add column AffiliateTheoreticalPayout decimal(10,2) after IPAddress;
2002.07.29	2002.11.15	New	A field, AffiliatePlan, has been added to Invoice database to track the affiliate plan in place at the time an order is generated. The following line can be used in the mysql client to add the field to an existing table: alter table Invoice add column AffiliatePlan text after AffiliateTheoreticalPayout;
2002.07.29	2002.11.15	Feature Request	All reports have been rewritten to generate in a more efficient manner.
2002.07.29	2002.11.15	Comment	The reports in the "Sales Reports" category have been retitled "Orders Reports." Orders reports are only generated for orders that have been completed or have the possibility of being completed, i.e., the status of the order is not In Cart, Abandoned, Declined, or Canceled.
2002.07.28	2002.11.15	New	Several new date dependent gross income reports have been added that incorporate all product costs, affiliate payouts, and adjustments to yield a final monetary value for all transaction during the time period. Gross income reports are only calculated for orders that have been processed (have status of Processed, Back Ordered, Shipped, or Mixed Status).
2002.07.26	2002.11.15	New	A date dependent drop ship pending report has been added that summarizes product name, quantity, subtotal, cost, and net revenue by supplier and invoice for orders that have been processed but have not shipped.
2002.07.26	2002.11.15	Feature Request	The item quantity can now be set when adding an item to the cart by passing the quantity as a value for a parameter named CartAddQty. Defaults to 1 if not specified.
2002.07.26	2002.11.15	Feature Request	A date dependent drop ship report has been added that summarizes subtotal, cost, and net revenue by supplier and invoice.
2002.07.25	2002.11.15	Feature Request	A default price group can now be set by specifying the name of the price group in store configuration. Note that the price group pricing will not be applied to any products that are already in the customer's cart.
2002.07.25	2002.11.15	Bug	The quick search form no longer returns all products when a member only item restriction is used.
2002.07.24	2002.11.15	New	Added "Local Delivery" shipping option to edit order form in order administration. The tracking number field can be used to store the driver's name if necessary.

2002.07.24	2002.11.15	New	Drop shipping is now supported. More details are provided in the documentation, but here's an overview: - Products can be assigned a supplier that will be notified when the order is processed (ready for shipment). - In order administration, when an order is assigned a status of "Processed," then all relevant drop ship suppliers are notified. - The drop ship email notice can be customized for each supplier. - Suppliers are provided with an interface similar to a scaled down order administration where orders or individual ordered items can be marked as shipped or back ordered. - Supplier order administration interface can be customized for each supplier. - The customer status change notice can be customized for each supplier.
2002.07.23	2002.11.15	Bug	Promotions are now properly applied when a maximum discount value is not set.
2002.07.23	2002.11.15	Bug	Fixed a problem that would cause new accounts to not be associated with their order in some cases.
2002.07.22	2002.11.15	New	Added "Canceled (Pending)" to the status field on the order search form in order administration.
2002.07.20	2002.11.15	Feature Request	Error messages set internally for the storefront, customer account, and affiliate account areas of the store can now be modified via hook variables. Consult the documentation for full details.
2002.07.20	2002.11.15	Documentation	Fixed errors in admincustomer hooks documentation. The global files included are admincustomer_entry.lasso and admincustomer_exit.lasso, not storefront_entry.lasso and storefront_exit.lasso.
2002.07.20	2002.11.15	Bug	Added missing admincustomer hook files to the /hostedstore/thirdparty/hooks/admincustomer/ directory. Note: Absence of these files will not affect the functionality of HostedStore. The empty files have just been added as a convenience.
2002.07.19	2002.11.15	New	"Help" links have been added to store configuration next to each section. Clicking the link will open a new browser window with the section of the documentation that pertains to that topic.
2002.07.19	2002.11.15	New	Added a supplier administration section to store administration. Suppliers can be created and assigned to individual products in product administration. An interface is provided to email individual suppliers or all suppliers.
2002.07.19	2002.11.15	New	"Help" links have been added next to each section on the add/edit product form. Clicking the link will open a new browser window with the section of the documentation that pertains to that topic.
2002.07.19	2002.11.15	Feature Request	Added "list" hyperlinks next to several fields on the add/edit product form. Clicking the link will open a new browser window with a list of items already defined for the fields. For example, clicking the "list" hyperlink next to the category field will open a new browser window with all product categories listed.
2002.07.19	2002.11.15	Feature Request	All database related errors in store administration are now reported with the error message returned by Lasso to aid in troubleshooting connectivity and permissions problems.
2002.07.19	2002.11.15	Feature Request	Date dependent reports can now be generated in a printer friendly version. Check the "Display print version" checkbox on the report administration menu. This surrounds the report with a template created with the reporttemplate_top.lasso and reporttemplate_bottom format files located in the /hostedstore/adminstore/html/ directory.
2002.07.19	2002.11.15	Feature Request	Added an "Orders from New Customers" date dependent report.

2002.07.19 2002.11.15 [Feature Request](#) The store status page now displays the customer ID and hyperlink to customer orders (or note that it is a new customer) for all orders in the queue.

The edit order form displays if the customer is new or provides a hyperlink to all of the customer orders.

2002.07.17 2002.11.15 New The store administration navigation menu has been redesigned. The current core features have been moved to buttons along the bottom where the "Configure," "Docs," and "Bugs/Requests" buttons were in older versions of the menu.

This change will facilitate the adding of more text links to new features in the gray area of the menu--it was becoming a tad crowded as navigation for some of the planned new features were added in.

Version 2002.07.15

Completed	Version	Type	Description
2002.07.15	2002.07.15	Feature Request	Added an abandoned orders report that displays the invoice number (hyperlinked to the edit order form), subtotal, and date marked as abandoned.
2002.07.15	2002.07.15	Feature Request	Added a canceled orders report that displays the invoice number (hyperlinked to the edit order form), subtotal, and date canceled.
2002.07.15	2002.07.15	Feature Request	Information returned from the payment gateway is now displayed in a "gateway information" section in the order summary area of the edit order form.
2002.07.15	2002.07.15	Feature Request	Price groups now have a tax exempt option. When enabled, customers belonging to the price group will not be charged tax even when they reside in a taxable jurisdiction.
2002.07.15	2002.07.15	Bug	Orders marked as abandoned in the maintenance area of store administration will now display the date the order was abandoned on the edit order form in order administration.
2002.07.15	2002.07.15	Bug	A "1" character is no longer inserted between the date and time on various reports.
2002.07.15	2002.07.15	Bug	The order results content page (order_result.lasso in the /hostedstore/storefront/html/directory) now masks all but the last four digits of account numbers.
2002.07.14	2002.07.15	New	Added support for the E-xact and Skip Jack payment gateways for credit card processing.
2002.07.14	2002.07.15	Documentation	The customer administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The product administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The promotion administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The review administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The price group administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The report administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The store status administration sections of the documentation have been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The store administration maintenance section of the documentation has been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Documentation	The store administration login section of the documentation has been fleshed out with additional text and screenshots.
2002.07.14	2002.07.15	Comment	Modified the way communication occurs with the DirectOne, Echo, and Plug'nPay payment gateways. This will not affect anything with regards to HostedStore--it is just a change to take advantage of a bug in the Include_Secure_URL LJAPI tag that was fixed for us.

2002.07.14	2002.07.15	Comment	<p>They key field name for the customer table has been changed from CustomerID to ID.</p> <p>This change was implemented so that all tables will have the same key field name. Identical key field names is needed to support the use of other database applications that automatically assign a key field name--a future feature of HostedStore will support such database applications (OpenBase, for example).</p>
2002.07.14	2002.07.15	Bug	<p>The file upload form in product administration has been changed to display the proper base directory for uploads, /hostedstore/images/products/, instead of /images/products/. Note that this was just an aesthetic issue, the files did and still do upload to the proper directory.</p>
2002.07.14	2002.07.15	Bug	<p>Settled Orders and Settled Orders by Batch reports no longer include purged orders.</p>
2002.07.14	2002.07.15	Bug	<p>Modified the order_detail.lasso content page (in the /hostedstore/admincustomer/html/ directory) to not display "Shipped via:" for shipped ordered items that do not have a shipping vendor specified.</p>
2002.07.13	2002.07.15	Bug	<p>Items that are a part of an order (ItemType field does not have a value of "Cart") are no longer deleted when a saved cart is deleted. The CartName and CartEmail fields are cleared instead so there is no conflict with viewing items that are shared between an order and a saved cart.</p>
2002.07.12	2002.07.15	New	<p>Added tracking of date an order was declined to order administration.</p> <p>The following field must be added to the invoice table:</p> <p>DateDeclined datetime</p>
2002.07.12	2002.07.15	Documentation	<p>The order administration sections of the documentation have been fleshed out with additional text and screenshots.</p>
2002.07.12	2002.07.15	Documentation	<p>The affiliate administration sections of the documentation have been fleshed out with additional text and screenshots.</p>
2002.07.11	2002.07.15	Bug	<p>The cart buttons no longer display for an empty cart when manually adding orders or returns in order administration.</p>
2002.07.10	2002.07.15	New	<p>A button to list affiliates awaiting payout has been added to the affiliate administration menu.</p> <p>The listing shows a summary of affiliates awaiting payout: affiliate ID, compensation plan, compensation amount, order count, aggregate subtotal, and aggregate payout. The affiliate ID is hyperling to the affiliate payout screen.</p> <p>A similar list shows a summary of affiliates with orders in the queue.</p>
2002.07.07	2002.07.15	Feature Request	<p>Added support for price groups.</p>
2002.07.07	2002.07.15	Comment	<p>Modified HS_Account cookie to support price groups.</p> <p>The cookie is a double pipe () delimited string blowfish encrypted with a key of HS_Encrypt. The contents of each part of the string are detailed below by field number:</p> <ol style="list-style-type: none"> 1) The key field value for the customer record. 2) The CustomerID field from the customer record. 3) The PriceGroupID field from the customer record. 4) The Type field from the price group table. 5) The Value field from the price group table. 6) The ApprovedForPO field from the customer record.
2002.07.06	2002.07.15	Bug	<p>The state field select list on the customer administration menu no longer has a value selected after adding or updating a customer record.</p>

- 2002.07.05 2002.07.15 New Products can be designated as member only items.
- If a product is available to members only, the following restrictions are enforced if the customer is not a current member:
- Member only products will not appear in product search results.
 - Member only products cannot be added to the cart.
 - Member only products cannot be added to a wish list.
- The restriction does not apply if:
- The member only product is purchased from a saved cart.
 - The member only product is purchased from a shopping list.
 - The member only product is purchased from a wish list.
- 2002.07.05 2002.07.15 [Feature Request](#) Savable/Recoverable carts have been added. This additions incorporated several changes. A brief overview is provided below - details can be found in the documentation.
- Added HS_UseSavedCarts and HS_SavedCartDays settings to store configuration.
 - Purge carts in the maintenance area of store administration does not delete saved carts that meet the age criteria unless the saved cart has expired. There is also separate results reporting for cart records deleted and saved cart records deleted.
 - Two content pages were added to the /hostedstore/storefront/html/ directory. cart_saveform.lasso is the form used by the customer to save their cart. cart_recoverform.lasso is used by the customer to recover their cart.
 - Teo internal format files were added to process saved carts. Hooks were added for each.
- 2002.07.05 2002.07.15 [Feature Request](#) Added forced search ranking that will sort product search results based on the search ranking prior to the true sort field.
- Note that this feature is a little strange due to the way sorting works in MySQL. Assign a two digit number as the ranking where larger numbered ranks are displayed first, i.e., a product ranked 99 appears before a product ranked 1.
- This feature involved two changes:
- SearchRank field added to product table.
 - Search Rank added to product administration in the Categorization section of the add/edit product form.
- 2002.07.05 2002.07.15 Documentation Documentation was changed to reflect that "Delete Files" permission must be granted to the user used for file uploads on the Setup>Groups>Files tab in Lasso Administration.
- 2002.07.04 2002.07.15 [Feature Request](#) Added hook variables to set email subject lines. All variables are prefixed with HSH_Email_. Details can be found in the HostedStore Hooks section of the documentation.
- The email subject hook variables are not available for any store administration emails nor for the following emails:
- Order confirmation sent to customer.
 - Order confirmation sent to HS_SalesEmail.
 - Out of stock message sent to HS_StoreManagerEmail.
 - Membership purchase/update notice sent to HS_MembershipsEmail.

2002.07.03 2002.07.15 Comment Added [\$HS_PathToStore] to all hyperlinks and form actions in the content pages. These changes were made to avoid possible future problems that could arise from non-root paths when third-party modules are used.

The following structures were changed:

```
href="/HostedStore.LassoApp  
href="HostedStore.LassoApp  
action="/HostedStore.LassoApp  
action="HostedStore.LassoApp
```

The new structures, respectively:

```
href="[$HS_PathToStore]/HostedStore.LassoApp  
href="[$HS_PathToStore]/HostedStore.LassoApp  
action="[$HS_PathToStore]/HostedStore.LassoApp  
action="[$HS_PathToStore]/HostedStore.LassoApp
```

2002.07.03 2002.07.15 Comment The EAD feature was rewritten. There are still some browser compatibility issues with direct downloading via the hyperlink. Clearly instructing customers to use the browser "Save As..." capability should eliminate most problems.

Version 2002.07.01

Completed	Version	Type	Description
2002.06.30	2002.07.01	Bug	Radio button attributes are now displayed properly on the modify item content page.
2002.06.30	2002.07.01	Bug	Code was added to the modify item hyperlink on the cart detail page and to the code to reload the cart after modifying the item to prevent caching.
2002.06.29	2002.07.01	Bug	Radio button display option for products now retains its value when loading the update product form in product administration.
2002.06.22	2002.07.01	Bug	An error is no longer generated when emailing a quote from the cart. This was due to changes earlier that changed the structure of HSP_Cart to accommodate publishing of the ProductID on the cart detail content page and linking to non-configurable product detail pages from the cart detail content page.
2002.06.22	2002.07.01	Bug	Fixed erroneous date published on order detail content page in customer administration (/hostedstore/admncustomer/html/order_detail.lasso) due to the addition of the date canceled by customer feature. The old date was accessed by \$HSP_Order->Get:3->Get:6, the correct value is now \$HSP_Order->Get:3->Get:7.
2002.06.22	2002.07.01	Comment	Modified payment_modify and shipping_info content pages: 1) Removed MICR number symbol replacement text and changed label to "Account Number." 2) Removed CCV2 entry because this value cannot be retained. This has been moved to the payment gateway forms where appropriate.
2002.06.22	2002.07.01	Comment	Removed internal code that validated CCV2 presence when the HS_UseSecurityCode is set in store configuration. CCV2 entry has been moved to the payment gateway forms so the value is no longer stored per CC company requirements.
2002.06.22	2002.07.01	New	Completed support for PayPal IPN.
2002.06.19	2002.07.01	Bug	Affiliate menu variables are now cleared before loading the add affiliate form.
2002.06.19	2002.07.01	Bug	Division by zero error has been trapped for the general statistics report.
2002.06.19	2002.07.01	Bug	Secure path to store setting in store configuration is now maintained.

Version 2002.06.17

Completed	Version	Type	Description
2002.06.06	2002.06.17	New	Credit card processing through the NETbilling payment gateway is now supported.
2002.06.04	2002.06.17	Bug	Enabled "Sales by Affiliate, Itemized" report.
2002.06.03	2002.06.17	New	<p>The HS_PrintableReceipt module has been incorporated into HostedStore.</p> <p>The content page is named order_printinvoice.lasso and is located in the /hostedstore/storefront/html/ directory.</p> <p>The page is called by passing a form parameter named ID with a value of the encrypted invoice number for the order.</p>
2002.05.31	2002.06.17	Bug	<p>The "Total with Tax" column no longer appears on the default cart display content page if the store has been configured with no taxing jurisdictions. The "Total without Tax" column displays right aligned instead and the title "Total without Tax" at the top of the column is eliminated.</p> <p>This fix required changes to two files:</p> <ul style="list-style-type: none">- cart_display.lasso (in the /hostedstore/storefront/html/ directory)- calculate_tax.lasso (in the /hostedstore/includes/calculations/ directory)
2002.05.29	2002.06.17	Bug	Fixed incorrect link in email_friend.lasso file in the /hostedstore/storefront/email/ directory to reflect LassoApp syntax.
2002.05.29	2002.06.17	Bug	<p>There was an error in the mapping between the store configuration form and the reading of the parameters. The issue has been fixed. Note that this is not just an issue for Windows 2000. Mac OS X is also affected.</p> <p>A temporary workaround for older versions of HostedStore is to use the "Affiliate" database name for the name of the database containing the product table and to use the "Product" database name for the name of the database containing the affiliate table.</p>
2002.05.29	2002.06.17	Comment	<p>Special thanks to Philip McDonnell of Pagehit Internet Solutions <http://www.pagehit.com> and Keith Schuster of Schuster & Company <http://www.schusterandcompany.com> for their time and valuable assistance in working out the discrepancies between Lasso on Windows and OS X that caused HostedStore to fail on Windows.</p> <p>Thanks guys! It works now.</p>
2002.05.27	2002.06.17	Bug	Fixed a bug in the export code for Stone Edge Order Manager that inserted the shipping email address after the shipping company. The shipping email address has been removed from the export.

Version 2002.05.27

Completed	Version	Type	Description
2002.05.27	2002.05.27	Bug	All internal table references have been switched to lowercase. This change addresses a table referencing issue on Windows where mysql automatically converts table names to lowercase, but Lasso sends the request in uppercase even if a table alias in uppercase has been set in Lasso Administration.
2002.05.27	2002.05.27	Feature Request	<p>Added support for shared SLL certificates.</p> <p>To configure the store for shared SSL certificates, the following must be done in store configuration:</p> <ol style="list-style-type: none">1) Select "Use Shared Certificate Path" from the Secure Path to Store select list in the URLs section.2) Type the full URL path to the /hostedstore directory (do not include /hostedstore in the path) for the store in the Shared Certificate Path text box. Note that it will be necessary to include the https protocol in the URL. <p>For example, if the shared hosting directory is /mystore and the URL for the shared certificate is www.hostingprovider.com, then the following would be entered into the Shared Certificate Path text box: https://www.hostingprovider.com/mystore</p>
2002.05.26	2002.05.27	Bug	<p>Fixed a bug that was a side effect of making the modification that allowed non-configurable items to be linked from the cart display content page to the product detail content page.</p> <p>The following items are now working properly:</p> <ul style="list-style-type: none">- order review content page- manual order add item content page- manual order checkout content page- manual return add item content page- manual return checkout content page
2002.05.26	2002.05.27	Comment	<p>The "View Image Library" report has been renamed "Image Library."</p> <p>The report has been rewritten to increase performance. On our test database of 2347 products, the report processing time was decreased to 53 seconds from 171 seconds. Of course, rendering time can take much longer especially for a large product catalog due to the creation of a large table with multiple images.</p>
2002.05.26	2002.05.27	Feature Request	Added "View Payout History" and "Post Payout(s)" buttons to the edit affiliate form.
2002.05.26	2002.05.27	Feature Request	<p>A "Payout History by Affiliate, Itemized" date dependent report has been added.</p> <p>The report groups payouts by affiliate and reports the same information as the affiliate payout history generated from the edit affiliate form.</p>
2002.05.26	2002.05.27	Feature Request	<p>The customer order cancelation process has been changed. When a customer cancels an order, the order status is changed to "Canceled (Pending)." The sales email address is still sent a notice.</p> <p>The store manager then needs to change the status to "Canceled" if the order is indeed to be canceled, otherwise the status should be set to the appropriate value.</p> <p>When the order status is set to "Canceled" then inventory is updated for all ordered items and inventory adjusting attributes for the ordered items.</p>

2002.05.26 2002.05.27	New	<p>A new report, Missing Images, has been added to the date independent reports. The purpose of this report is to decrease the load time of the Image Library report when all that is wanted is a list of missing images.</p> <p>If zoom images are used in the store (HS_UseZoomImages is set to Y in store configuration), then a product will be reported as missing an image if the thumbnail, detail, or zoom image is missing. If zoom images are not used, then a product will be reported as missing an image if the thumbnail or detail image is missing. Even in the latter case, the presence of absence of a zoom image will be reported--it just is not a criteria for reporting.</p>
2002.05.25 2002.05.27	Feature Request	<p>The age of the product record (the current date less the date in the DateCreated field) has been added to the HSP_ProductList and HSP_Product variables.</p> <p>\$HSP_ProductList->Get:Loop_Count->Get:7->Get:9 is used to access the value on the search_list.lasso content page.</p> <p>\$HSP_Product->Get:13->Get:13 is used to access the value on the product_detail.lasso content page.</p>
2002.05.25 2002.05.27	Feature Request	<p>Non-configurable items in the cart are now hyperlinked to the product detail page.</p> <p>(The request for hyperlinked wish list items was already implemented. Unless we misunderstood the intent of the request.)</p>
2002.05.25 2002.05.27	Feature Request	<p>Promotions can be duplicated from the edit promotion form.</p>
2002.05.23 2002.05.27	Feature Request	<p>The AffiliatePayout field in the Invoice table now records the following double pipe () delimited information:</p> <ol style="list-style-type: none"> 1) Date of payout in YYYY-MM-YY format 2) Check number 3) Check amount (as a decimal without currency formatting). 4) Affiliate plan at the time 5) Amount/percentage of the plan at the time 6) The theoretical payout with currency formatting as specified by currency formatting preferences set in store configuration.
2002.05.23 2002.05.27	New	<p>The "Check for Bug Reports for Your Version" button on the maintenance menu in store administration is now enabled.</p> <p>Clicking the button will query the HostedStore server for bug reports for the version of HostedStore as reported by HS_Version. If bugs have been reported, then a list of bugs including date reported, status, version released (if the bug has a closed status), title, and brief description will be displayed. The title will be displayed as a hyperlink that when clicked will open a new browser window with the full bug report and any comments.</p>
2002.05.22 2002.05.27	Bug	<p>Defined global variable for HSI_Review so that an error would not occur if reviews were not enabled for the store.</p>
2002.05.22 2002.05.27	Feature Request	<p>HS_BuildMySQLTables.LassoApp utility provides an interface that will create all of the tables necessary (and optional tables) for HostedStore and all HostedStore modules released by HostedStore Corporation.</p>

Version 2002.05.20

Completed	Version	Type	Description
2002.05.20	2002.05.20	Bug	Added additional attribute validation to prevent a URL hack that could be used to alter cart attributes.
2002.05.20	2002.05.20	Feature Request	<p>Added product sales ranking to array returned to the default product detail content page (product_detail.lasso in the /hostedstore/storefront/html/ directory).</p> <p>The ranking is obtained with \$HSP_Product->Get:13->Get:12.</p>
2002.05.20	2002.05.20	Workaround	<p>Rewrote internal files and content pages to workaround an enormous bug in Lasso Professional 5 that prevented HostedStore from functioning properly on Windows OS.</p> <p>Note that this change was comprehensive and incorporated literally thousands of code modifications. Unfortunately so many changes has likely caused bugs to creep into HostedStore. What was previously functioning may no longer function properly.</p> <p>We have done our best to debug before release, but please be patient if anything has broken. Please report any bugs and we will fix it ASAP.</p>
2002.05.19	2002.05.20	Feature Request	<p>Product attributes can be displayed as radio buttons by checking the "Display as radio button" checkbox for the attribute in product administration. If not checked, then the attribute selection display defaults to a select list.</p> <p>Note that the radio button option and the default select list are only relevant for stores that use the default product detail content page (product_display.lasso) and the default product modification content page for configurable products (cart_modifyitem.lasso) in the /hostedstore/storefront/html/ directory.</p>
2002.05.18	2002.05.20	Feature Request	<p>The default product detail content page can be changed from product_detail.lasso to another format file by specifying the file name in the add/edit product form.</p> <p>Two important notes:</p> <ol style="list-style-type: none">1) Paths to the format files should be based on a root directory of /hostedstore/storefront/html/.2) The existence of the format file is not checked when adding/updating the product to facilitate populating the product catalog before code is written. However, if the format file does not exist during runtime, then the default content page will be used instead.
2002.05.18	2002.05.20	Feature Request	Added affiliate payout tracking.
2002.05.17	2002.05.20	Bug	The product detail page no longer displays product complements that are out of stock and not flagged to display but not sell.
2002.05.17	2002.05.20	Feature Request	<p>Added product return capability into order administration. Returns are created just as if they were orders, i.e., the return is assigned to a customer, products are shopped for using a shopping cart, and a checkout process is used. All quantities for the returned items is added back to the item's inventory. Returns must be processed just as if they were orders so the return must be shipped to close it out.</p> <p>Any comments to be associated with the order can be placed in the comments area during checkout. Note that the comments will go into the same field that is used when customers add comments to orders so any sensitive info should be reserved for the administrator comments field. The comments field is meant for notes like "Returned productd from Invoice#XXXXX."</p>

2002.05.17 2002.05.20 New Order export for Stone Edge Order Manager has been moved from the Maintenance menu to the Order Administration menu.

2002.05.16 2002.05.20 [Feature Request](#) Customers can request to be notified when an out of stock product is in stock by submitting their email address to a "wait list."

This feature required several changes:

- 1) WaitList field added to the Product table.
- 2) Modification of the product_detail.lasso default content page in the /hostedstore/storefront/html/ directory.
- 3) The wait list script is accessed by sending the following values to HostedStore.LassoApp?-ResponseLassoApp=detail.lasso:
"1" - The email address to add to the wait list.
"Submitted_EmailNotify" - The name of the button.

If the email address is successfully added to the wait list, then variable "1" will have no value and variable "Submitted_EmailNotify" will have a value. If an error occurred, then both variable "1" and variable "Submitted_EmailNotify" will have a value.

Email addresses are stored in the WaitList field of the Products table as double pipe (||) delimited entries.

2002.05.15 2002.05.20 Bug Calculation used to determine out of stock items was changed in the product search and product detail scripts. This affects the way products are returned to the search_list.lasso and product_detail content pages in the /hostedstore/storefront/html/ directories.

Version 2002.05.15

Completed	Version	Type	Description
2002.05.14	2002.05.15	Bug	<p>Table export code in maintenance area of store administration has been changed to write to file for each record.</p> <p>Note that while this change will decrease on memory usage, it could lead to incomplete exports if somehow the export process is interrupted (like a restart of Lasso Service, for example).</p>
2002.05.13	2002.05.15	Bug	<p>Modified the affiliate payout reports to be calculated based on net subtotal (subtotal plus subtotal adjustment).</p> <p>This change affects the report that the affiliate views and the date dependent "Sales by Affiliate" report generated in store administration.</p>
2002.05.13	2002.05.15	Feature Request	<p>Batch settlement reports now sort by invoice number. The "Settled Orders by Batch" report sorts on batch number and then invoice number.</p>
2002.05.13	2002.05.15	New	<p>Added support for Stone Edge Order Manager. Exports can be generated in the maintenance area of store administration. Order Manager can also connect directly to HostedStore to download data.</p> <p>More information on Store Manager can be found at <http://www.stoneedge.com>.</p>
2002.05.12	2002.05.15	Bug	<p>Restricted products are now enforced even if inventory enforcement is not used.</p>
2002.05.12	2002.05.15	Feature Request	<p>Static product detail pages can now be generated for the found set on the product search results list. A single static product detail page can also be generated on the product add/edit form at the same time as when the product is added, updated, or saved as a new product.</p>
2002.05.12	2002.05.15	New	<p>Optimized some database queries to allow for better indexing.</p> <p>Created database index suggestions based on queries currently executed by HostedStore. Note that indexes take extra hard drive space so the store administrator may have to make choices as to whether or not to use all, some, or none of the index suggestions.</p> <p>All possible indexes were not created. We looked for the indexes that would increase performance for a majority of queries.</p>
2002.05.11	2002.05.15	Feature Request	<p>Inventory can now be optionally enforced during checkout.</p> <p>Enforcement was made optional because use of this feature could lead to confused and frustrated customers. This feature, however, does have its place with regards to one-of-a-kind items, limited quantities, or clearance items.</p> <p>Changes made:</p> <ol style="list-style-type: none">1) The edit product page in store administration has a checkbox to flag the product for inventory enforcement.2) An array named HSP_EnforceInventory will be returned to the cart detail page if items have been ordered with a quantity larger than the inventory on hand. Details of the array elements (the number corresponds to the position in the array):<ol style="list-style-type: none">1. ProductID field.2. ProductName field.3. Quantity ordered.4. Quantity available (OnHand field less the WebSellQty field).

2002.05.11 2002.05.15 [Feature Request](#) Added support for restricted products.

In product administration, products can be assigned a class. The class groups products into a restriction category. Products can also be restricted to a specific class. If a product is restricted, then the customer will not be able to checkout if a product from the restricted class is not in the cart.

Adding this feature involved several changes:

- 1) Added "Class" and "RestrictTo" fields to the add/edit product form under a new heading named "Dependencies/Restrictions."
- 2) Added a new array, HSP_RestrictedProduct, to the default cart display content page. HSP_RestrictedProduct identifies the restricted product that have not met the restriction. Details of the array elements (the number corresponds to the position in the array):
 1. ProductID field for the restricted product.
 2. ProductName field for the restricted product.
 3. RestrictTo field for the restricted product.
- 3) Added a new keyword (class) to the search script. Passing the restriction class as a form parameter to the search script will display a list of products that can be purchased to satisfy the restriction.

2002.05.10 2002.05.15 [Feature Request](#) Made changes to how product reviews function:

- 1) In store configuration, the default auto-publish status for reviews can be set.
- 2) In store configuration, an email notification can be requested every time a review is added by a customer.
- 3) The publish flag and text of reviews can be modified in store administration. A reviews tab was added to the navigation.

2002.05.10 2002.05.15 [Feature Request](#) Added a "View Orders" button to the edit form in customer administration. Clicking the button will pass the customer's ID to the order search script.

2002.05.10 2002.05.15 [Feature Request](#) Added a "View Orders" button to the edit form in affiliate administration. Clicking the button will pass the affiliate's ID to the order search script.

2002.05.10 2002.05.15 [Feature Request](#) Added an "Adjust Inventory" and an "Adjust Pricing" button to the product search results list in store administration. Clicking the appropriate button will load the found set into a form for inventory or pricing adjustment.

2002.05.10 2002.05.15 [Feature Request](#) Added support for volume pricing.

Read the product administration section of the documentation for details.

2002.05.06 2002.05.15 [Documentation](#) The online documentation has now been split into smaller pages. This required the creation of a separate LassoApp named HostedStore_Documentation.LassoApp. To view the documentation, simply install the file in a Web serving directory and then point to it with a browser. The "Documentation" button in store administration now opens a new browser window that points to the file. It is assumed that the file is installed at the root of the Web serving domain to which HostedStore has been licensed.

2002.05.04 2002.05.15 [Feature Request](#) Added creation and modification search to product administration. Search operator defaults to equals.

2002.05.04	2002.05.15	Feature Request	<p>Added optional email notification when the product minimum on hand to sell (WebSellQty field) threshold is reached.</p> <p>This feature required several modifications:</p> <ol style="list-style-type: none"> 1) The out of stock email address to receive notifications is set in the store configuration file. This field is required even if the notification is never used. 2) Notification is set at the product level by checking a checkbox on the add/edit product screen in product administration. 3) The email body for the notification is set using a new file, email_outofstock.lasso, located in the /hostedstore/adminstore/email/ directory. An array variable, HSP_OutOfStock, contains some product information that may be used in the email. Details of the array elements (the number corresponds to the position in the array): <ol style="list-style-type: none"> 1. ProductID field. 2. ProductName field. 3. WebSellQty field. 4. OnHand field (updated with new value). 5. OutOfStockDisplay field.
2002.05.04	2002.05.15	New	<p>Product detail and thumbnail images that are missing will be replaced by an image named noimage.gif from the /hostedstore/images/products/detail/ and /hostedstore/images/products/thumbnail/ directories, respectively.</p> <p>Code for missing product detail images was added to the following pages in the /hostedstore/storefront/html/ directory:</p> <ul style="list-style-type: none"> - cart_modifyitem.lasso - product_detail.lasso <p>Code for missing product thumbnail images was added to the following pages in the /hostedstore/storefront/html/ directory:</p> <ul style="list-style-type: none"> - email_friend.lasso - review_detail.lasso - review_form.lasso - review_list.lasso - search_list.lasso
2002.05.01	2002.05.15	New	Added support for check processing through the PayByCheck.com payment gateway.
2002.05.01	2002.05.15	New	Added support for credit card processing through the Echo payment gateway.
2002.05.01	2002.05.15	New	<p>Added 10 check processing parameters and 10 credit card processing parameters to the store configuration form. These parameters will be used in support of various payment gateway settings like merchant ID, transaction type, etc.</p> <p>Details about each parameter are provided in the documentation where the payment gateway settings and processing forms are described.</p>
2002.04.30	2002.05.15	Bug	Wishlist items can now be added to the shopping cart from the customer administration area when a cart has not already been created.
2002.04.25	2002.05.15	Feature Request	<p>Added support for intermediate search forms.</p> <p>Setting the -ResponseLassoApp parameter to searchpage.lasso (instead of search.lasso) will pass through all parameters to the content page located at /hostedstore/storefront/html/search_searchpage.lasso. No search will be conducted. The parameters are simply passed through.</p> <p>To specify a different intermediate page to return, pass a parameter named page with a value of the path and file name to return. Paths should be set from the /hostedstore/storefront/html/ directory.</p>

2002.04.25 2002.05.15 [Feature Request](#) Added support for checkbox generated search criteria.

Parameters were added that are consistent with the numbering scheme currently used for search form parameters.

s1cb through s8cb are used to notify HostedStore that the search criteria for the field was generated from checkboxes. A contains search will be performed for all of the values.

s1lop through s8lop are used to specify the logical operator to use for all of the criteria. Use AND if all criteria must be met. Use OR if any criteria can be met. Defaults to OR.

2002.04.25 2002.05.15 [Feature Request](#) Added support for multiple search criteria.

Parameters were added that are consistent with the numbering scheme currently used for search form parameters.

s1d through s8d are used to notify HostedStore that the search criteria for the field may contain multiple search strings. The parameter value should be the delimiter used to separate search strings. A contains search will be performed for all of the values.

s1lop through s8lop are used to specify the logical operator to use for all of the criteria. Use AND if all criteria must be met. Use OR if any criteria can be met. Defaults to OR.

Version 2002.04.15

Completed	Version	Type	Description
2002.04.15	2002.04.15	New	<ul style="list-style-type: none">• Added intermediate search pages.• Added support for checkbox based search criteria where all or some of the selected criteria can apply.• Added support for multiple criteria in textbox and textarea form elements where all or some of the criteria can apply and the delimiter can be specified.

Manifest

A list of all files distributed with HostedStore is provided below.

```
/hostedstore
  /admincustomer
    /email
      email_cancelorder.lasso
      email_password.lasso
      email_subscribe.lasso
      email_unsubscribe.lasso
      email_wishlist.lasso
    /html
      address.lasso
      addressbook.lasso
      billingaddress.lasso
      files_list.lasso
      loginform.lasso
      menu.lasso
      newsletter.lasso
      order_detail.lasso
      order_list.lasso
      password.lasso
      shoppinglist_detail.lasso
      shoppinglist_list.lasso
      wishlist_email.lasso
      wishlist_menu.lasso
      wishlist_preferences.lasso
      wishlist_purchases.lasso
      wishlist_view.lasso
  /adminstore
    /email
      email_customerstatus.lasso
      email_outofstock.lasso
      email_subscribe.lasso
      email_supplier.lasso
      email_unsubscribe.lasso
      email_waitlist.lasso
    /suppliers
  /html
    module_menu.lasso
    order_mailinginvoice.lasso
    order_packinginvoice.lasso
    order_picklist.lasso
  /report
  /selectlists
    customer_emailgroups.lasso
  /templates
    /report
      bottom.lasso
      top.lasso
  /adminsupplier
    /email
      email_customerstatus.lasso
```

```
/templates
  /default
    bottom.lasso
    ss.css
    top.lasso
/affiliate
  /email
    email_password.lasso
  /html
    loginform.lasso
    orders_list.lasso
/backups
/EADFiles
/helpcenter
  /email
    email_customerservice.lasso
  /html
    about.lasso
    browsersupport.lasso
    cancelorder.lasso
    contactus.lasso
    customerserviceform.lasso
    errors.lasso
    faq.lasso
    firsttime.lasso
    manageaccount.lasso
    menu.lasso
    orderproblems.lasso
    orderstatus.lasso
    ordertracking.lasso
    paymentoptions.lasso
    placingorders.lasso
    preorderpolicy.lasso
    pricing.lasso
    privacypolicy.lasso
    promotions.lasso
    returnpolicy.lasso
    review.lasso
    salestax.lasso
    searchtips.lasso
    securitypolicy.lasso
    shipping.lasso
/images
  /products
    /descriptions
    /detail
    /thumbnail
    /zoom
/includes
  /calculations
    calculate_shipping.lasso
    calculate_tax.lasso
  /javascript
    popup_options.lasso
    popup_products.lasso
/navigation
  browseproducts.lasso
```

```
    navfooter.lasso
/selectlists
    countries_shipping.lasso
    countries.lasso
    creditcards.lasso
    months.lasso
    states.lasso
    years.lasso
/ordermanager
/products
/storefront
    /email
        email_customerapproval.lasso
        email_friend.lasso
        email_membershipsales.lasso
        email_newaccount.lasso
        email_password.lasso
        email_quote.lasso
        email_salesapproval.lasso
        email_subscribe.lasso
    /suppliers
/gateways
    cc_achdirect.lasso
    cc_authorizenet_adc.lasso
    cc_authorizenet_aim.lasso
    cc_bluepay.lasso
    cc_directone.lasso
    cc_echo.lasso
    cc_eprocessingnetwork.lasso
    cc_exact.lasso
    cc_iongate.lasso
    cc_linkpoint_basic.lasso
    cc_netbilling.lasso
    cc_offline.lasso
    cc_other_form.lasso
    cc_other_script.lasso
    cc_payoffshore.lasso
    cc_planetpayment_adc.lasso
    cc_plugnpay.lasso
    cc_psigate.lasso
    cc_securepay.lasso
    cc_skipjack.lasso
    cc_verisign_payflowlink.lasso
    cc_verisign_payflowpro.lasso
    chk_achdirect.lasso
    chk_authorizenet_adc.lasso
    chk_authorizenet_aim.lasso
    chk_linkpoint_basic.lasso
    chk_netbilling.lasso
    chk_other_form.lasso
    chk_other_script.lasso
    chk_paybycheck.lasso
    chk_planetpayment_adc.lasso
    chk_plugnpay.lasso
    chk_verisign_payflowlink.lasso
    chk_verisign_payflowpro.lasso
    paypal.lasso
```

```
/html
  cart_display.lasso
  cart_modifyitem.lasso
  cart_recoverform.lasso
  cart_saveform.lasso
  customer_info.lasso
  customer_login.lasso
  email_friend.lasso
  index.lasso
  order_display.lasso
  order_printinvoice.lasso
  order_result.lasso
  payment_modify.lasso
  product_detail.lasso
  review_detail.lasso
  review_form.lasso
  review_list.lasso
  search_form.lasso
  search_list.lasso
  search_searchpage.lasso
  search_subcategories.lasso
  shipping_info.lasso
  shipping_modify.lasso
  store_closed.lasso
  wishlist_detail.lasso
  wishlist_list.lasso
  wishlist_loginform.lasso
  wishlist_searchform.lasso
/templates
  /HS_1
    bottom.lasso
    /images
      blue_line.gif
      btn_addtocart.gif
      btn_back.gif
      btn_back_over.gif
      btn_checkout.gif
      btn_checkout_over.gif
      btn_continueshopping.gif
      btn_continueshopping_over.gif
      btn_emailfriend.gif
      btn_info.gif
      btn_recalculate.gif
      btn_recalculate_over.gif
      btn_reviewproduct.gif
      btn_viewcart.gif
      btn_wishlist.gif
      btn_zoom.gif
      bullet.gif
      logo.gif
      noinfo.gif
      search_advanced.gif
      search_advanced_over.gif
      search_bevel_left.gif
      search_bevel_right.gif
      search_div.gif
      search_endbevel.gif
```

```
search_go.gif
search_go_over.gif
search_left.gif
search_right.gif
search_tips.gif
search_tips_over.gif
search_title.gif
search_top.gif
search_topleft.gif
search_topright.gif
sidebar_bottom.gif
sidebar_bottomright.gif
sidebar_browse_title.gif
sidebar_cart_title.gif
sidebar_right.gif
sidebar_top.gif
sidebar_topright.gif
spacer.gif
topbar_about.gif
topbar_about_over.gif
topbar_account.gif
topbar_account_over.gif
topbar_bevel.gif
topbar_div.gif
topbar_help.gif
topbar_help_over.gif
topbar_home.gif
topbar_home_over.gif
topbar_right.gif
ss.css
top.lasso
/HS_2
bottom.lasso
/images
  browse_products.gif
  btn_addtocart.gif
  btn_checkout.gif
  btn_continueshopping.gif
  btn_emailfriend.gif
  btn_go.gif
  btn_info.gif
  btn_recalculate.gif
  btn_reviewproduct.gif
  btn_viewcart.gif
  btn_wishlist.gif
  btn_zoom.gif
  bullet.gif
  cart_bg.gif
  cart_bottom.gif
  logo.gif
  noinfo.gif
  products_bg.gif
  products_bottom.gif
  search_bg.gif
  search_box_cap.gif
  side_stretch.gif
  spacer.gif
```

```
    template2_13.gif
    top_bar.gif
    top_table_bg.gif
ss.css
top.lasso
/HS_3
bottom.lasso
/images
    aboutusbotton.gif
    advancedbutton.gif
    backtomainbutton.gif
    bluebar.gif
    bottombrowse.gif
    bottomcart.gif
    bottomleftsearch.gif
    bottomyellowbar.gif
    browseheader.gif
    btn_addtocart.gif
    btn_checkout.gif
    btn_continueshopping.gif
    btn_emailfriend.gif
    btn_emailfriend_withlogo.gif
    btn_info.gif
    btn_recalculate.gif
    btn_reviewproduct.gif
    btn_wishlist.gif
    btn_zoom.gif
    bullet.gif
    cartheadheader.gif
    checkoutbutton.gif
    gobutton.gif
    helpbutton.gif
    homebutton.gif
    logo.gif
    noinfo.gif
    searchtipsbutton.gif
    spacer.gif
    topleftsearch.gif
    topmiddlesearch.gif
    toprightsearch.gif
    vieworderbutton.gif
    youraccountbutton.gif
ss.css
top.lasso
/HS_4
bottom.lasso
/images
    aboutusbotton.gif
    advancedbutton.gif
    backtomainbutton.gif
    bluebar.gif
    bottombrowse.gif
    bottomcart.gif
    bottomleftsearch.gif
    bottomrightsearch.gif
    bottomyellowbar.gif
```

```
    browseheader.gif
    btn_addtocart.gif
    btn_checkout.gif
    btn_continueshopping.gif
    btn_emailfriend.gif
    btn_emailfriend2.gif
    btn_info.gif
    btn_recalculate.gif
    btn_reviewproduct.gif
    btn_wishlist.gif
    btn_zoom.gif
    bullet.gif
    carthead.gif
    checkoutbutton.gif
    gobutton.gif
    graymiddlesection.gif
    helpbutton.gif
    homebutton.gif
    leftblackbar.gif
    logo.gif
    midleftsearch.gif
    noinfo.gif
    rightblackbar.gif
    searchtipsbutton.gif
    spacer.gif
    topleftsearch.gif
    topmiddlesearch.gif
    toprightsearch.gif
    vieworderbutton.gif
    youraccountbutton.gif
ss.css
top.lasso
/HS_5
    bottom.lasso
    /images
        aboutusbotton.gif
        advancedbutton.gif
        backtomainbutton.gif
        bluebar.gif
        bottmleftsearch.gif
        bottombrowse.gif
        bottomcart.gif
        bottomleftsearch.gif
        bottomrightsearch.gif
        bottomyellowbar.gif
        browseheader.gif
        btn_addtocart.gif
        btn_checkout.gif
        btn_continueshopping.gif
        btn_emailfriend.gif
        btn_emailfriend2.gif
        btn_info.gif
        btn_recalculate.gif
        btn_reviewproduct.gif
        btn_wishlist.gif
        btn_zoom.gif
        bullet.gif
```

carthead.gif
checkoutbutton.gif
gobutton.gif
graymiddlesection gif
greybar.gif
helpbutton.gif
homebutton.gif
leftblackbar.gif
leftgreybar.gif
logo.gif
middlesearch.gif
midleftsearch.gif
noinfo.gif
quicksearch.gif
rightblackbar.gif
searchblueend.gif
searchfieldline.gif
searchtabtop.gif
searchtipsbutton.gif
spacer.gif
topleftsearch.gif
topmiddlesearch.gif
toprightsearch.gif
vieworderbutton.gif
youraccountbutton.gif
ss.css
top.lasso
/HS_6
bottom.lasso
/images
aboutusbotton.gif
advancedbutton.gif
backtomainbutton.gif
bluebar.gif
bottmleftsearch.gif
bottombrowse.gif
bottomcart.gif
bottomleftsearch.gif
bottomleftside.gif
bottomrightsearch.gif
bottomyellowbar.gif
browseheader.gif
btn_addtocart.gif
btn_checkout.gif
btn_continueshopping.gif
btn_emailfriend.gif
btn_emailfriend2.gif
btn_info.gif
btn_recalculate.gif
btn_reviewproduct.gif
btn_wishlist.gif
btn_zoom.gif
bullet.gif
carthead.gif
checkoutbutton.gif
gobutton.gif
graymiddlesection gif

```
greybar.gif
helpbutton.gif
homebutton.gif
leftblackbar.gif
leftgreybar.gif
logo.gif
middlesearch.gif
midleftsearch.gif
noinfo.gif
quicksearch.gif
rightblackbar.gif
searchblueend.gif
searchfieldline.gif
searchtabtop.gif
searchtipsbutton.gif
spacer.gif
topgreenbar.gif
topleftsearch.gif
topmiddlesearch.gif
toprightsearch.gif
vieworderbutton.gif
whitelines.gif
youraccountbutton.gif
ss.css
top.lasso
/HS_7
bottom.lasso
/images
  aboutusbotton.gif
  advancedbutton.gif
  backtomainbutton.gif
  bluebar.gif
  bottmleftsearch.gif
  bottombrowse.gif
  bottomcart.gif
  bottomleftsearch.gif
  bottomleftside.gif
  bottomrightsearch.gif
  bottomyellowbar.gif
  browseheader.gif
  btn_addtocart.gif
  btn_checkout.gif
  btn_continueshopping.gif
  btn_emailfriend.gif
  btn_emailfriend2.gif
  btn_info.gif
  btn_recalculate.gif
  btn_reviewproduct.gif
  btn_wishlist.gif
  btn_zoom.gif
  bullet.gif
  cartheadheader.gif
  checkoutbutton.gif
  gobutton.gif
  graymiddlesection gif
  greybar.gif
  helpbutton.gif
```

```
homebutton.gif
leftblackbar.gif
leftgreybar.gif
logo.gif
middlesearch.gif
midleftsearch.gif
noinfo.gif
quicksearch.gif
rightblackbar.gif
searchblueend.gif
searchfieldline.gif
searchtabtop.gif
searchtipsbutton.gif
spacer.gif
topbar.gif
topleftsearch.gif
topmiddlesearch.gif
toprightsearch.gif
vieworderbutton.gif
whitelines.gif
whitelines2.gif
whitelines3.gif
youraccountbutton.gif
ss.css
top.lasso
/thirdparty
  /hooks
    /admincustomer
      address_entry.lasso
      address_exit.lasso
      admin_entry.lasso
      admin_exit.lasso
      admincustomer_entry.lasso
      admincustomer_exit.lasso
      cancelorder_entry.lasso
      cancelorder_exit.lasso
      files_entry.lasso
      files_exit.lasso
      index_entry.lasso
      index_exit.lasso
      shoppinglist_entry.lasso
      shoppinglist_exit.lasso
      wishlist_entry.lasso
      wishlist_exit.lasso
    /adminstore
      adminstore_entry.lasso
      adminstore_exit.lasso
      adminuser_entry.lasso
      adminuser_exit.lasso
      affiliate_entry.lasso
      affiliate_exit.lasso
      affiliatepayout_entry.lasso
      affiliatepayout_exit.lasso
      affiliatepayouthistory_entry.lasso
      affiliatepayouthistory_exit.lasso
      application_entry.lasso
      application_exit.lasso
```

autoship_entry.lasso
autoship_exit.lasso
autoshipsweep_entry.lasso
autoshipsweep_exit.lasso
batch_entry.lasso
batch_exit.lasso
customer_entry.lasso
customer_exit.lasso
image_entry.lasso
image_exit.lasso
index_entry.lasso
index_exit.lasso
inventory_entry.lasso
inventory_exit.lasso
maintenance_entry.lasso
maintenance_exit.lasso
manualcheckout_entry.lasso
manualcheckout_exit.lasso
manualorder_entry.lasso
manualorder_exit.lasso
manualreturn_checkout_entry.lasso
manualreturn_checkout_exit.lasso
manualreturn_entry.lasso
manualreturn_exit.lasso
menu_entry.lasso
menu_exit.lasso
order_entry.lasso
order_exit.lasso
ordermanager_entry.lasso
ordermanager_exit.lasso
price_entry.lasso
price_exit.lasso
priceglobaladjust_entry.lasso
priceglobaladjust_exit.lasso
pricegroup_entry.lasso
pricegroup_exit.lasso
product_entry.lasso
product_exit.lasso
promotion_entry.lasso
promotion_exit.lasso
review_entry.lasso
review_exit.lasso
supplier_entry.lasso
supplier_exit.lasso
update_entry.lasso
update_exit.lasso
/affiliate
 affiliate_entry.lasso
 affiliate_exit.lasso
 index_entry.lasso
 index_exit.lasso
/helpcenter
/storefront
 cart_entry.lasso
 cart_exit.lasso
 cartrecover_entry.lasso

cartrecover_exit.lasso
cartsave_entry.lasso
cartsave_exit.lasso
checkout_entry.lasso
checkout_exit.lasso
closed_entry.lasso
closed_exit.lasso
detail_entry.lasso
detail_exit.lasso
emailfriend_entry.lasso
emailfriend_exit.lasso
emailfriend_postdesc.lasso
emailfriend_postemail.lasso
emailfriend_predesc.lasso
emailfriend_preemail.lasso
helpcenter_entry.lasso
helpcenter_exit.lasso
index_entry.lasso
index_exit.lasso
index_postfavorites.lasso
index_postfeatured.lasso
index_prefavorites.lasso
index_prefeatured.lasso
modifyitem_entry.lasso
modifyitem_exit.lasso
modifypayment_entry.lasso
modifypayment_exit.lasso
modifypayment_postapply.lasso
modifypayment_postform.lasso
modifypayment_postvalid.lasso
modifypayment_preapply.lasso
modifypayment_preform.lasso
modifypayment_prevalid.lasso
modifyship_entry.lasso
modifyship_exit.lasso
modifyship_postapply.lasso
modifyship_postform.lasso
modifyship_postvalid.lasso
modifyship_preapply.lasso
modifyship_preform.lasso
modifyship_prevalid.lasso
review_entry.lasso
review_exit.lasso
review_postapply.lasso
review_postlist.lasso
review_postlogin.lasso
review_postrate.lasso
review_postread.lasso
review_preapply.lasso
review_prelist.lasso
review_prelogin.lasso
review_prerate.lasso
review_preread.lasso
search_entry.lasso
search_exit.lasso
search_filter.lasso
search_postsearch.lasso

search_presearch.lasso
storefront_entry.lasso
storefront_exit.lasso
wishlist_entry.lasso
wishlist_exit.lasso

/modules

/tags

HostedStore_Documentation.pdf

HostedStore_Tags.LassoApp

HostedStore.LassoApp

Just checking...

Did anybody actually read this far?