# OpenUI Look & Feel

*Last Updated: 11/24/2002, 9:00 PM*

# Table of Contents

# 1 Introduction

## OpenUI Look & Feel

*Last Updated: 11/24/2002, 9:00 PM*

## 1.1 Licensing

OpenUI License
Last updated: Last updated: Mon 02 Sep 15:05 CDT 2002
BY DOWNLOADING AND/OR IMPLEMENTING THIS SOFTWARE YOU AGREE TO THE FOLLOWING LICENSE:

Definitions:

(i) "Miva Script" refers to a proprietary server side scripting language software implemented in a variety of Miva Corporation products and developed and distributed by Miva Corporation, a California, USA Corporation.

(ii) "Miva Merchant" is the application which allows for the creation, use, maintenance and modification of electronic catalogs written for commercial and non-commercial internet use, which requires the use of Miva Script.

(iii) "OpenUI" (hereinafter "Product") is the application and source code which accompanies this License, and which allows for the creation, use, maintenance and modification of electronic catalogs written for commercial and non-commercial internet use, which requires the use of Miva Script and/or Miva Merchant.

(iv) "You" and "Developer" refers to the person, entity or organization which is using the software and source code known as "Product", and any successor or assignee of same.

(v) "Darren Ehlers, dba OpenUI Developer Consortium" (hereinafter "Owner") is the sole owner of the Product which allows for the development and use of advanced programming techniques, which requires the use of Miva Script.

(vi) "Internet Domain" is a symbolic name assigned by a valid Internet Naming Authority such as the service managed by Network Solutions.

(vii) "Storefront" is a collection of products, also referred to as a "Store" and a "Catalog" in the Miva Merchant software.

1. This license is for the downloading, acquisition and use of Product. By downloading, acquiring and using Product, YOU expressly agree to abide by this License.

2. This license is between Developer and Owner. Developer acknowledges that Owner is and has granted this license in reliance upon representations made by Developer and Developer warranties the truthfulness and accuracy of such representations. Any breach of this warranty

of truthfulness and accuracy shall constitute a breach of this License. Owner 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.

3. Developer is granted a non-exclusive license to use Product in conjunction with one Storefront on one Internet Domain (based upon License purchased). License to use Product with more than one Storefront and/or more then one Internet Domain must be separately purchased from Owner and is specifically not granted in this License.

4. Developer agrees to validate that the computer system running Product is using a licensed version of Miva Merchant.

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

6. Developer may not convert Product to a different computer language, either manually, or using an automated conversion tool, such that Product or any modification thereof will run under any language, software, or program other than or independent of, Miva Merchant and/or Miva Script.

7. Developer may not reverse-engineer, decompile, or disassemble the Product, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

8. Developer may sell, trade, give away, or otherwise transfer (hereinafter collectively "transfer") one copy of the Product, for each license purchased by Developer or by a customer of Developer, if Developer complies with the following:

   (i) Developer does not convert Product to a different computer language, either manually, or using an automated conversion tool, such that Product or any modification thereof will run under any language, software, or program other than or independent of, Miva Merchant and/or Miva Script.

   (ii) Developer has not previously transferred a copy of Product for the License purchased by Developer.

   (iii) Developer must have a written contract or agreement between Developer and the transferree that contains the following language: "The use of this software is restricted as set forth in a License and Operating Agreement between [name of Developer] and Owner, a copy of which is attached hereto as an Exhibit. [name of Transferree] has read such License, and agrees to not violate any provisions of such License. [name of Developer] and [name of Transferree] agree that Owner is a third party beneficiary to [name of Developer] to the extent necessary for Owner to obtain the benefit of its bargain as set forth in the License, and [name of Developer] herewith assigns its rights to Owner to enforce any terms and conditions set forth in the License in the event of a breach thereof by [name of Transferree]. Furthermore, [name of Transferree], may not transfer the software to another party.

9. Any license, right, or permission to use Product is automatically revoked if Developer breaches any term or condition of this License. Developer agrees that in the event of a breach of this License, and upon the demand thereof by Owner, Developer will immediately return any physical copies of Product to Owner, will cease any further use of Product, will remove the

original and any modified or altered Product from his/her/its computers and drives, will cease his/her/its use of Product, and will cease any claimed breach of this License. Developer agrees that monetary damages alone is 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 Developer will not oppose any reasonable request for a temporary restraining order, preliminary injunction, or other relief sought by Owner in the event of a breach of this License. Owner shall not be required to notify Developer of any breach, nor make any demand or claim against Developer resulting from any such breach, or for a demand to stop any use or distribution in violation of the terms of this License, and Developer agrees that any breach of this License and damages resulting therefrom shall relate back to the first and earliest breach thereof. Failure of Owner to enforce its rights pursuant to this License shall not constitute a waiver of such rights, and shall not prejudice Owner in any later enforcement of its rights or rights to seek damages therefrom.

10. Owner WILL NOT BE LIABLE FOR (A) ANY BUG, ERROR, OMMISSION, DEFECT, DEFICIENCY, OR NONCONFORMITY IN PRODUCT; (B) IMPLIED MERCHANTIBILITY 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 Owner'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, DEVELOPERS GOODWILL, OR USE. IN NO CASE SHALL Owner's LIABILITY EXCEED THE PRICE THAT DEVELOPER PAID FOR PRODUCT.

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

12. This License Agreement shall be governed by the laws of the State of Texas, 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. Developer acknowledges that Product may be subject to United States export controls, pursuant to the Export Administrations Regulations, 15 C.F.R. Parts 768-799. Developer shall comply strictly with all requirements of the Export Administration Regulations regarding use of Product.

14. This License Agreement constitutes the entire and sole agreement between Developer and Owner, and is agreed to and entered into on the date You download or implement Product, whichever date is earlier.

# 2 Installation

## 2.1 OpenUI module installation

### 2.1.1 Standard installation (COMPILED)

* There is NO alternate installation process for the compiled OpenUI, because there are NO submodules to extract.

FTPing the Required Graphics Files

1. FTP the three **.gif* files (*openui_bg.gif*, *openui_sth.gif*, *openui_th.gif*) into your *Merchant2/graphics/en-US/admin/* subdirectory
2. (Optionally, you can upload these images thru the store's **Domain Settings** in the *admin.mv* instead of FTPing the images. This can only be done AFTER step 7 below.)

Installing the OpenUI module

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on the **Add Module** link
4. Click the **Upload** graphic button
5. Enter the filename of the module (*oui.mvc*) on your local drive
6. Press the **Upload** button
7. Press the **Add** button

### 2.1.2 Standard Installation (UNCOMPILED)

* If you get a timeout error while following the Standard Installation proceedures, your server is not able to handle the installation process. You will need to follow the Alternate Installation instructions.

FTPing the Required Graphics Files

1. FTP the three **.gif* files (*openui_bg.gif*, *openui_sth.gif*, *openui_th.gif*) into your *Merchant2/graphics/en-US/admin/* subdirectory
2. (Optionally, you can upload these images thru the store's **Domain Settings** in the *admin.mv* instead of FTPing the images. This can only be done AFTER step 7 below.)

Installing the OpenUI module

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on the **Add Module** link
4. Click the **Upload** graphic button
5. Enter the filename of the module (*oui.mv*) on your local drive
6. Press the **Upload** button
7. Press the **Add** button

At this point, you will see a number of "sub-modules" extracted and written out and at the end of the list should be the standard Merchant Module dialog and the message:

Module 'OpenUI Look & Feel' installed

* If you get any errors and/or you do not see the above message, then you will need to follow the

Alternate Installation instructions.

### 2.1.3 Alternate Installation (UNCOMPILED)

**(The following assume you are using Merchant v4.13. If you are using an earlier version of Merchant, just replace the /4.13/ with your Merchant version)**

FTPing the Required Graphics Files

1. FTP the three **.gif* files (*openui_bg.gif*, *openui_sth.gif*, *openui_th.gif*) into your *Merchant2/graphics/en-US/admin/* subdirectory
2. (Optionally, you can upload these images thru the store's **Domain Settings** in the *admin.mv* instead of FTPing the images. This can only be done AFTER step 7 below.)

FTPing the Support Files

1. Create a sub-directory called *oui* in the *Merchant2/4.13/modules/ui/* directory
2. FTP all of the *oui_*.mv* files into this directory
3. Follow the steps below

Installing the OpenUI module

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on the **Add Module** link
4. Click the **Upload** graphic button
5. Enter the filename of the module (*oui.mv*) on your local drive
6. Press the **Upload** button
7. Press the **Add** button

## 2.2 OpenUI UI Settings Export module installation

Installing the OpenUI UI Settings Export module

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on the **Add Module** link
4. Click the **Upload** graphic button
5. Enter the filename (*uiexport.mv/uiexport.mvc*) of the module on your local drive
6. Press the **Upload** button
7. Press the **Add** button

## 2.3 OpenUI UI Settings Import module installation

Installing the OpenUI UI Settings Import module

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on the **Add Module** link
4. Click the **Upload** graphic button
5. Enter the filename (*uiimport.mv/uiimport.mvc*) of the module on your local drive
6. Press the **Upload** button
7. Press the **Add** button

## 2.4 Exporting UI Settings

Exporting your store's UI settings

1. Run the *admin.mv*
2. Open the **Stores** branch
3. Click on the arrow next to the store name
4. Click on the arrow next to the **Utilities** branch
5. Click on the arrow next to the **Export** branch
6. Click on the **OpenUI UI Settings Export** module
7. To have it emailed to you, click on the **Email** tab and enter your email address
8. Click the **Export** button

## 2.5 Switching to OpenUI Look & Feel module

Setting up your store to use the OpenUI Look & Feel module

1. Run the *admin.mv*
2. Open the **Stores** branch
3. Click on the store name
4. Click on the **Layout** link tab
5. Select the **OpenUI Look & Feel** radio button
6. Press the **Update** button

## 2.6 Importing UI Settings

Importing your store's UI settings

1. Run the *admin.mv*
2. Open the **Stores** branch
3. Click on the arrow next to the store name
4. Click on the arrow next to the **Utilities** branch
5. Click on the arrow next to the **Import** branch
6. Click on the **OpenUI UI Settings Import** module
7. Click on the **Upload** graphic and fill out the filename
8. Click the **Upload** button
9. Click the **Import** button

## 2.7 Adding OpenUI Admin Extensions (UNCOMPILED)

There are NO OpenUI Admin Extensions for the COMPILED version of the OpenUI.

Perform the following ONLY if you feel comfortable editing your Merchant script (*.mv) files.

**(The following assume you are using Merchant v4.13. If you are using an earlier version of Merchant, just replace the /4.13/ with your Merchant version)**

To install the OpenUI Admin Extensions, you (or someone for you) will need to edit 1 specific file in your Mechant installation.

1. *Merchant2/4.13/features/cus/cus_mv*

*Earlier versions of the OpenUI Admin Extensions required changes also to the merchant.mv and admin.mv file. These changes have been moved to inside the OpenUI module and no longer requires modification of these files.*

### 2.7.1 Adding OpenUI Admin Extensions to cus.mv

Perform the following ONLY if you feel comfortable editing your Merchant script (*.mv) files.

**(The following assume you are using Merchant v4.13. If you are using an earlier version of Merchant, just replace the /4.13/ with your Merchant version)**

Editing your *cus.mv* file

1. Download your *Merchant2/4.13/features/cus/cus.mv*
2. Save a copy of the file BEFORE you edit it (just in case you need to undo your changes)
3. Open the original file in a text editor (or *notepad.exe*)
4. Search for `<MvFUNCTION NAME = "Action_Customer_Insert">`(around line 111)
5. Search for the first `<MvELSE>` following the above line (around line 142), and replace this line with the following text:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   </MvIF>
   <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
   <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
   <MvWHILE EXPR = "{ l.found }">
      <MvASSIGN NAME = "g.XMOD_Return"  VALUE = "">
      <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
      <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.ok" VALUE = "{
Module_Customer_Update(Customers.d.id) }">
         <MvIF EXPR = "{ NOT l.ok }">
            <MvWHILESTOP>
         </MvIF>
      </MvIF>
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
   </MvWHILE>

   <MvIF EXPR = "{ l.ok }">
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****
   <MvELSE>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

6. Search for `<MvFUNCTION NAME = "Action_Customer_Update">`(around line 181)
7. Search for the second `<MvIF EXPR = "{ l.ok }">` following the above line (around line 215), and insert the following text immediately before:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   <MvIF EXPR = "{ l.ok }">
      <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
      <MvWHILE EXPR = "{ l.found }">
         <MvASSIGN NAME = "g.XMOD_Return"    VALUE = "">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
         <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
            <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.ok" VALUE = "{
Module_Customer_Update(Customers.d.id) }">
            <MvIF EXPR = "{ NOT l.ok }">
               <MvWHILESTOP>
            </MvIF>
         </MvIF>
         <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
      </MvWHILE>
   </MvIF>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

8. Search for `<MvDO FILE = "{ g.Module_Feature_CUS_DB }" NAME = "l.ok" VALUE = "{ Customer_Delete( l.id ) }">` (around line 267)
9. Immediately prior to this line, insert the following text:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   <MvIF EXPR = "{ l.ok }">
      <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
      <MvWHILE EXPR = "{ l.found }">
         <MvASSIGN NAME = "g.XMOD_Return"     VALUE = "">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
         <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
            <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.ok" VALUE = "{
Module_Customer_Delete(Customers.d.id) }">
            <MvIF EXPR = "{ NOT l.ok }">
               <MvWHILESTOP>
            </MvIF>
         </MvIF>
         <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
      </MvWHILE>
   </MvIF>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

10. Search for `<MvDO FILE = "{ g.Module_Admin }" NAME = "l.draw_tabs" VALUE = "{ DrawTabs( 'IDEN', l.tab_list ) }">` (around line 1906)
11. Immediately prior to this line, insert the following text:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
   <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
   <MvWHILE EXPR = "{ l.found }">
      <MvASSIGN NAME = "g.XMOD_Return"  VALUE = "">
      <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
      <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.module_tabs" VALUE
= "{ Module_Customer_Tabs( Customers.d.id ) }">
         <MvIF EXPR = "{ len(l.module_tabs) }">
            <MvASSIGN NAME = "l.tab_list" VALUE = "{ l.tab_list $ ',' $ l.module_tabs }">
         </MvIF>
      </MvIF>
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
   </MvWHILE>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

12. Skip two lines, then insert the following text:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
   <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
   <MvWHILE EXPR = "{ l.found }">
      <MvASSIGN NAME = "g.XMOD_Return"  VALUE = "">
      <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
      <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.ok" VALUE = "{
Module_Customer_Content( g.Tab, g.Load_Fields, Customers.d.id ) }">
         <MvIF EXPR = "{ NOT l.ok }">
            <MvWHILESTOP>
         </MvIF>
      </MvIF>
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
   </MvWHILE>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

13. Search for `<MvFUNCTION NAME = "FieldError" PARAMETERS = "tab, field, message" STANDARDOUTPUTLEVEL = "">` (around line 3436)
14. Go back 3 lines, and immediately prior to the line `<MvFUNCTIONRETURN VALUE = 1>` (about line 3433), insert the following text:

```
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
   <MvASSIGN NAME = "g.XMOD_Action"  VALUE = "{ 'HOOKFLAGS' }">
   <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindFirst() }">
   <MvWHILE EXPR = "{ l.found }">
      <MvASSIGN NAME = "g.XMOD_Return"  VALUE = "">
      <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.trash" VALUE = "{
Module_API_Version() }">
      <MvIF EXPR = "{ 'CUSTOMER' CIN g.XMOD_Return }">
         <MvDO FILE = "{ g.Module_Root $ Modules.d.module }" NAME = "l.module_tabs" VALUE
= "{ Module_Customer_Tabs( Customers.d.id ) }">
         <MvIF EXPR = "{ len(l.module_tabs) }">
            <MvASSIGN NAME = "l.tab_list" VALUE = "{ l.tab_list $ ',' $ l.module_tabs }">
         </MvIF>
      </MvIF>
      <MvDO FILE = "{ g.Module_Library_DB }" NAME = "l.found" VALUE = "{
StoreModule_FindNext() }">
   </MvWHILE>
<MvCOMMENT>**** 3rd Party CUSTOMER Modification ****</MvCOMMENT>
```

15. Save the changes
16. Upload the file back to the original subdirectory, replacing the existing file

# 3 Updating/Upgrading

## 3.1 Upgrading v3.x (or v2.x) to v4.x

Switching to the MMUI

1. Exporting UI Settings
2. Click on the store name in the **Stores** branch of the left menu
3. Click on the **Layout** link tab
4. Select the **Merchant Look & Feel** radio button
5. Press the **Update** button
6. Open the **Modules** branch in the left menu
7. Click on **OpenUI Look & Feel** module
8. Press the **delete** button
9. Press the **delete** button again to confirm
10. Click on **OpenUI UI Settings Export** module
11. Press the **delete** button
12. Press the **delete** button again to confirm
13. Click on **OpenUI UI Settings Import** module
14. Press the **delete** button
15. Press the **delete** button again to confirm

**At this point, run your Merchant v4.x *upgrade.mv* script...**

Installing the new OpenUI v4.x

1. Standard Installation or Alternate Installation
2. OpenUI UI Settings Export module installation
3. OpenUI UI Settings Import module installation

Switching back to the OpenUI

1. Switching to OpenUI Look & Feel module
2. Importing UI Settings
3. Adding OpenUI Admin Extensions

At this point you should be up and running, with Merchant v4.x. You will need to notify your 3rd party modules to "re-hook" into the OpenUI. This is normally done by going to the configuration screen for the module and clicking the **Update** button.

## 3.2 Updating v4.x to v4.x

### 3.2.1 Standard Upgrade (UNCOMPILED)

Updating from v4.x to v4.x

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on **OpenUI Look & Feel** module

4. Click on the **Files** Tab in the right-hand frame
5. Click the **Upload** graphic button
6. Enter the filename of the module (*oui.mv*) on your local drive or use **Browse** to find it
7. Check the **Overwrite** box so that the updated module will overwrite the old version
8. Press the **Upload** button
9. Press the **Update** button

### 3.2.2 Standard Upgrade (COMPILED)

Updating from v4.x to v4.x

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on **OpenUI Look & Feel** module
4. Click on the **Files** Tab in the right-hand frame
5. Click the **Upload** graphic button
6. Enter the filename of the module (*oui.mvc*) on your local drive or use **Browse** to find it
7. Check the **Overwrite** box so that the updated module will overwrite the old version
8. Press the **Upload** button
9. Press the **Update** button

### 3.2.3 Alternate Upgrade

**(The following assume you are using Merchant v4.13. If you are using an earlier version of Merchant, just replace the /4.13/ with your Merchant version)**

FTPing the Support Files

1. FTP the *oui.mv* file into the *Merchant2/4.13/modules/ui/* directory
2. FTP all of the *oui_*.mv* files into the the *Merchant2/4.13/modules/ui/oui/* directory
3. Follow the steps below

Updating from v4.x to v4.x

1. Run the *admin.mv*
2. Open the **Modules** branch
3. Click on **OpenUI Look & Feel** module
4. Click on the **Files** Tab in the right-hand frame
5. Press the **Update** button

# Index