

Product Template Manager

Product Description:

Product Template Manager allows merchants to create multiple display templates that can then be assign to products. This module gives the merchant complete control over the entire look and feel of the product page.

Key Features:

- Multiple Templates
 - Create as many product templates as you like. Each product in your store can be assigned to its own template. If not template is assigned, the default template is used.
- Breadcrumb Trail
 - The breadcrumb trail shows your customers their position in your store relative to the category and product they are viewing.
- Product Category Listing
 - This option allows you to display all the categories the product is associated with on the product display.
- Attribute Templates
 - By using the attribute template feature you can complete control over the look and feel. Examples would be displaying attributes in a multi-column format, displaying the prices with each attribute / option, or dynamic attribute pricing.
 - Related product support.
- Dynamic Attribute Pricing
 - Product price is updated as attribute selections are made.
- Flat File Import
 - Easily update your product template associations with this flat file import tool.
- Acceptable Product Quantities
 - This feature allows you to specify quantities the product is available in. This is great for products you might sell in cases or packs of 6 for example.

Support:

Support is offered on all Sebenza modules via our online helpdesk. Module support includes, but not limited to, installation issues, compatibility, and module usage.

Requests outside of the modules scope will be addressed on a case by case scenario. In most cases these requests will on a custom development basis at our current hourly rate.

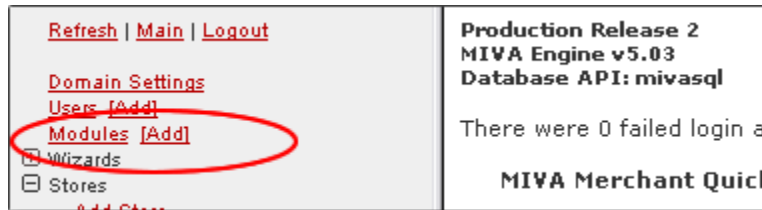
Before contacting support, please refer to not only this document... but also to our online FAQ center.

URL: <http://www.sebenza.com/>

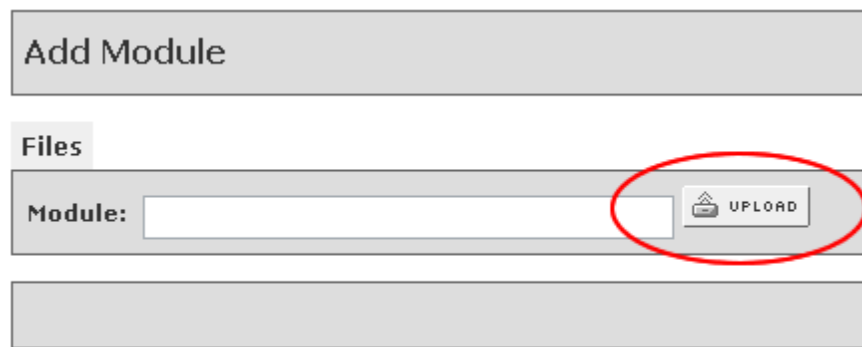
Installation Instructions

Step 1. After purchasing the Miva Merchant 5 module, download it to your computer. If it was supplied as a Zip file, unzip it to a convenient directory.

Step 2. Next, upload your new module to your Miva Merchant administration panel via the “Add” link to the right of the “Modules” link in the left navigation.



Step 3. Click the “Upload” button and select the module to be uploaded and installed. After you select the module click the “Add” button.



Now your module is installed to your Miva Merchant Domain. Next, we will need to assign the module to the store you want to use it with.

Step 4. Click “System Extension Configuration” within the left navigation under your selected store.

Step 5. Find the “Product Template Manager” module, select the checkbox, and hit “Update”.

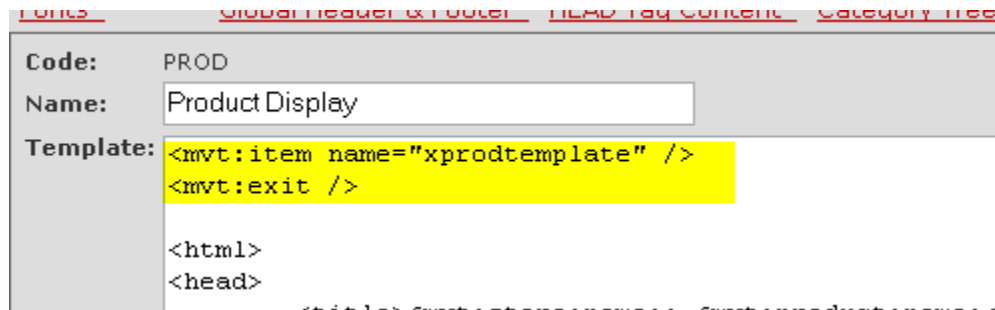
Step 6. Now that the module is assigned to the store, you will need to enter your license number. Enter your purchased license number and click “Update”.

Step 7. An “Item” named “*xprodtemplate*” was created during installation and automatically assigned to your PROD *Page Template*. In order for the Product Template Manager to control the PROD screen, we need to assign the *xprodtemplate* “Item” to the page template. Within the PROD Page template, there is a tab called Items. Click the Items tab to assign items or verify that the correct items are assigned.

Step 8. Add the following 2 lines to the beginning of the default PROD template under “Pages”:

```
<mvt:item name="xprodtemplate" />
```

```
<mvt:exit />
```



Note: By Adding the `<mvt:exit />`, you can leave the default template below it as a backup.

Creating Templates

Templates are created within the modules admin panel. The modules admin panel is located in the same spot you activated the license key... System Extension Configuration.

To add a new template, click the “Add Template” button. The module will pre-populate the template with some default template code and html. You may change this to your desire. You will also need to name your template. Once you have selected your settings, click the “Update” button to save the template.

Assigning Templates

Each product within your store now has a new tab within its settings. This new tab is named “Product Template Manager”. Within this tab, you will be able to assign the current product to a template. Hypothetically speaking, each product in your store could have its own template.

Module Usage:

The following are available arrays within each product template. Arrays can be parsed using the templates 'foreach' function.

Array: breadcrumb (Breadcrumb Trail)

Example:

```
<a href="&mvt:global:sessionurl;">&#72;&#111;&#109;&#101;</a>
<mvt:foreach iterator="crumb" array="breadcrumb">
  &#062; <a href="&mvt:global:sessionurl;Screen=CTGY&Category_Code=&mvt:crumb:code;">&mvt:crumb:name;</a>
</mvt:foreach>
```

```
&#062; &mvt:product:name;
```

Array: categories (Product Category Listing – Categories the product is assigned to)

Example:

```
<br>Product Categories:
<mvt:foreach iterator="cat" array="productcategories">
  <a href="&mvt:global:sessionurl;Screen=CTGY&Category_Code=&mvt:cat:code;">&mvt:cat:name;</a>&nbsp;
</mvt:foreach>
```

Available product fields: (Same as stock template)

product:id (example usage: &mvt:product.id;)
product:name
product:code
product:image
product:thumbnail
product:price
product:regprice (Non-Price Group Price)
product:cost
product:formatted_price (Price with currency support)
product:formatted_regprice (Non-Price Group Price with currency support)
product:descrip
product:weight
product:taxable
product:active
product:inv_active (Not null if inventory is active for product)
product:stock (Quantity in stock)
product:inv_level (Inventory level identifier : in, low, out)
product:inv_long (Inventory Long Message)
product:inv_short (Inventory Short Message)
product:quantity (Quantity of item currently in basket)
product:customfield_names:customfields:widget (Field name for field code "widget")
product:customfield_values:customfields:widget (Field value for field code "widget")

Sample Template:

```
<html>
<head>
  <title>&mvt:store.name;: &mvt:product.name;</title>
  <base href="&mvt:global.basehref;">
  <mvt:item name="head" />
</head>

<mvt:item name="body">
  <mvt:item name="hdf" param="global_header" />

  <table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr><td align="left" valign="bottom">
    <mvt:item name="hdf" param="header" />
  </td></tr>

  <tr><td align="left" valign="top" bgcolor="&mvt:colors.ctgy_bg;">
    <table border="0" cellpadding="10" cellspacing="0">
    <tr><td align="left" valign="top">
      <mvt:item name="category_tree" />
    </td></tr>
    </table>
  </td><td align="left" valign="top" width="80%">
    <br>
    <blockquote>

      <a href="&mvt:global.sessionurl;">&#72;&#111;&#109;&#101;</a>

      <mvt:foreach iterator="crumb" array="breadcrumb">
        &#062; <a
href="&mvt:global.sessionurl;Screen=CTGY&Category_Code=&mvt:crumb.code;">&mvt:crumb.name;</a>
      </mvt:foreach> &#062; &mvt:product.name;

    <mvt:item name="prod_ctgy_hdf" param="prod_header" />

    <table border="0" cellpadding="5" cellspacing="0">
    <tr><td align="left" valign="middle" colspan="2">
```



```
<b>Related Item(s)</b><br>
</mvt:item>
```

```
<mvt:item name="product_list" />
</mvt:if>
<mvt:item name="prod_ctgy_hdft" param="prod_footer" />
</blockquote>
</td></tr>
```

```
<tr><td align="left" valign="bottom">
  <mvt:item name="hdft" param="footer" />
</td></tr>
</table>
```

```
<mvt:item name="hdft" param="global_footer" />
</mvt:item>
</html>
```

Attribute Template

The Product Template Manager module offers you the ability to use an Attribute StorMorph Template.

How it works

The module will create an array of attributes that can then be parsed by using a templates. (Template code below)

If you use the Attribute Template feature, be sure to remove the default attribute item tag from your page template. If you dont, you will have your attributes listed twice.

Template Code

```
<table border="0">
<mvt:foreach iterator="attr" array="productattributes">
  <mvt:if expr="l.settings:attr:type EQ 'checkbox'">
    <tr><td>
      &nbsp;
    </td><td valign="middle">
      <mvt:if expr="l.settings:attr:required">
        <b>
      </mvt:if>
      <input type="checkbox" name="Product_Attributes[&mvt:attr:pos;]:value" value="Yes"> &mvt:attr:prompt;
      <mvt:if expr="l.settings:attr:required">
        </b>
      </mvt:if>
    </td></tr>
    <input type="hidden" name="Product_Attributes[&mvt:attr:pos;]:code" value="&mvt:attr:code;">
  </mvt:if>
  <mvt:if expr="l.settings:attr:type EQ 'select'">
    <tr><td>
      <mvt:if expr="l.settings:attr:required">
        <b>
      </mvt:if>
      &mvt:attr:prompt; :
      <mvt:if expr="l.settings:attr:required">
        </b>
      </mvt:if>
    </td><td valign="middle">
      <select name="Product_Attributes[&mvt:attr:pos;]:value">
        <mvt:foreach iterator="option" array="attr:options">
          <option value="&mvt:option:code;"><mvt:if expr="l.settings:attr:default_id EQ l.settings:option:id">
            SELECTED</mvt:if><
          &mvt:option:prompt;
        </option>
        </mvt:foreach>
      </select>
    </td></tr>
    <input type="hidden" name="Product_Attributes[&mvt:attr:pos;]:code" value="&mvt:attr:code;">
  <mvt:if expr="NOT ISNULL l.settings:attr:template_code">
```

```

        <INPUT TYPE="hidden" NAME="Product_Attributes[&mvt:attr:pos;]:template_code" VALUE="&mvt:attr:template_code;">
    </mvt:if>
</mvt:if>
<mvt:if expr="l.settings:attr:type EQ 'radio'">
    <tr><td valign="top">
        <mvt:if expr="l.settings:attr:required">
            <b>
        </mvt:if>
        &mvt:attr:prompt; :
        <mvt:if expr="l.settings:attr:required">
            </b>
        </mvt:if>
    </td><td valign="middle">
        <mvt:foreach iterator="option" array="attr:options">
            <input type="radio" name="Product_Attributes[&mvt:attr:pos;]:value" value="&mvt:option:code;"<mvt:if
expr="l.settings:attr:default_id EQ l.settings:option:id"> CHECKED</mvt:if>
            &mvt:option:prompt;<br>
        </mvt:foreach>
    </td></tr>
    <input type="hidden" name="Product_Attributes[&mvt:attr:pos;]:code" value="&mvt:attr:code;">
    <mvt:if expr="NOT ISNULL l.settings:attr:template_code">
        <INPUT TYPE="hidden" NAME="Product_Attributes[&mvt:attr:pos;]:template_code" VALUE="&mvt:attr:template_code;">
    </mvt:if>
</mvt:if>
<mvt:if expr="l.settings:attr:type EQ 'text'">
    <tr><td>
        <mvt:if expr="l.settings:attr:required">
            <b>
        </mvt:if>
        &mvt:attr:prompt; :
        <mvt:if expr="l.settings:attr:required">
            </b>
        </mvt:if>
    </td><td valign="middle">
        <input type="text" size="40" name="Product_Attributes[&mvt:attr:pos;]:value">
    </td></tr>
    <input type="hidden" name="Product_Attributes[&mvt:attr:pos;]:code" value="&mvt:attr:code;">
    <mvt:if expr="NOT ISNULL l.settings:attr:template_code">
        <INPUT TYPE="hidden" NAME="Product_Attributes[&mvt:attr:pos;]:template_code" VALUE="&mvt:attr:template_code;">
    </mvt:if>
</mvt:if>
<mvt:if expr="l.settings:attr:type EQ 'memo'">
    <tr><td valign="top">
        <mvt:if expr="l.settings:attr:required">
            <b>
        </mvt:if>
        &mvt:attr:prompt; :
        <mvt:if expr="l.settings:attr:required">
            </b>
        </mvt:if>
    </td><td valign="middle">
        <textarea name="Product_Attributes[&mvt:attr:pos;]:value" rows="10" cols="58" WRAP="on"></textarea>
    </td></tr>
    <input type="hidden" name="Product_Attributes[&mvt:attr:pos;]:code" value="&mvt:attr:code;">
    <mvt:if expr="NOT ISNULL l.settings:attr:template_code">
        <INPUT TYPE="hidden" NAME="Product_Attributes[&mvt:attr:pos;]:template_code" VALUE="&mvt:attr:template_code;">
    </mvt:if>
</mvt:if>
</mvt:foreach>
<input type="hidden" name="Product_Attribute_Count" value="&mvt:attr_count;">
</table>

```

Available attribute fields:

&mvt:attr:template_code;
 &mvt:attr:prompt;
 &mvt:attr:price;
 &mvt:attr:formatted_price;
 &mvt:attr:cost;
 &mvt:attr:weight;
 &mvt:attr:type;

Available option fields:

&mvt:option:id;
 &mvt:option:code;
 &mvt:option:prompt;
 &mvt:option:price;
 &mvt:option:formatted_price;
 &mvt:option:cost;
 &mvt:option:weight;

&mvt:attr:required;
 &mvt:attr:image;
 &mvt:attr:default_id; – If an option is set to be the default, this will specify the options id.
 &mvt:attr:code;
 &mvt:attr:pos; – attributes numerical position

&mvt:option:image;

Example Attribute Image Display:

```
<mvt:if expr="l.settings:attr:image">
  
</mvt:if>
```

Using the Attribute Template on the Related Product Display:

You can use the attribute template feature on related products. For the most part the template will work in the same manner. The difference would be:

```
<mvt:foreach iterator="attr" array="product:productattributes">
```

```
<input type="hidden" name="Product_Attribute_Count" value="&mvt:product:attr_count;">
```

Dynamic Attribute Pricing

Dynamic attribute pricing allows you to update the product price on the fly as attributes are being selected. Lets say you have a base price of \$49.95 and an attribute that adds \$10.00 if selected. When the \$10.00 attribute is selected, the product price will automatically change to \$59.95. This is also works in conjunction with the quantity box. We do not offer support for this feature, but is offered for those who would like to try and use it.

Example: <http://www.sebenza.com/products/SS-PRODTEMPLATES5.html>

In order to use this feature, you will need to be using the attribute template feature mentioned above. So, make sure you create the “item” and assigned it to your PROD Page template. Example template code is included with the latest version of this module.

Available Product Quantities (Optional)

Within the modules product tab you will see a textbox for “Product Quantities”. Here you can specify the quantities the product is available in. Seperate each numerical value with a comma. For example: 2,4,6,8,10

Template code for this feature: Product Display

```
<mvt:if expr="l.settings:quantities">
  <select name="Quantity">
    <mvt:foreach iterator="qty" array="quantities">
      <option value="&mvt:qty:value;">&mvt:qty:value;</option>
    </mvt:foreach>
  </select>
</mvt:if>
<input type="text" name="Quantity" value=1 size=4>
</mvt:if>
```

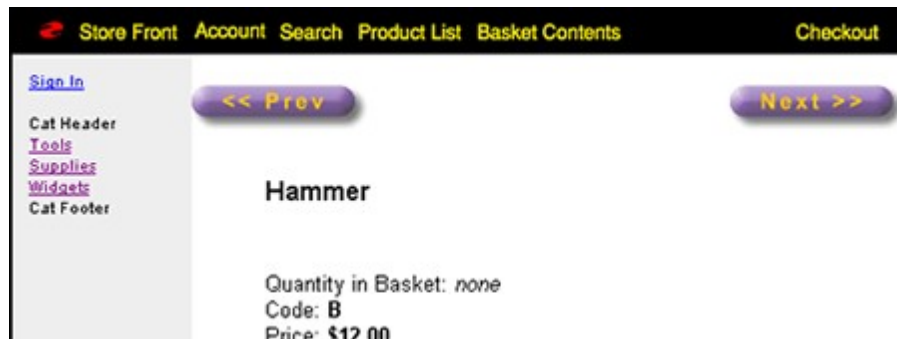
Template code for this feature: Basket Display

```
<mvt:if expr="l.settings:item:quantities">
  <select name="Quantity">
    <mvt:foreach iterator="qty" array="item:quantities">
      <option value="&mvt:qty:value;"><mvt:if expr="l.settings:item:quantity EQ l.settings:qty:value">
SELECTED</mvt:if>&mvt:qty:value;</option>
    </mvt:foreach>
  </select>
</mvt:if>
<input type="text" name="Quantity" value="&mvt:item:quantity;" size=4>
</mvt:if>
```

Note: You will need to assign the “xprodtemplate” item to your BASK Page Template under the “Items” tab.

Previous Product & Next Product

This feature will allow you to display options to your customer to view the previous or next product found in the current category.



Example template code for this feature:

```
<mvt:if expr="!l.settings:prevproduct OR !l.settings:nextproduct">
  <br>
  <table align="center" cellpadding="0" cellspacing="0" width="95%">
  <tbody>
  <tr><td align="left" width="50%">
    <mvt:if expr="!l.settings:prevproduct">
      <a
href="&mvt:global:sessionurl;Screen=PROD&Product_Code=&mvt:prevproduct:code;&Category_Code=&mvt:prevproduct:ctgy;">Previous Product</a>
    </mvt:if>
  </td><td align="right" width="50%">
    <mvt:if expr="!l.settings:nextproduct">
      <a
href="&mvt:global:sessionurl;Screen=PROD&Product_Code=&mvt:nextproduct:code;&Category_Code=&mvt:nextproduct:ctgy;">Next Product" alt="Next Product" border="0"></a>
    </mvt:if>
  </td></tr>
  </tbody>
  </table>
</mvt:if>
```

Parameter Functions

Parameter Functions allow you to execute MivaScript functions from the template level. For example, if we wanted to truncate our product description to 250 characters, we would use the following example:

```
<mvt:item name="xprodtemplate" param="substring(l.settings:product:descrip, 1, 250)" />
```

More API docs can be found here: <http://smallbusiness.miva.com/support/docs/api/>

Flat File Import

Included with the Product Template Manager is a Product Template Manager Import module. The module is installed via the "Modules" link in your admins left navigation. Once installed, the import control panel is located under Utilities > Import Data.

Your flat file should contain two columns: Product Code, Product Template ID. These two fields can be separated by a tab, pipe, or comma delimiter.

TroubleShooting

Runtime error in /mm5/5.00/modules/component/prod_templates5.mvc @ [00000030:0000005a]: F:MvWorkMvCOOLmm5SS-
PRODTEMPLATESprod_templates5.mv: Line 1409: MvDO: Unable to open '/mm5/5.00/templates/sbnz/prod-template-
1130797802.mvc': No such file or directory

The above error usually means that the template Merchant is trying to call has not been compiled. You can force a template to compile by going to the desired template in the module and hitting the "Update" button

License:

BY DOWNLOADING AND/OR IMPLEMENTING THIS SOFTWARE YOU AGREE TO THE FOLLOWING LICENSE:

Definitions:

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

(ii) "Module" (hereinafter "Module") 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.

1. This license is for the downloading, acquisition and use of Module and any permitted modifications of Module source code. By downloading, acquiring and using Module, YOU expressly agree to abide by this License.
2. This license is between LICENSEE and Sebenza. Licensee acknowledges that Sebenza 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. Sebenza 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. Licensee is granted a non-exclusive license to use Module in conjunction with one Miva Merchant Store or one Miva Merchant Domain (based upon Module and License purchased). License to use Module with more than one Miva Merchant Store or with more than one Miva Merchant Domain must be separately purchased from Sebenza or affiliate and is specifically not granted in this License.
4. Module contains valuable proprietary rights of Sebenza. There is no transfer to Licensee of any title to or ownership of Module or any copyright, trade name, trade secret, source code, or other proprietary rights relating to Module, regardless of the form that Module exists in. Licensee may not violate these rights, and Licensee must take all appropriate steps to protect Sebenza's rights. Sebenza may at any time replace, modify, alter, improve, enhance, or change Module without prior notice.
5. Any license, right, or permission to use Module 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 Sebenza, Licensee will immediately return any physical copies of Module to Sebenza, will cease any further use of Module, will remove the original and any modified or altered Module source code from his/her/its computers and drives, will cease his/her/its use of Module, and will cease any claimed breach of this License. Licensee 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 Licensee will not oppose any reasonable request for a temporary restraining order, preliminary injunction, or other relief sought by Sebenza in the event of a breach of this License. Sebenza 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 Sebenza to enforce its rights pursuant to this License shall not constitute a waiver of such rights, and shall not prejudice Sebenza in any later enforcement of its rights or rights to seek damages therefrom.
6. SEBENZA WILL NOT BE LIABLE FOR (A) ANY BUG, ERROR, OMISSION, DEFECT, DEFICIENCY, OR NONCONFORMITY IN MODULES; (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 SEBENZA 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 SEBENZA'S LIABILITY EXCEED THE PRICE THAT DEVELOPER PAID FOR MODULE.
7. This License is granted, Module was developed, technical support is being provided in, and Licensee is downloading this copy of Module via miscellaneous resellers, distributors, gateways and servers, and is paying for the use of Module, in and from the County of El Paso, State of Colorado, and the parties agree that this License is to be performed in the County of El Paso, Colorado.
8. This License Agreement shall be governed by the laws of the State of Colorado, 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.
9. Licensee acknowledges that Module 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 Module.
10. This License Agreement constitutes the entire and sole agreement between Licensee and Sebenza, and is agreed to and entered into on the date you download or implement Module, whichever date is earlier.