Bixie VirtueMart-Chronoform Module

www.bixie.org

By Matthijs Alles - (c)2010 Bixie
License: GNU/GPL

For Joomla! 1.5 and VirtueMart 1.1.3 to 1.1.7, Chrono Contact 3.x

More info: www.bixie.org/forum/vm-chrono-module.html

For support you can use the Bixie Support Center!

Rocketheme-template users, please check this topic.

Important hints:

- The module **must** be published in a position starting with '**vmchrono**'!
- You must use a seperate copy of the module for different forms.
- Always select the Settings-mode in module parameters to set up and check the module.
- Check your form in Chronoforms. Make sure the form submits okay, and you get a JSON-message like
  {"rec_id":"formname|","mclass":"error","formkey":" formname |4c25f011a9413"}.
- Check if the values are stored correctly in the forms database table (including the 'formkey').

Step-by-Step install shortlist:

- Install Bixie VM-Chrono module.
- Install Chronoforms component and plugin, publish the plugin.
- Install VirtueMart patches and check VirtueMart settings.
- Create a form in Chronoforms with the wizard, or load an example form from Bixie.
- Create a database table for the form using Chronoforms.
- Set up your VM-product: Load the form and module in the description and set product-attribute ('Form-id' by default).
- Set up the module: Set formname and VM-product attribute and select settings mode.
- Check settings and click apply to fix any problems.
- Set the module back to module-mode and you're ready to go.

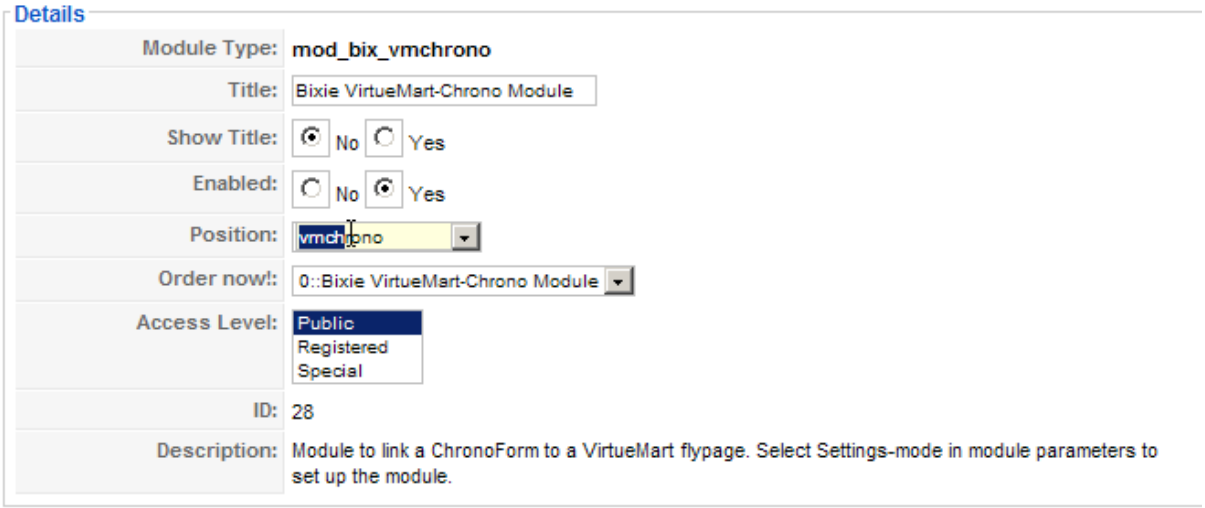
More detailed descriptions in document below.

If you use the VirtueMart-Chronoform module, please post a rating and a review at the Joomla! Extensions Directory!

INSTALLATION:

Install the module mod_bix_vmchrono.zip in the backend of Joomla (Extensions->Install/Uninstall).

Set module position to “vmchrono”. You can type this value in the pulldown-select (yes, you can!)



You can use several modules for different forms or settings. Create a different position for each module. The positionname must start with “vmchrono”! So use position “vmchrono_1” or “vmchrono_formname”.

Attention:

-Module mod_bix_vmchrono must be published in a position that starts with "vmchrono"!

Module Settings:

You can use a different set of settings for each moduleposition.

Setting:	Default:	Description:
Module Class Suffix		Default Joomla module class suffix
Module mode	Module	Select the "Settings" mode to set up the module. Set to "Module" for standard usage.
Chronoform name		(Unique) name of the Chronoform-form.
VM Product attribute		Name of the attribute added to the VirtueMart-product. Default: "Form-id". Case-sensitive!
Hide database fields	price_adjust	Comma-seperated list of fieldnames in database that should not be shown in cart-info
Currency symbol	€	Currency symbol displayed in form-info
Module text		Text in module. You can use HMTL. Leave empty to show no text.
Form required for order	Yes	Require the form to be sent before ordering the product.
Use ajax for addtocart	Yes	Set this setting according to your template settings in VirtueMart.
Show wait-status	Yes	Show wait-text on loading flypage.
Width message box	100%	The width of the message area in the module.
Add to cart after submit	No	Add the product to cart after submitting the form
Confirm add to cart	Yes	Ask a confirmation from the user before adding to cart
Hide VM-product attributes	No	Hide the complete product-attribute box. This also hides and disables any other attributes!
Hide addtocart form	No	Hide the VirtueMart addtocart-form and use the form-submit button as addtocart-button.

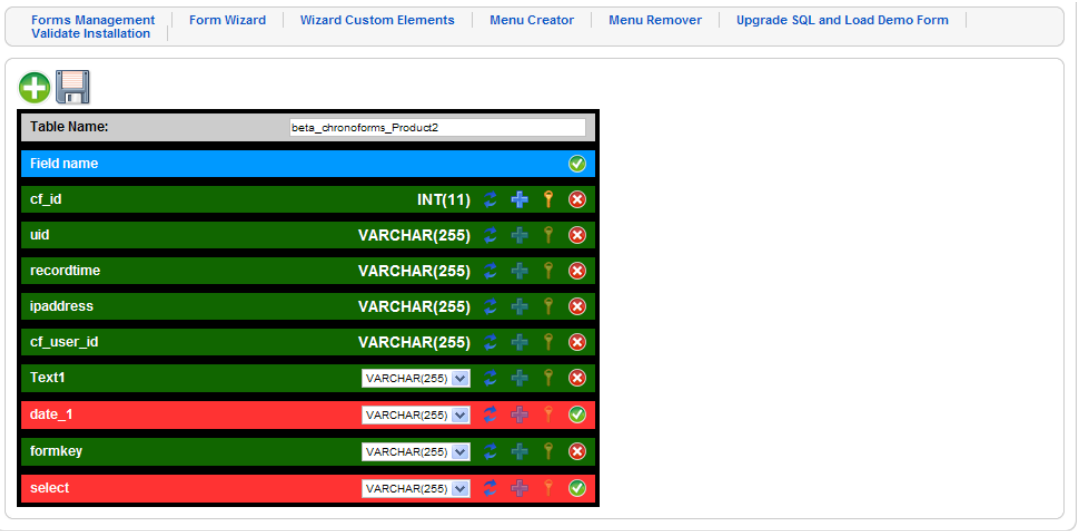
Install Chronoforms:

Install the component and plugin of Chronoforms. Don't forget to publish the plugin.

For more info and support, please visit their website.

SETTING UP YOUR CHRONOFORM

You can use the wizard of ChronoForms to create your form. Make sure you add a hidden field to the form with the name "formkey". Your form needs 2 other settings to work with the Bixie VM-Chronoform module. The redirect url must be set to "*index.php?option=com_chronocontact&chronoformname=bix_ajax_return&tmpl=component*". In the "form tag attachment" you must set: *onsubmit="BixFormUpdate()"* .
You don't have to add these settings yourself, the Bixie VM-Chronoform module will set it for you when it's in the "Settings"-mode.
You can add price changes to the form fields when you are using pulldown-selects. Place a "|" separator behind your values and then add your adjustments (=10,+5,-2.5). Use prices without tax! If you want to change the text in the pulldown menu to a better looking value, edit the form code when you're done with the wizard.
When your form is finished you will have to attach it to a database table. Use Chronoform to create a table. Be sure to check all fields you want to be available in VirtueMart so they get a green background color. Leave the default fields enabled. Make sure the hidden field "formkey" is in the table. Save the table.



In the form-settings, set the option "Enable Data storage" to yes and select the correct table. Make sure the "Saving Data/Emails order" is set to "After email" if you use email notification.

Attention:
-You can download the demoforms from the Bixie website to use as a base!
-More info and support on creating forms on the ChronoEngine website.

VIRTUEMART SETTINGS:

These settings are done in the backend-section of the VirtueMart component. Add these settings to each product you want to attach a form to.

First make sure the option "Enable content mambots / plugins in descriptions?" in the Global VirtueMart config is checked. Check that the native Joomla-plugin "Load module" is published.

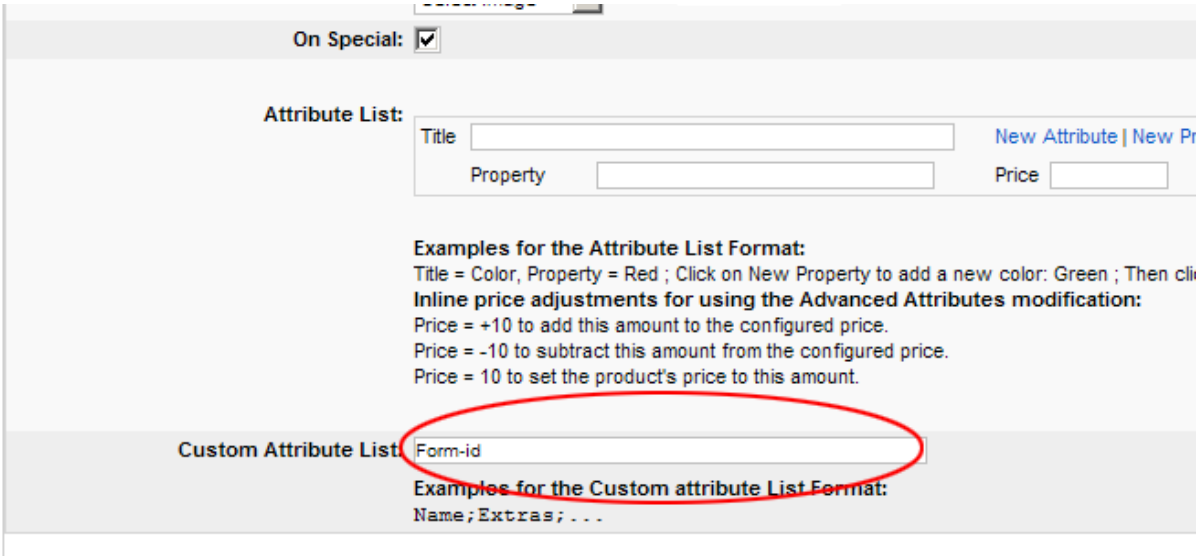
To add your Chronoforms form to the flypage you have to use the Chronoform-plugin, which can be downloaded here. Install and publish the plugin.

In ***product edit***:
- Now add the form and the VM-Chronoform module to your flypage using these tags in the **Product Description**:

If the module is published in a different position than 'vmchrono', change it here.

```
{chronocontact}Formname{/chronocontact}
{loadposition vmchrono }
```

-Add an "**Custom Attribute List**" with the name "Form-id".
*This is done in '**Custom Attribute List**' on the tab 'Product status'. You can use a different name, but you must change the setting in the VM-Chronoform module settings accordingly.*



Attention:
-Module mod_bix_vmchrono must be published in a position that starts with "vmchrono".
-The option **Enable content mambots / plugins in descriptions?** in the Global VirtueMart config must be checked.
- If you use spaces in the attribute, replace them with underscores (_) in the module settings!

VIRTUEMART PATCHES:

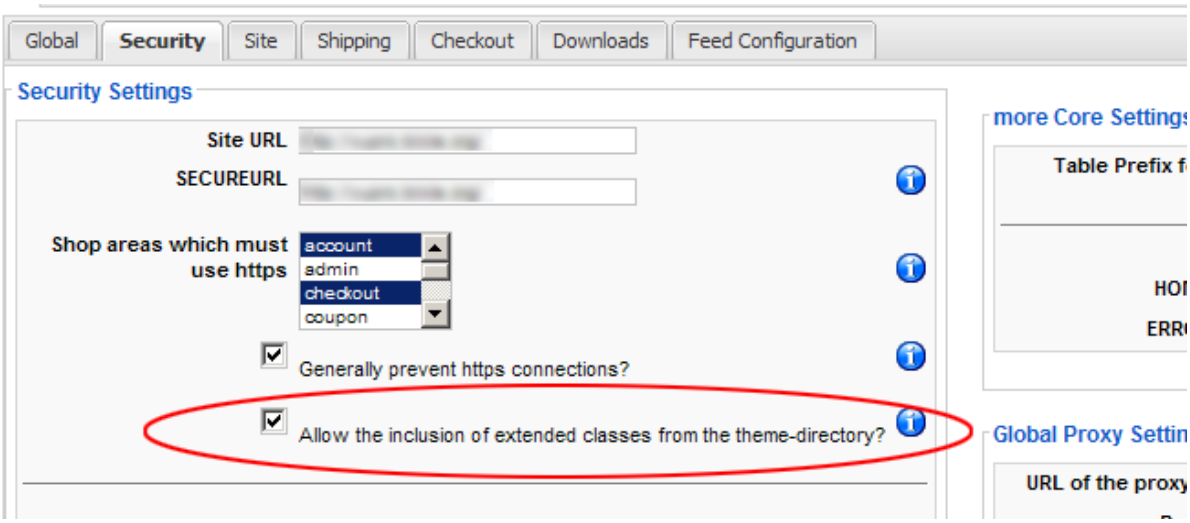
Installing patch

Go to the VirtueMart backend, to Admin menu option "Check for updates". Then choose the second tab "Upload a patch". Click browse and select the right patch for your VirtueMart version. If you are using Bixie VirtueUpload (Pro) as well, select the correct patch. Click "Upload & Preview" to proceed.
This is essentially an update patch for VirtueMart. Make sure you use the right files for your VirtueMart version! A list of files is shown. Click "Apply patch now" to overwrite your files.
To see the form-info in the VM backend, one administrator file is changed. Remember that this change is undone when VM is updated.
-In folder administrator/components/com_virtuemart/html

- order.order_print.php

Attention:
In VirtueMart 1.1.4, make sure you apply the Make custom attributes optional-hack!

Attention:
-There are separate patches for VirtueMart 1.1.3 to VirtueMart 1.1.6! Make sure you use the right ones!
-This only works on a standard VirtueMart 1.1.3 to 1.1.6 installation in default-theme. If you use a customized VirtueMart or custom theme, use the manual adjustments described below.
-Always make a backup before overwriting your files!
-From VirtueMart 1.1.5 the user classes are used. This means that there are no core-files of VirtueMart are changed. All changes are made in the template directory. This way the changes are preserved when you update VirtueMart.
This means you have to allow the user classes. This is done in the "security" tab of the global VirtueMart configuration.



Manual adjustments

These adjustments do not need to be made when you installed the VirtueMart patch as described above.

The changes you must make depend on whether you are using only the Bixie VirtueMart-Chronoform Module, or you're using it in combination with Bixie VirtueUpload (Pro).

These templatefiles must be changed:
-In folder components/com_virtuemart/themes/yourthemename/templates

- /basket/basket_b2c.html.php
- /basket/ro_basket_b2c.html.php
- /pages/account.order_details.php
- /order_emails/confirmation_email.tpl.php

To see the form-info in the VM backend, you will have to change 1 administrator file
-In folder administrator/components/com_virtuemart/html

- order.order_print.php

Place on top of **each templatefile that you change**, just below _JEXEC check:

```
if (file_exists (JPATH_SITE.DS."modules".DS."mod_bix_vmchrono".DS."helper.php")) {
        require_once (JPATH_SITE.DS."modules".DS."mod_bix_vmchrono".DS."helper.php");
}
```

This includes the VM-Chronoform modulehelper in the file.

The VM-classes below only need to be changed when you need price-changing.
For VirtueMart 1.1.3
-In folder administrator/components/com_virtuemart

- /classes/ps_product. php

For VirtueMart 1.1.4 and higher
-In folder components/com_virtuemart/themes/yourthemename

- /user_class/ps_product.php

(create folder and file if not exists, also see security settings on previous page)

VirtueMart 1.1.4, make sure you apply the Make custom attributes optional-hack!

Templatefiles

In ***basket/basket_b2c.html.php*** (line 33) and ***basket/ro_basket_b2c.html.php*** (line 36) (same for vm1.1.3 and vm1.1.4):,

Replace line:

```
<td><?php echo $product['product_name'] . $product['product_attributes' ?></td>
```

With:

```
<td><?php echo modBixvmChronoHelper::getFormValues( $product['product_name']).
$product['product_attributes'] ?></td>
```

If you are also using Bixie VirtueUpload (Pro), you must replace it with:

```
<td><?php echo modBixvmChronoHelper::getFormValues( $product['product_name'] ).
VUOutput::UploadInfo ( $product['product_name'] , 'cart').$product['product_attributes']
?></td>
```

The same for ***basket/basket_b2b.html.php*** and ***basket/ro_basket_b2b.html.php*** if you are using secure urls.

In ***pages/account.order_details.php***

- Starting from line 361 you see (same for vm1.1.3 and vm1.1.4)::

```
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
                                    if( $link_to_product) {
                                    echo '<a href="'.$sess->url(
$mm_action_url."index.php?page=shop.product_details&product_id=$product_id") .'" title="'.$dbcart-
>f("order_item_name").'">';
                                    }
                                    $dbcart->p("order_item_name");
                                    echo " <div style=\"font-size:smaller;\">" . $dbcart-
>f("product_attribute") . "</div>";
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
                                    if( $link_to_product) {
                                            echo "</a>";
                                    }
```

Replace that bit with:

```
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
        if( $link_to_product) {
                $enda = "</a>";
                }
                if( $link_to_product) {
echo '<a href="'.$sess->url( $mm_action_url."index.php?page=shop.product_details&product_id=$product_id") .'
" title="'.$dbcart->f("order_item_name").'">';
                }
        $dbcart->p("order_item_name");
echo $enda." <div style=\"font-size:smaller;\">" . modBixvmChronoHelper::getFormValues( $dbcart-
>f("product_attribute") ) . "</div>";
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
```

If you are also using Bixie VirtueUpload (Pro), you must replace it with:

```
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
        if( $link_to_product) {
                $enda = "</a>";
                }
                if( $link_to_product) {
echo '<a href="'.$sess->url( $mm_action_url."index.php?page=shop.product_details&product_id=$product_id") .'
" title="'.$dbcart->f("order_item_name").'">';
                }
        $dbcart->p("order_item_name");
echo $enda." <div style=\"font-size:smaller;\">" . modBixvmChronoHelper::getFormValues( $dbcart-
>f("product_attribute") ) .VUOutput::UploadInfo ( $dbcart->f("product_attribute") , 'cart' ) . "</div>";
// ***** Change guard to use new flag $link_to_product instead of !empty( $product_id )
```

In **order_emails/*confirmation_email.tpl.php*** line 184 (same for vm1.1.3 and vm1.1.4).
Replace line:

```
<td><?php $dboi->p("product_name")?>
<?php echo ($dboi->f("product_attribute") ? ' ('.$dboi->f("product_attribute").')' : ''); ?></td>
```

With:

```
<td><?php $dboi->p("product_name")?> <?php echo modBixvmChronoHelper::getFormValues( $dboi->f(
"product_attribute") ); ?>
```

If you are also using Bixie VirtueUpload (Pro), you must replace it with:

```
<td><?php $dboi->p("product_name")?> <?php echo modBixvmChronoHelper::getFormValues( $dboi->f(
"product_attribute") ) . VUOutput::UploadInfo ( $dboi->f("product_attribute") , 'mail' ); ?>
```

In **order.order_print.php** line 319 (same for vm1.1.3 and vm1.1.4).
Replace line:

```
echo "<br /><span style=\"font-size: smaller;\">" . ps_product::getDescriptionWithTax($dbt-
>f("product_attribute")) . "</span>";
```

With:

```
echo "<br /><span style=\"font-size: smaller;\">" . modBixvmChronoHelper::getFormValues(
Array("product_name"=>$dbt->f("product_attribute"),"product_id"=>$dbt->f("product_id") ) ).
"</span>";
```

If you are also using Bixie VirtueUpload (Pro), you must replace it with:

```
echo "<br /><span style=\"font-size: smaller;\">" . modBixvmChronoHelper::getFormValues(
Array("product_name"=>$dbt->f("product_attribute"),"product_id"=>$dbt->f("product_id") )
).VUOutput::UploadInfo ( $dbt->f("product_attribute") , 'cart' ) . "</span>";
```

VirtueMart classes

For VirtueMart 1.1.3:
You must hack one VirtueMart core file. Remember that this will be lost when you update VirtueMart.

- In administrator/components/com_virtuemart/classes/***ps_product.php***, line 1813
  Just below these lines:

```
                                        $setprice += $modifier;
                                        $set_price = true;
                                    }
                                }
                            } else {
                                continue;
                            }
```

Add these lines:

```
//Bixie VM-Chrono price adjustment
if (file_exists  JPATH_SITE.DS."modules".DS."mod_bix_vmchrono".DS."helper.php")) {
      require_once (JPATH_SITE.DS."modules".DS."mod_bix_vmchrono".DS."helper.php");
}

$bix_priceadjustment = modBixvmChronoHelper::getPriceInfo($description);
if ($bix_priceadjustment->apply == true) {
      $adjustment += $bix_priceadjustment->adjustment;
      $setprice = $bix_priceadjustment->setprice;
      $set_price = $bix_priceadjustment->set_price;
}
//end Bixie
```

For VirtueMart 1.1.4:
Add a user_class folder to your theme folder and create a new file called ***ps_product.php***.
This file extends the VM-class and replaces the regular VM-function get_adjusted_attribute_price().

You can find the correct file in the VM 1.1.4 patch zipfile!
components\com_virtuemart\themes\default\user_class\ps_product.php