

# ProxiBlue Dynamic Category Products

Thank you for purchasing our product.

Support, and any queries, please log a support request via <http://support.proxiblue.com.au>

**If you are upgrading from a pre v3 version, please see the upgrade path section at the end of this document.**

**If you are upgrading from a pre v4.9.0 version, please see the upgrade path section at the end of this document for upgrade to 4.9.0, or later**

**CRON / Scheduled Builds has changed from version 4.9.0 - Please read the section 'General Extension Configuration' and configure your rebuild cron.**

**Product Tab has changed from version 4.9.0 Please read the section on Product Tabs if you are upgrading from a previous version**

## Install

- Access your My Account area on our website ([www.proxiblue.com.au](http://www.proxiblue.com.au))
- Select the version you have purchased (later versions may have been added as they get released)
- Download to a folder on your computer.
- Unzip the downloaded file.
- Disable Compilation if enabled
- Copy the contents of the folder to the root of your magento website root folder.
  - In most instances this is achieved using FTP
  - Contact your IT support to assist you in this procedure if you do not know how to update your site.
  - If you require install assistance, please purchase an install service option here: <http://www.proxiblue.com.au/premium-services/install-service.html>
- If this is a new install, no files will be replaced.
- If this is an upgrade to a newer version, allow overwrite of all files.
- If you have customized any sections, or layouts or templates, confirm your custom versions contain all the latest functionality.
- Clear all cache
- Enable compilation if you use that feature in magento

## Overview

This module will enable you to dynamically assign products to display in a category. The category is a default magento category and all magento category related functionality still applies.

This module functionality allows you to create categories such as 'Shop by Brands', 'Shop by Color', 'Shop all Special Priced Products / Sale Category', 'Shop by all Black and Green products'.

It really is limited to your imagination.

No longer do you need to manually update, spending a lot of time managing, your category products.

## General Extension Configuration:

Under System->Configuration->Dynamic Category Products, you will find the following admin options:

### *Rebuild Options:*

- Use External Cron Configuration: **\*\*\*\* Please read this section \*\*\*\***

The extension rebuilds dynamic categories via 3 cron jobs. These crons used to run inside magento's cron system, however, it has been found that they can run for extended times, especially when there is a large catalog. This can have negative effects on other magento cron jobs which get delayed, or never completed.

**From version 4.9.0, the preferred rebuild option is to use external cron jobs, via your hosting server. This option will thus default to YES.**

You can set to this NO, and the extension will use the older way of running crons inside magento. If you choose to do so, you do not need to configure external cron.

*The three cron jobs are:*

Rebuild ALL (daily)

Command to run: <MAGENTO ROOT>/shell/rebuildDynamic.php --type all

Schedule: 0 0 \* \* \*

Rebuild changed (every 10 minutes)

Command to run: <MAGENTO ROOT>/shell/rebuildDynamic.php --type changed

Schedule: \*/10 \* \* \* \*

Rebuild delayed (every 5 minutes)

Command to run: <MAGENTO ROOT>/shell/rebuildDynamic.php --type delayed

Schedule: \*/5 \* \* \* \*

Please contact your IT Support Team, or service provider to help you setup crons on your hosting environment / server

Please note that <MAGENTO ROOT> above is the full path to your magento root install folder. On CPANEL this will be ~/public\_html/ or /home/<USER>/public\_html/

On CPANEL, follow this guide:

<https://documentation.cpanel.net/display/ALD/Configure+cPanel+Cron+Jobs>

See end of this document for some images showing CPANEL configuration.

ProxiBlue offers an Install Service. Please purchase this service if you need us to configure the cron for you.

- Disable Dynamic Build on Save :

This affects how the extension deals with products that get edited (attribute values that get changed) Best result is to leave this NO. However if you find admin area performance issues, you can set this to Yes. Note that setting to yes will only make your products appear in their categories when teh dailt (--type all) cron runs.

- Disable Time to execute Workaround:

Some hosting providers do not allow magento to set the time a script can run. Disable this if your cron jobs seem to not acting correctly, or you get errors about stting an invalid option in php. Generally happens if PHP is in safe mode.

- Time to execute workaround:

The longest time a cron script may run. In seconds. If you get timeout errors, try an increase this value.

- Delay build of category dynamic products on category save:

If you have a large (really large) catalog, then saving the category may cause magento to process for a long time. This can delay your work. Enable this setting to delay the rebuild by 5 minutes. You can then continue work, and magento will rebuild the products via cron after 5 minutes using the --type delayed cron

- Disable indexing whilst cron jobs run:

Can improve cron performance. Normal magento behaviour is to re-index as soon as a category saves. If you are rebuilding more than one category (which usually happens via cron) it can improve performance to temporarily disable crons, rebuild and save all categories, and once completed, re-enable indexing. This will result in indexing only re-building at the end, but can result in a longer running index process.

Depending on the server configuration, and resources, it may be beneficial to do it all at the end, however on some systems it is best to do it 'per save' and have the process take longer in total, but split up indexers to each save.

- Completely ignore indexing:

The module will not deal with indexing at all. No indexing will be initiated by the extension. Indexer modes will not be changed.

### *Products Tab: (NOTE THIS HAS CHANGED FROM VERSION 4.9.0)*

The module supports two tab modes:

*Merged:* This mode will merge the Category Products Tab display with the Dynamic Rules display, creating one tab that has the rules above the products. This is the preferred mode.

*Standalone:* This will display two separate tabs. The usual products tab, and a separate tab for the product rules. This is a backup mode, in case you have other 3rd party extensions installed which also adjusts the category display tab.

NOTE: From version 4.9.0, the extension is 100% compatible with the FREE Enhanced Admin Grids Editor, which will allow you to place custom columns onto your products grid.

<http://www.magentocommerce.com/magento-connect/enhanced-admin-grids-editor.html>

If you have used the Enhanced Admin Grids Editor prior to version 4.9.0, and you get an alert that the Enhanced Grid could not rewrite the Dynamic Grid, then you must run the following SQL Query against your database:

```
DELETE FROM `customgrid_grid` WHERE `block_type` =  
'dyncatprod/adminhtml_catalog_category_tab_product';
```

This will remove the alert. Functionality will be fine if you don't do this, but the alert can be annoying!

The (old) 'enhanced grid', used prior to 4.9.0, is no longer available / included. If you need more advanced grid display, please use the before mentioned Advanced Admin Grids extension.

The extension does add in a new column, called 'Dynamic', which will indicate that a product is dynamic, or manual. If it is dynamic, it will show a green tick. This column will be added irrespective to the mode selected.

Price	Position	Dynamic
From: <input type="text"/>	From: <input type="text"/>	
To: <input type="text"/>	To: <input type="text"/>	
\$50.00	103 <input type="text" value="103"/>	<input checked="" type="checkbox"/>
\$50.00	102 <input type="text" value="102"/>	<input checked="" type="checkbox"/>
\$150.00	101 <input type="text" value="101"/>	<input checked="" type="checkbox"/>

**TROUBLESHOOT TIP:** The module protects your store from any unforeseen errors. If a serious error occurs in the processing of any rules, the Merged Tab will disappear, and the error logged. If you find that the Merged tab has gone missing, set this option to Standalone, which will restore the default core magento products tab. This will allow you to manually manage your category, whilst proxiblue support investigates the error encountered.

Please note that if you are upgrading to version 4.9.0+, you may also need to delete two files from the extension folder.

If the following files exist in your installed location, please delete them:

app/code/community/ProxiBlue/DynCatProd/Block/Adminhtml/Catalog/Category/Tab/Product.php  
app/code/community/ProxiBlue/DynCatProd/controllers/Catalog/CategoryController.php

### *Global Rule Options:*

You can set global rule options in this section. Each option has a comment in admin, which fully describes the option functionality.

### *Notification Options:*

Configure here how/when to send email notification. Please make contact if you need additional events added.

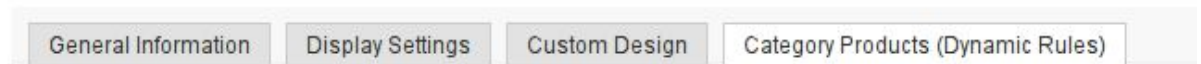
## Debug options:

Ideally the debug options must be disabled in a production environment. Your developers can use this option to view debug information on what the module is doing in the system.log of magento.

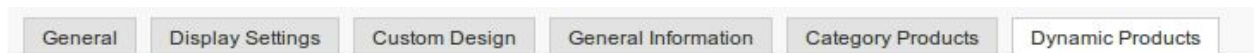
## Using the Admin:

### 1. Create a new Dynamic Category Product Listing.

- Access the Category Manager grid. (Catalog->Manage Categories)
- Select / Edit / Create a category.
- On the top 'Tabs' you will see a products tab
  - If you are using MERGED mode you will see a tab as follows:



- If you are using STANDALONE mode, you will see a separate tab called 'Dynamic Rules'
- Select the tab.
- You will be presented with the Dynamic Products Rule Manager.
- This works exactly as the normal magento promotions rules.



If you view a category that has Child Categories, you will see a rule area for Parent Rules. Parent defined rules will automatically be applied to all child categories.



- To define a new rule, simply click the + and follow the prompts. If you have used magento core Catalog / Cart rules, you will be familiar with the interface, as it works the same.
- You can use any combination of rules together. There is no limit. You do not have to

worry about setting the rules in the correct order, the extension will work the correct order out for you when the rules are run. After saving the rules will be displayed in the order that they are processed by the module.

## Setting Ordering Position:

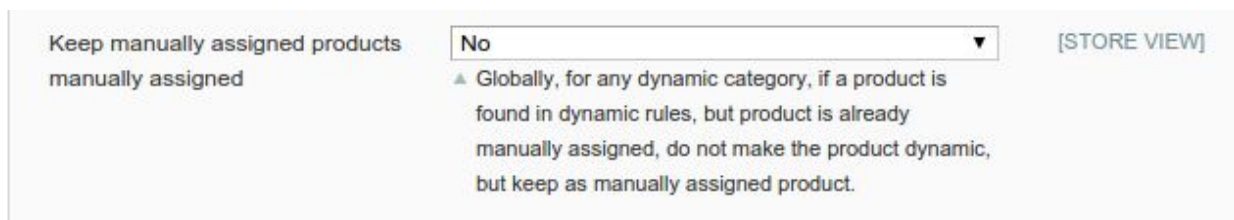
From v3 it is possible to set the ordering position for dynamic assigned products in the admin. If the category is rebuilt, the order value of any products that remain in the category will be re-used.

## Mixing Dynamic and Manually added products:

You can mix manually added products with dynamic added products. The manually added products will always remain in the category, until manually removed.

If a product is manually assigned to a category, and then a rule also assigns the product to the category, then the product will no longer be manually assigned, but become dynamically assigned. Once the dynamic rule no longer matches that product, it will be removed from the category.

If you want manually assigned product to remain manually assigned, please set the required option under Global Rule Options.



The screenshot shows a configuration option for 'Keep manually assigned products manually assigned'. The dropdown menu is currently set to 'No'. Below the dropdown, there is a tooltip that reads: '▲ Globally, for any dynamic category, if a product is found in dynamic rules, but product is already manually assigned, do not make the product dynamic, but keep as manually assigned product.' To the right of the dropdown is a '[STORE VIEW]' link.

## Using wildcards in textual fields

When using the a filter with textual fields, you can add start/end wildcards (denoted by a %) to the textual field value. If you place no wildcard, then the entire word or phrase will be enclosed in wildcards. Thus without wildcards, any string will be match if part of another string.

*Example:*

%shirt will find any match where the word shirt is preceded by some text

shirt% will find any match where the word shirt is followed by some text

%shirt% (the default if no wildcards given) will find any text that has the word shirt anywhere

## Using dates in textual fields

You may need to find products that have textual date entries in their names, or descriptions. You can insert a dynamic date into any text field using a date placeholder.

The format is:

```
{{date(FORMATTING OPTIONS)+|-DAYS}}
```

*Example:*

`{{date(Ymd)+1}}` will result in the current date PLUS one day. in the format of YYYYMMDD  
`{{date(l jS \of F Y h:i:s A)+1}}` which will result in the date string : Saturday 2nd of May 2015 09:25:22 AM (as run on the 1st May 2015)

The date format option can be any of the PHP DATE function options:

[http://www.w3schools.com/php/func\\_date\\_date.asp](http://www.w3schools.com/php/func_date_date.asp)

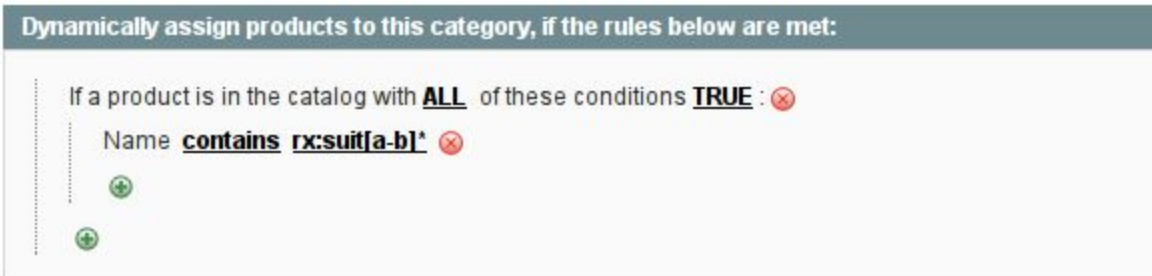
## Using Regular Expressions:

You can use **regular expressions** to match string values in product attributes (example use on Name, Description or any textual attribute)


Regular expression patterns follow standard MYSQL Pattern Matching rules:


<http://dev.mysql.com/doc/refman/5.7/en/pattern-matching.html>


To invoke regular expression matching simply prefix any string with rx:




Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions **TRUE** : 

Name **contains** **rx:suit[a-b]\*** 





## Dynamic Variables

It is possible to use *Dynamic Variables* in the rule values.

An example is to find all products that has the category name in the product name.



**Dynamically assign products to this category, if the rules below are met:**

If a product is in the catalog with **ALL** of these conditions **TRUE** : ❌

Name **contains** **{{CATEGORY.NAME}}** ❌ ?

+

+

Variables are placed within double {{ }} curly brackets  
 Currently the following dynamic variables are supported:

**Category Attributes:**

Variable format: {{CATEGORY.ATTRIBUTE}} where ATTRIBUTE is any of the category attributes.

Examples :

- {{CATEGORY>NAME}} - match the category name
- {{CATEGORY>URL\_KEY}} - match the url key of the category

## Parent / Global Category Rules:

You can place rules on Parent Categories, which will then be applied to all child categories, recursively.

You can set child categories to exclude parent rules.

If you place a rule on the Top Root/Default Category, every category in your catalog will get that rule applied, making it 100% global

Below a view of the default category. This rule will filter down to the entire catalog.

General Information | Display Settings | Custom Design | **Products (Enhanced)**

⚠ You are viewing a category with child categories. All child categories will automatically include the PARENT rules. You can disable this on child category.

**Parent Category Rules:**

If a product is in the catalog with **ALL** of these conditions **TRUE** : ❌

Associated Website **contains** **Main Website** ❌

+

+

**Dynamically assign products to this category:**

+

below you can see the rule defined in the parent category, being noted that it will be applied to the category.

This is also a parent category, so itself can define parent rules, for only its child categories.

The screenshot displays the configuration interface for a category named "Women (ID: 4)". At the top, there are tabs for "General Information", "Display Settings", "Custom Design", and "Products (Enhanced)". A red banner states: "This category will include rules defined in parent categories:". Below this, a section titled "From: Default Category" shows a rule: "If a product is in the catalog with ALL of these conditions TRUE: ==> Associated Website contains Main Website". A red note next to it says "a parent category, that also has a parent with rules". A yellow warning icon with text reads: "You are viewing a category with child categories. All child categories will automatically include the PARENT rules. You can disable this on child category." The "Parent Category Rules:" section lists a rule: "If a product is in the catalog with ALL of these conditions TRUE : ⓧ" with a sub-condition "Attribute Set contains Clothing ⓧ". A red note says "The rules define for this parent, and all its children". The "Dynamically assign products to this category:" section shows a single green plus icon. A red note says "rules for this category".

One of the lowest level categories, showing it will apply rules from all the parents.

New Arrivals (ID: 10)

General Information
Display Settings
Custom Design
Products (Enhanced)

**This category will include rules defined in parent categories:**

**From: Default Category**

If a product is in the catalog with ALL of these conditions TRUE:  
 ==> Associated Website contains Main Website

**From: Women**

If a product is in the catalog with ALL of these conditions TRUE:  
 ==> Attribute Set contains Clothing

**Dynamically assign products to this category:**

Transform the resulting data using the following rules: ✖

- + If a complex product type was found then also add its associated p... ✖
- +
- +

## Using 'Sales Reports' Rules:

These rules hang directly off your store 'Sales Reporting'. If your magento store is correctly configured for cron, your sales reports will update daily, which then means the results for these rules will also update daily. If your Sales Reporting statistics are not updated, then the results given in the rules will not be up-to-date.

If you find that these rules do not result as expected, check that your Report Statistics update daily.

Due to the nature of the Sales Report, you cannot set manual positions for the dynamic items. If you want to re-order items in the result, you must exclude the items via the rules (example: 'SKU not in' rule added to the rule set) and manually add the items you wish to reorder, as non dynamic items.

### Dynamic Updates:

Each evening at 00:00, the module will run a scheduled cron job, which will rebuild the Dynamic Products of all categories that has Dynamic Rules set. This will ensure your categories are always up-to-date, especially regarding date based rules.

After a product is edited, and any attribute value is changed, the module will schedule an update of all categories that has Dynamic Rules based on the changed attribute. This will ensure any product updates will dynamically adjust the products associated to any category.

## Date Range Attribute Transformation

Magento comes with a few 'Date Range' attributes that work in pairs. You will know some of these as the New From and New To attributes which allows you to define the start and end dates that a product will be set as new.

The module caters for these type of range attributes by combining them into one attribute, which has then the ability to display products where the date range of From and To is valid for the day on which the category is viewed.

Thus, by example:

If a product has the New From Date set at 1/1/2014 and the New To date set as 10/1/2014, then the product will appear in the category on and between those two dates (thus a Date Range)

The products are re-calculated every day at 00:00, thus will be up-to-date on the day of viewing.

The module will automatically do this for any attribute (pairs) where the attribute code conforms to the standard magento attribute code naming convention for these types of attributes. Thus any attribute (from a 3rd party extension) where the attribute code (pairs) are named as xxxxxx\_to and xxxxxx\_from will automatically be transformed.

But that is living in an ideal world, and we do not. You may find 3rd party extensions that break from the standards, and place date range attributes that do not use \_from and \_to in the attribute code naming. An example may be 'start\_date' and 'end\_date'

To allow you to transform non standard named (pairs) of date range attributes, the extension has the ability to drop in an xml file into the etc folder of the module (<magento root>/app/code/community/ProxiBlue/DynCatProd/etc)

The xml file allows you to define the pairs of attribute code that must be transformed into the date range value, thus allowing for any attribute to work as the core attributes.

The xml file is not distributed with the extension, thus allowing for the file to not be replaced if

you update the extension.

You can find an example of the file in the downloadable section of your order.

The format is as follows:

```
<date_range>
  <transform>
    <description>LABEL THAT WILL APPEAR IN DROPDOWN IN RULES</description>
    <code>
      <start>START DATE ATTRIBUTE CODE</start>
      <end>END DATE ATTRIBUTE CODE</end>
    </code>
  </transform>
</date_range>
```

## Import / Export Rules

Once installed the extension will insert two advanced dataflow import / export profiles.  
You can access this via the menu option: System->Import/Export->Dataflow Advanced Profiles

you will see this screen:

**Advanced Profiles**

---

Page 1 of 1 pages | View 20 per page

ID	Profile Name
8	Import Dynamic Category Rules
7	Export Dynamic Category Rules

To export:

Select the export profile.  
You will see the profile XML as the first view.

```
Profile Name *      Export Dynamic Category Rules
Actions XML *      <action type="dyncatprod/convert_parser_category" method="unparse">
                   <var name="store"><![CDATA[1]]></var> ← set store id
                   </action>
                   <action type="dyncatprod/convert_parser_csv" method="unparse">
                   <var name="delimiter"><![CDATA[,]></var>
                   <var name="enclose"><![CDATA[""]></var>
                   </action>
                   <action type="dataflow/convert_adapter_io" method="save">
                   <var name="type">file</var>
                   <var name="path">var/export</var> ← set export path
                   <var name="filename"><![CDATA[dyncatprod.csv]]></var>
                   </action>
```

Here you can set the store id (defaults to 1) and the export path (defaults to magento default of var/export)

If you make any edits, remember to save

Next you must click on the 'Run Profile' tab  
You will be presented with the profile run button.  
Simply click this button to start.

You will be presented with the export progress display.

```
Starting profile execution, please wait...
Warning: Please do not close the window during importing/exporting data
Starting ProxiBlue_DynCatProd_Model_Convert_Parser_Category :: unparse
Starting ProxiBlue_DynCatProd_Model_Convert_Parser_Csv :: unparse
Starting Mage_Dataflow_Model_Convert_Adapter_Io :: save
Saved successfully: "dyncatprod.csv" [8636 byte(s)].
Finished profile execution.
```

Once complete, you will find the exported .csv file in the configured folder (normally var/export from the magento root folder)

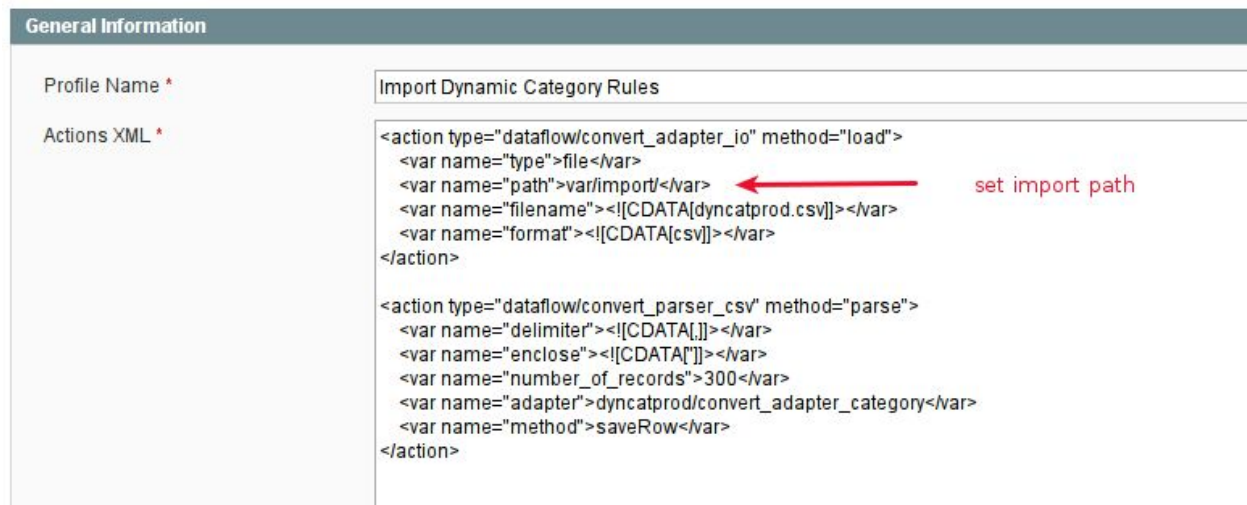
The exported file is called **dyncatprod.csv**

## To Import

**ATTENTION:** The source and destination must share the same category structure, and specifically the same category ids.

Select the export profile.

You will see the profile XML as the first view.



**General Information**

Profile Name \* Import Dynamic Category Rules

Actions XML \*

```
<action type="dataflow/convert_adapter_io" method="load">
  <var name="type">file</var>
  <var name="path">var/import/</var>
  <var name="filename"><![CDATA[dyncatprod.csv]]></var>
  <var name="format"><![CDATA[csv]]></var>
</action>

<action type="dataflow/convert_parser_csv" method="parse">
  <var name="delimiter"><![CDATA[,]></var>
  <var name="enclose"><![CDATA["]]></var>
  <var name="number_of_records">300</var>
  <var name="adapter">dyncatprod/convert_adapter_category</var>
  <var name="method">saveRow</var>
</action>
```

set import path

You can set the import path here. The default is standard magento of var/import  
If you make any changes, please remember to save.

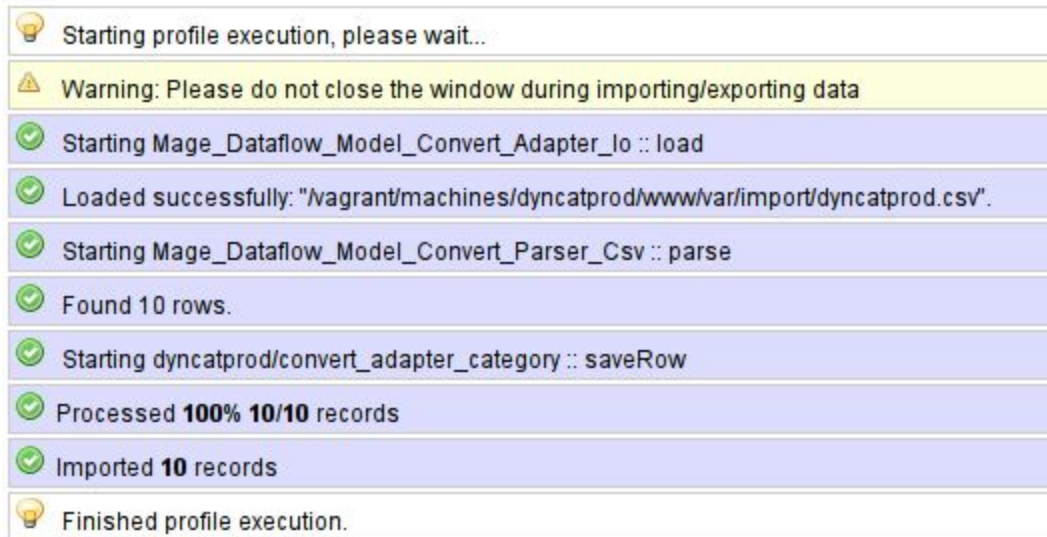
First copy your previously exported dyncatprod.csv file to the configured import folder.

Next you must click on the 'Run Profile' tab

You will be presented with the profile run button.

Simply click this button to start.

You will be presented with the import progress display.



Once completed you have imported the rules.  
They must still be actioned, which can be done via ron, or via the command line tool, or simply by saving the category in admin.

## Copy / Copy and Paste Rules

This is documented via an online blog:

<http://www.proxiblue.com.au/blog/copy-rules-dynamic-category-products/>

## Some Example rules:

For a more extensive list of example rules, please see our demo store admin:  
Also note that the rule layout may differ slightly in the images below.  
Rule structure is constantly improved, and evolving to make the module work better.

<http://giftpromodemo.proxiblue.com.au/admin>

Username: demo

Password: dem0dem0

## Categories Created from Product Attributes

All Products :



Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Visibility **contains** **Catalog, Search** ❌



All products with the word 'shirt' in the description

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Short Description **contains** **shirt** ❌

Visibility **contains** **Catalog, Search** ❌



All computers

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Attribute Set **contains** **Computer** ❌



All Black Computers

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Attribute Set **contains** **Computer** ❌

Color **contains** **Black** ❌



Products that has a special date set, where TODAY is within the range set

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Special Price Date From/To **is within range on day of viewing** ❌



## All in stock products

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Stock Availability **contains In Stock** ❌



## Products Populated from Sales Reports

### Low Stock

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Any product that will be generated by these Sales Reports: ❌

Products from the low stock report where the stock level is **more than or equals to 1** on the day the category is viewed ❌



Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Any product that will be generated by these Sales Reports: ❌


Products from the low stock report where the stock level is **less than or equals to 100** on the day the category is viewed ❌



## Most Viewed Products

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Any product that will be generated by these Sales Reports: 


Products from the most viewed report **starts 900** day(s) before the day the category is viewed 




## Best Sellers

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Any product that will be generated by these Sales Reports: 


Products from the best sellers sales report **starts 90** day(s) before the day the category is viewed 




## Categories Created from Discounts given

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Any product that matches these discount conditions: 

If a product has a special price value set that gives **more than or equals to 10%** off the product price 

If a product has a Catalog Price Rule applied which gives **more than or equals to 10%** off the product price 



## Data Transformation Rules

Dynamically assign products to this category, if the rules below are met:

If a product is in the catalog with **ALL** of these conditions true:

Color **contains Black, Brown, Green, Magneta, ...** ❌

If a simple product was found **then replace it with Parent Product (Configurable o...** ❌



The system uses the core category system, thus any and all category functionality will be used when dynamic products are displayed (filters, layered navigation etc)

## CRON EXAMPLES

### Add a New Cron Job

Common Settings:	Once Per Five Minutes (*/*5 ****)			REBUILD DELAYED
Minute:	*/5	Once Per Five Minutes (*/*	✓	
Hour:	*	Every Hour (*)	✓	
Day:	*	Every Day (*)	✓	
Month:	*	Every Month (*)	✓	
Weekday:	*	Every Day (*)	✓	
Command:	<input type="text"/>			
	<input type="button" value="Add New Cron Job"/>			

### Add a New Cron Job

Common Settings:	Once Per Day (0 0 ***)			REBUILD ALL CRON
Minute:	0	:00 (At the beginning of th	✓	
Hour:	0	12:00 a.m. Midnight (0)	✓	
Day:	*	Every Day (*)	✓	
Month:	*	Every Month (*)	✓	
Weekday:	*	Every Day (*)	✓	
Command:	<input type="text"/>			
	<input type="button" value="Add New Cron Job"/>			

### Add a New Cron Job

Common Settings:	<input type="text" value="-- Common Settings --"/>		
Minute:	<input type="text" value="*/10"/>	Once Per Ten Minutes ('*/' ▼)	✓
Hour:	<input type="text" value="*"/>	Every Hour (*)	✓
Day:	<input type="text" value="*"/>	Every Day (*)	✓
Month:	<input type="text" value="*"/>	Every Month (*)	✓
Weekday:	<input type="text" value="*"/>	Every Day (*)	✓
Command:	<input type="text"/>		
<input type="button" value="Add New Cron Job"/>			

REBUILD CHANGED

## Upgrade path

### Upgrade from v2 to v3 is a simple upgrade.

Simply unzip the v3 module files into the root of your magento install, replacing all files. The v3 module will automatically update your rules to the new v3 rules layout. You are however encouraged to manually confirm the rules have properly updated, and save each category once after you have confirmed the update of the rules. Unfortunately we cannot foresee all potential configurations of Dynamic Rules used. All our tests with many rule combinations worked 100%.

### Upgrade from pre v2 modules

**Important:** v3 of the module is not backwards compatible to older versions (v0/v1). You must follow the upgrade path to install v3 after v0/v1. Additionally, the dynamic rule definitions are also not compatible, and all dynamic categories rules will need to be re-done after the upgrade.

Before you install v3, as per install instructions above, you must first clear out all previous version files.

You must remove these files:

```
/app/design/adminhtml/default/default/template/dyncatprod
/app/code/community/ProxiBlue/DynCatProd
/app/etc/modules/Proxiblue_DynCatProd.xml
/app/design/adminhtml/default/default/layout/dyncatprod.xml
/skin/adminhtml/default/default/dyncatprod
```

You can do this manually, or you can use the `uninstall_old.sh` script, located in the root of the v3 zip files. Simply copy the script to your magento root folder, and run it using the command:  
`/bin/sh ./uninstall_old.sh`

After you have removed the old module files, you can proceed with a normal v2 install.

**IMPORTANT:** v3 no longer make use of core rewrite directives. If you have had to alter any module code (3rdparty modules) to fix module rewrites, these will need to be reverted, and the original 3rdparty module code restored.

## **Upgrade from pre v4.9.0**

Version 4.9.0 drops the custom grid, in favour of using other 3rd party grid enhancements.

The upgrade removes two files from the extension folder. You will most likely need to manually delete the files, after you have unpacked the zip for the upgrade.

A symptom that this needs to be done is that, after you hit save, or use a column filter, the grid reverts back to the old custom grid.

Simply delete the following files from your magento install:

```
app/code/community/ProxiBlue/DynCatProd/Block/Adminhtml/Catalog/Category/Tab/Product.php  
app/code/community/ProxiBlue/DynCatProd/controllers/Catalog/CategoryController.php
```

That is all.