

Transients

Current Version 1.012

A Transient is a simple way of storing cached data in the database temporarily, by giving it a custom name and a timeframe after which it will expire and be deleted.

Transients are useful when pinging another API (i.e. Wordpress, Instagram, twitter), and saving data for a set period of time.

Once the module is installed, you will find a tab under **Utilities**. This batch list is used for debugging, viewing transients, and deleting transients.

XML Provisioning for deleting transients is supported in v1.011.

Miva JSON API for deleted transients is supported in v1.012.

Please Note

To use this, you must use mvt:do, and having experience with mvt:do is recommended.

Usage

There are two simple functions you can use: - Set_Transient - Get_Transient

Setting a Transient

```
Set_Transient(key, value, expiration)
```

Key: This should be unique. If this key already exists, it will overwrite the value you had before. It is recommended you use this as a code (example: recent_posts, recent_tweets)

Value: The value you are saving/ storing. If you are wanting to save a miva array, serialize this before saving it

Expiration: The amount of seconds to save the transient

Example Syntax of Set_Transient

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc' "  
name="l.set_transient" value="Set_Transient( 'recent_posts',  
l.settings:my_value_here, 60*60*2)" />
```

Getting a Transient

```
Get_Transient(key)
```

This will retrieve your transient. If you have a transient that has not expired, it will return the value. If it has expired, or does not exist it will return null.

Example Syntax of Get_Transient

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:recent_posts" value="Get_Transient( 'recent_posts' )" />
<mvt:if expr="l.settings:recent_posts">
    <mvt:eval expr="l.settings:recent_posts" />
</mvt:if>
```

Example

This is an example of pulling a Blog's most recent posts, and displaying them on SFNT. In this case, the URL we are calling to will display the most recent posts.

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:recent_posts" value="Get_Transient( 'recent_posts' )" />
<mvt:if expr="l.settings:recent_posts">
    <mvt:eval expr="l.settings:recent_posts" />
<mvt:else>
    <mvt:assign name="l.settings:recent_posts" value="" />
    <mvt:call action="'http://www.mydomainname.com/blog/recent-posts/'"
method="'POST'">
        <mvt:assign name="l.settings:recent_posts" value="l.settings:recent_posts $
s.callvalue" />
    </mvt:call>
    <mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.set_transient" value="Set_Transient( 'recent_posts',
l.settings:recent_posts, 60*60*2)" />
    &mvt:recent_posts;
</mvt:if>
```

Other Useful Functions with Examples

In Version 1.006 there are 4 new functions: - [Transient_ReadyTheme_NavigationSet\(readytheme_code, expires \)](#) - [Transient_Load_NavigationSet\(readytheme_code \)](#) - [Transient_ReadyTheme_Image\(readytheme_code, expires \)](#) - [Transient_Load_Link\(item var \)](#) - [Transient_ReadyTheme_ContentSection\(readytheme_code, expires, all_settings var \)](#)

Transient_ReadyTheme_NavigationSet

```
Transient_ReadyTheme_NavigationSet( readytheme_code, expires )
```

readytheme_code: The ReadyTheme Navigation Set Code

expires: Amount of seconds to save the transient

This will first check if the transient exists. If it does, it returns the transient. If it does not, it will set a transient with the key `navigationset__readytheme_code`, and save it. The return value is the rendered HTML data.

Example Syntax of Transient_ReadyTheme_NavigationSet

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:my_navigation_set" value="Transient_ReadyTheme_NavigationSet(
'navigation_bar', 60*60*24)" />
<mvt:if expr="l.settings:my_navigation_set">
    &mvt:my_navigation_set;
<mvt:else>
    <mvt:comment> ==[ Failsafe, incase the Transient Fails ]== </mvt:comment>
    <mvt:item name="readytheme" param="navigationset( 'navigation_bar' )" />
</mvt:if>
```

Using custom items in your Navigationset, like customfields?

```
Transient_ReadyTheme_NavigationSet_WithSettings( readytheme_code, expires,
all_settings var )
```

Example Syntax of Transient_ReadyTheme_NavigationSet_WithSettings

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:navigation_bar_customfields"
value="Transient_ReadyTheme_NavigationSet_WithSettings(
'navigation_bar_customfields', 60*60*24, l.settings)" />
<mvt:if expr="l.settings:navigation_bar_customfields">
    &mvt:navigation_bar_customfields;
<mvt:else>
    <mvt:comment> ==[ Failsafe, incase the Transient Fails ]== </mvt:comment>
    <mvt:item name="readytheme" param="navigationset( 'navigation_bar_customfields'
)" />
</mvt:if>
```

Transient_Load_NavigationSet

```
Transient_Load_NavigationSet( readytheme_code )
```

readytheme_code:The ReadyTheme NavigationSet Code

This returns the ReadyTheme Navigation Set `l.settings:readytheme` variable, including an extra variable in the navigationset, `:link_url` (the navigation item's link). This does **not** cache the data, this function only returns it.

Example Syntax of Transient_Load_NavigationSet

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:load_navigationset" value="Transient_Load_NavigationSet(
'navigation_bar')" />
<mvt:if expr="l.settings:load_navigationset">
  <mvt:comment> ==[ Do something with data..]== </mvt:comment>
  <mvt:eval expr="glosub(miva_array_serialize(l.settings:load_navigationset),
',', '<br />')" />
</mvt:if>
```

Transient_ReadyTheme_Image

```
Transient_ReadyTheme_Image( readytheme_code, expires )
```

readytheme_code: The ReadyTheme Image Code

expires: The amount of seconds to save the transient

This will first check if the transient exists. If it does, it returns the transient. If it does not, it will set a transient with the key `image__readytheme_code`, and save it The return value is the serialized variable of the image.

The variable contains the following:

active, code, cropped_url, id, image_hght, image_id, image_size, image_wdth, link, link_dest, link_targ, link_type, link_url, name, url

Example Syntax of Transient_ReadyTheme_Image

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:banner_1_image" value="Transient_ReadyTheme_Image( 'banner_1',
24*60*60)" />
<mvt:if expr="l.settings:banner_1_image">
  <mvt:assign name="l.settings:banner_1_image" value="miva_array_deserialize(
l.settings:banner_1_image )" />
  <mvt:if expr="l.settings:cropped_url">
    <img data-lazy="&mvt:banner_1_image:cropped_url"
alt="&mvt:banner_1_image:alt;" />
  <mvt:else>
```

```
<img data-lazy="&mvt:banner_1_image:url" alt="&mvt:banner_1_image:alt" />
</mvt:if>
```

Transient_Load_Link

```
Transient_Load_Link( item var )
```

item var: The item variable

This returns the item, as well as `:link` and `:link_url`. This does **not** cache the data, this function only returns it. The variable must have the following in it's structure: - `link_type` (P, C, G, U, N) - `link_dest` (string) - `link_targ` (string) - `name` (string)

Example Syntax of Transient_Load_Link

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:my_var" value="Transient_Load_Link( l.settings:my_var )" />
<mvt:if expr="l.settings:my_var:link_url">
  <a href="&mvt:my_var:link_url">&mvt:my_var:name;</a>
<mvt:else>
  <span>&mvt:my_var:name;</span>
</mvt:if>
```

Transient_ReadyTheme_ContentSection

Please Note: If you have custom logic that is outputting variables (i.e. `g.show_breadcrumbs`), this will not be cached/ output. This is best used when trying to display certain information.

```
Transient_ReadyTheme_ContentSection( readytheme_code, expires, all_settings var )
```

readytheme_code: The ReadyTheme content section code you want to use

expires: Expiration Date

all_settings var: `l.settings` needs to be passed through

Example Syntax of Transient_ReadyTheme_ContentSection

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:about_us_section" value="Transient_ReadyTheme_ContentSection(
'about_us', 24*60*60, l.settings )" />
<mvt:if expr="l.settings:about_us_section">
  &mvt:about_us_section;
<mvt:else>
  <mvt:comment> <!-- Failsafe! --> </mvt:comment>
```

```
<mvt:item name="readytheme" param="contentsection( 'about_us' )" />
</mvt:if>
```

Need a dynamic key for your content section?

When this would be useful: Why not load all your customfield information through a content section, and save the ****rendered*** HTML instead?

```
Transient_ReadyTheme_ContentSection_LowLevel( key, readytheme_code, expires,
all_settings var )
```

Example Syntax of Transient_ReadyTheme_ContentSection_Lowlevel (v1.007)

```
<mvt:do file="g.Module_Root $ '/modules/util/transients.mvc'"
name="l.settings:customlogic" value="Transient_ReadyTheme_ContentSection_Lowlevel(
l.settings:product:code $ '__customlogic', 'customlogic', 24*60*60, l.settings )"
/>
<mvt:if expr="l.settings:customlogic">
    &mvt:customlogic;
<mvt:else>
    <mvt:comment> <!-- Failsafe! --> </mvt:comment>
    <mvt:item name="readytheme" param="contentsection( 'customlogic' )" />
</mvt:if>
```

XML Provisioning

Delete a specific Transient

```
<Module code="tg_transients" feature="util">
    <Delete_Transient key="my_key_here" />
    <Delete_Transient key="my_other_key_here" />
</Module>
```

Delete All Transients

```
<Module code="tg_transients" feature="util">
    <Delete_All_Transients />
</Module>
```

Delete Expired Transients

```
<Module code="tg_transients" feature="util">
    <Delete_Expired_Transients />
</Module>
```

Miva JSON API

Make sure your API User has access to **Store Utility Settings** > Delete, or access will be denied.

Delete a specific Transient

```
{
  "Store_Code": "YOUR_STORE_CODE_HERE",
  "Function": "Module",
  "Module_Code": "tg_transients",
  "Module_Function": "Delete_Transient",
  "Code": "TRANSIENT_CODE_HERE"
}
```

Delete All Transients

```
{
  "Store_Code": "YOUR_STORE_CODE_HERE",
  "Function": "Module",
  "Module_Code": "tg_transients",
  "Module_Function": "Delete_All_Transients"
}
```

Delete Expired Transients

```
{
  "Store_Code": "YOUR_STORE_CODE_HERE",
  "Function": "Module",
  "Module_Code": "tg_transients",
  "Module_Function": "Delete_Expired_Transients"
}
```