

TN3 Gallery

User Guide revision 2.1

www.tn3gallery.com

Installation

Installing the jQuery javascript library

1. Visit <http://jquery.com/> and click the download button. Save the file on your hard drive somewhere you will be able to access it later.

Installing the TN3 jQuery plugin

2. Unzip the TN3 download package you received when you purchased the TN3 Gallery. If you no longer have it you can redownload it from your TN3 Gallery account by logging in at <http://www.tn3gallery.com>
3. Save the TN3 files somewhere on your hard drive you'll be able to access later.

Installing the Default Skins

4. Locate the skins in the TN3 download package. Each skin is provided in a single folder inside the 'skins' folder and contains the user interface images, preloader animated gif, HTML file and a CSS file.
5. Copy the tn3 skins folder into a location on your hard drive you'll be able to access later.

Quick Start

The TN3 Gallery plugin is extremely simple to use, however if you are new to jQuery or working in a browser we recommend that you read the 'Getting Started' section of this userguide in place of this quick start guide.

1. Create a new HTML file.
2. Copy the 'skins' folder into the same directory where your HTML file is.
3. Create an 'images' folder in the same folder where your HTML file is and place some images in it.

In this example we will assume that the large (full size) images are named 1.jpg, 2.jpg, 3.jpg... and that their corresponding thumbnails are named thumb_1.jpg, thumb_2.jpg, thumb_3.jpg...

4. Import the default style sheet:

```
<link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
```

5. Import the latest version of jQuery and the TN3 gallery file:

```
<script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="js/jquery.tn3.min.js"></script>
```

6. Add the following markup to the body of the HTML file, in order to build a gallery containing three images:

```
<div class="tn3gallery">
  
  
  
</div>
```

In order to categorize images into separate albums, or to add titles and descriptions to the images you must use additional markup. Refer to the Getting Started section of this userguide for more details.

Note that to use separate thumbnails you should specify the name of the thumbnail in the img tag and surround it with a tags, specifying the name of the full size image as the href attribute, like so:

```
<a href="images/1.jpg"></a>
```

You can now test the HTML file in a browser to see your TN3 Gallery.

Getting started

Setting up your directories

1. Having downloaded the TN3 files from your account create a new folder on your hard drive where you wish to develop your gallery.
2. Inside the development folder create another folder named 'images' (without the quotes). This directory will hold all of your images.
3. For this userguide we will create 2 directories inside the 'images' directory, one named 'fixed'(to hold normal size images) and one named 'thumbs' (to hold thumbnail images).
4. Back inside the main directory create a folder named 'skins'. This will hold the skins for your gallery.
5. Still inside your main directory create a new folder named 'js' (without the quotes), this will hold your javascript library files.

Your directory structure should now look like this:

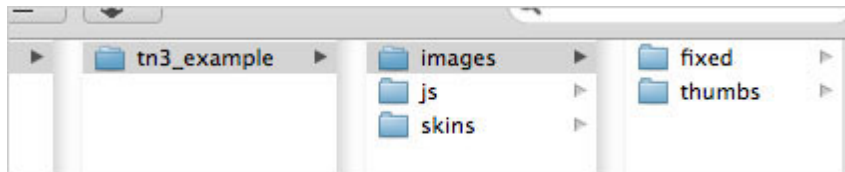


Fig. 1 Directory structure on Mac

Creating your HTML file

6. Open the application you prefer to code markup in. This could be a WYSIWYG editor such as Adobe Dreamweaver or a plain text editor such as Notepad on Windows or TextEdit on Mac.

It's important that you don't use an application that applies formatting to content such as Word as the additional formatting instructions that are saved can interfere with the markup.

If you are going to use TextEdit on a Mac, ensure that the output is set to 'plain text' not 'rich text' in the preferences.

If you're using a WYSIWYG editor such as Dreamweaver start a new HTML file and select 'XHTML 1.0 Strict' as your doctype. Then select the 'Code' view.

If you're not using a WYSIWYG editor, start a new file and type (or copy and paste) the following markup:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>My First TN3 Gallery</title>
  </head>
</body>
</html>
```

The first section of this markup (beginning with <!DOCTYPE... and ending with ...dtd"> should be typed on one line. This is the html file's doctype declaration which is required as part of a valid html document.

Note that unlike most html tags the doctype does not require a forward slash to close it.

If you wish to alter the doctype declaration to use a different doctype, such as xhtml 1.0 transitional you can do so.

The next line is the the opening <html> tag for the document together with the xhtml namespace which is required for the markup to validate.

The next line <head> is the opening section of the header information for the html file. This is where we'll place the TN3 javascript references and styles later on.

The next line which begins with <meta.... and ends ...=UTF-8" /> provides data about your html page, it helps search engines rank your page correctly and like the doctype, it is required if you want your markup to validate.

Most of the html elements we'll use have a separate tag to close the element such as <body> and </body>, the meta tag closes itself with a space followed by a forward slash followed by the closing bracket. Do not omit this.

As with the doctype, if you want to change the charset of your page from UTF-8 to, for example, ISO-8859-1 you can do so.

The next line <title>My First TN3 Gallery</title> is the title of your document. You can change this to anything you like provided the title is surrounded by the <title> and </title>

tags.

Next, we close the head of the document with `</head>`. Then we open the `<body>`, this is the section where we'll enter our gallery data. But for now, close the `</body>` section again.

Finally, close the html document with `</html>`

7. Save the file as 'gallery.html' in the directory you created in step one.

Creating a Gallery

8. To create a gallery in the TN3 markup you add a `<div>` element with the class 'gallery' inside the `<body>` element of your file.

You can add the gallery div anywhere on the page and you can add other content above or below the gallery but for the purposes of this tutorial the only content our file will have will be the gallery.

Your file should now look like this (the new edition is highlighted in bold):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
</head>
<body>
  <div class="gallery">
  </div>
</body>
</html>
```

Adding images

9. To include an image you nest an `` element containing a reference to the thumbnail inside an `<a>` element containing a reference to the large image:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>My First TN3 Gallery</title>
</head>
<body>
    <div class="gallery">
        <a href="images/images/1.jpg">
            
        </a>
    </div>
</body>
</html>

```

Dividing images into Albums

10. The TN3 features the ability to categorize your images into a number of different albums. To add an album to your gallery you first need to add a `<div>` with the class 'tn3 album' inside the gallery div surrounding the images you want to include in it (new code is highlighted in bold):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>My First TN3 Gallery</title>
</head>
<body>
    <div class="gallery">
        <div class="tn3 album">
            <a href="images/images/1.jpg">
                
            </a>
        </div>
    </div>
</body>
</html>

```

11. You can also add a title and description to your album like so:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

```

```

    <title>My First TN3 Gallery</title>
</head>
<body>
    <div class="gallery">
        <div class="tn3 album">
            <h4>My First TN3 Photo Album</h4>
            <div class="tn3 description">1st album I've built.</div>
            <a href="images/images/1.jpg">
                
            </a>
        </div>
    </div>
</body>
</html>

```

Notice that in this description element there's an ' to indicate an apostrophe. You can use character entities as normal when using the TN3 Gallery.

Adding Titles and Descriptions to your Images

12. In order to add data to your images you first need to surround the image with the elements necessary to make it part of an unordered list, like so:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>My First TN3 Gallery</title>
</head>
<body>
    <div class="gallery">
        <div class="tn3 album">
            <h4>My First TN3 Photo Album</h4>
            <div class="tn3 description">1st album I've built.</div>
            <ul>
                <li>
                    <a href="images/images/1.jpg">
                        
                    </a>
                </li>
            </ul>
        </div>
    </div>
</body>

```



```
</html>
```

13. Next, for each list item add an h4 element for the image title, and div elements for additional fields. Here, we will add only one field named 'description', but you can name it whatever you wish and add an unlimited number of other fields:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
</head>
<body>
  <div class="gallery">
    <div class="tn3 album">
      <h4>My First TN3 Photo Album</h4>
      <div class="tn3 description">1st album I've built.</div>
      <ul>
        <li>
          <h4>Image 1</h4>
          <div class="tn3 description">My Image 1 description</div>
          <a href="images/images/1.jpg">
            
          </a>
        </li>
      </ul>
    </div>
  </div>
</body>
</html>
```

14. Finally, repeat the last two steps to add another two images to your gallery. Your HTML file should now look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
</head>
<body>
  <div class="gallery">
    <div class="tn3 album">
      <h4>My First TN3 Photo Album</h4>
      <div class="tn3 description">1st album I've built.</div>
      <ul>
```

```

<li>
  <h4>Image 1</h4>
  <div class="tn3 description">My Image 1 description</div>
  <a href="images/images/1.jpg">
    
  </a>
</li>
<li>
  <h4>Image 2</h4>
  <div class="tn3 description">My Image 2 description</div>
  <a href="images/images/2.jpg">
    
  </a>
</li>
<li>
  <h4>Image 3</h4>
  <div class="tn3 description">My Image 3 description</div>
  <a href="images/images/3.jpg">
    
  </a>
</li>
</ul>
</div>
</div>
</body>
</html>

```

Importing Styles

15. Copy the 'tn3' folder that came with your download (you'll find this inside the 'skins' directory of your distribution archive), into the 'skins' folder that you previously created.
16. Add the following link element to the head of your HTML document to import the styles for the gallery (new code highlighted in bold):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
  <link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
</head>
<body>
  <div class="gallery">

```

```

<div class="tn3 album">
  <h4>My First TN3 Photo Album</h4>
  <div class="tn3 description">1st album I've built.</div>
  <ul>
    <li>
      <h4>Image 1</h4>
      <div class="tn3 description">My Image 1 description</div>
      <a href="images/images/1.jpg">
        
      </a>
    </li>
    <li>
      <h4>Image 2</h4>
      <div class="tn3 description">My Image 2 description</div>
      <a href="images/images/2.jpg">
        
      </a>
    </li>
    <li>
      <h4>Image 3</h4>
      <div class="tn3 description">My Image 3 description</div>
      <a href="images/images/3.jpg">
        
      </a>
    </li>
  </ul>
</div>
</div>
</body>
</html>

```

Importing Scripts

18. The next stage is to import the required javascript into the file to power the TN3 gallery. Locate the jquery.tn3.min.js file that came with your download (inside 'js' folder) and copy it, then paste it into your javascript directory.
19. Next, locate the jQuery file that you downloaded. The name will vary depending on the current version of jQuery but it should look something like jquery-1.5.1.min.js. Copy the file and paste a copy into your 'javascript' directory.
20. Return to your html file. Modify the head section to import the jQuery file, like so (new code is highlighted in bold):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
  <link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
  <script type="text/javascript" src="javascript/jquery-1.5.1.min.js"></
script>
</head>
<body>
  <div class="gallery">
    <div class="tn3 album">
      <h4>My First TN3 Photo Album</h4>
      <div class="tn3 description">1st album I&apos;ve built.</div>
      <ul>
        <li>
          <h4>Image 1</h4>
          <div class="tn3 description">My Image 1 description</div>
          <a href="images/images/1.jpg">
            
          </a>
        </li>
        <li>
          <h4>Image 2</h4>
          <div class="tn3 description">My Image 2 description</div>
          <a href="images/images/2.jpg">
            
          </a>
        </li>
        <li>
          <h4>Image 3</h4>
          <div class="tn3 description">My Image 3 description</div>
          <a href="images/images/3.jpg">
            
          </a>
        </li>
      </ul>
    </div>
  </div>
</body>
</html>

```

It is very important that you ensure that the name of the jquery file you are using (in this example 'jquery-1.5.1.min.js') matches the name of the jquery file you have pasted into your 'js' folder.

21. Next, below the <script> element that imports the jquery file you must add a second pair

of `<script>` tags to import the TN3 file, like so:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
  <link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
  <script type="text/javascript" src="javascript/jquery-1.5.1.min.js"></script>
  <script type="text/javascript" src="javascript/jquery.tn3.min.js"></script>
</head>
<body>
  <div class="gallery">
    <div class="tn3 album">
      <h4>My First TN3 Photo Album</h4>
      <div class="tn3 description">1st album I've built.</div>
      <ul>
        <li>
          <h4>Image 1</h4>
          <div class="tn3 description">My Image 1 description</div>
          <a href="images/images/1.jpg">
            
          </a>
        </li>
        <li>
          <h4>Image 2</h4>
          <div class="tn3 description">My Image 2 description</div>
          <a href="images/images/2.jpg">
            
          </a>
        </li>
        <li>
          <h4>Image 3</h4>
          <div class="tn3 description">My Image 3 description</div>
          <a href="images/images/3.jpg">
            
          </a>
        </li>
      </ul>
    </div>
  </div>
</body>
</html>
```

Initialising the TN3 Gallery

22. The TN3 Gallery will initialize itself by default if the name of the div class is 'tn3gallery'. If however you wish to modify any of the default parameters you'll need to initialize it manually by adding the following code just before the closing </head> tag:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>My First TN3 Gallery</title>
  <link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
  <script type="text/javascript" src="javascript/jquery-1.5.1.min.js"></script>
  <script type="text/javascript" src="javascript/jquery.tn3.min.js"></script>
  <script type="text/javascript">

    $(document).ready(function() {
      $('.gallery').tn3({
        autoplay:true
      });
    });

  </script>
</head>
<body>
  <div class="gallery">
    <div class="tn3 album">
      <h4>My First TN3 Photo Album</h4>
      <div class="tn3 description">1st album I've built.</div>
      <ul>
        <li>
          <h4>Image 1</h4>
          <div class="tn3 description">My Image 1 description</div>
          <a href="images/images/1.jpg">
            
          </a>
        </li>
        <li>
          <h4>Image 2</h4>
          <div class="tn3 description">My Image 2 description</div>
          <a href="images/images/2.jpg">
            
          </a>
        </li>
        <li>
```

```
<h4>Image 3</h4>
<div class="tn3 description">My Image 3 description</div>
<a href="images/images/3.jpg">
  
</a>
</li>
</ul>
</div>
</div>
</body>
</html>
```

This code sets up the behaviour of the gallery. It will cause the slideshow to begin playing automatically.

For more information on possible parameters please refer to the Parameters section of this userguide.

Deploying the Gallery

23. Finally, save the file and open it in a browser to see the gallery working.
24. If you are deploying the gallery on a web server you must ensure that when you upload the files you maintain the directory structure and you upload all files, including the 'js' directory and the 'skins' directory.

In this tutorial we explained how to define the images that tn3 uses in an html file. You can also define gallery content using external sources such as the flickr service, an xml file, all of which is covered later in this userguide.

Using XML

Using XML allows you to define a gallery of images outside of your markup for easy updates. To use XML to define the images in your gallery follow these steps:

1. Create a new HTML file.
2. Copy the default skin folder (tn3) into the same directory.
3. Import the default style sheet:

```
<link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
```

4. Import the latest version of jQuery and the TN3 gallery file:

```
<script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="js/jquery.tn3.min.js"></script>
```

5. Add the following script to the head of your document to populate the gallery with the content of your XML file when the file is ready:

```
<script type="text/javascript">
$(document).ready(function(){
    $('.gallery').tn3({
        external:[{
            origin:"xml",
            url:"gallery.xml"
        }]
    });
});
</script>
```

This code assumes that your XML file, which you'll create shortly will be saved as 'gallery.xml'.

6. Add the following markup to your html to place the gallery in your document:

```
<div class="gallery"></div>
```

7. Create a new XML file.
8. Type or copy and paste the following code into the file:

```
<?xml version="1.0" encoding="utf-8"?>
<tn2>
    <gallery>
        <title>Gallery title</title>
```



```

        <description>Gallery description</description>
        <file_root>images/</file_root>
        <thumb_src>thumbs/1.jpg</thumb_src>
        <images>
            <!-- images will be added here -->
        </images>
    </gallery>
</tn2>

```

The `<title>` and `<description>` are self-explanatory. The `<file_root>` is the root image folder that all images are to be found in and the `<thumb_src>` is the path to the image to use as the album's thumbnail.

9. Inside the `<images></images>` element add the following code for each image you wish to add:

```

<image>
    <title>Image 1</title>
    <description>The 1st image</description>
    <image_src>fixed/1.jpg</image_src>
    <thumb_src>thumbs/1.jpg</thumb_src>
</image>

```

The `<title>` and `<description>` are self-explanatory, the `<image_src>` is the path from the main gallery's `file_root` path to the main image and the `<thumb_src>` is the path from the `file_root` to the thumbnail for the image.

10. Save your images and thumbnails for the images into `images/fixed` and `images/thumbs` directories respectively.

Parameters

In the Initialising the TN3 Gallery section of the Getting Started tutorial you added the autoplay parameter to the gallery, this caused it to auto-play through the slideshow. There are numerous parameters that can be added to configure the look and behaviour of the TN3 Gallery.

To add a parameter to the gallery you add it to the initialisation block of code. So for example, to start the TN3 Gallery from the album view you set the startWithAlbums parameter to true, like so:

```
<script type="text/javascript">
$(document).ready(function() {
  $('.gallery').tn3({
    startWithAlbums:true
  });
});
</script>
```

If you wish to include more than one parameter you must separate the values with a comma, like so:

```
<script type="text/javascript">
$(document).ready(function() {
  $('.gallery').tn3({
    startWithAlbums:true,
    autoplay:true
  });
});
</script>
```

Parameters that are not given specific values revert to their default settings.

General parameters

name	type	default	description
active	array	[]	List of active skin elements. An empty array (the default) means that all elements will be used.
albums_init	function		Function to be triggered when each button in an album list is initialized.
albums_click	function		Function to be triggered when the album button is clicked.

width	number	null	width of the gallery. If value is null, width will be taken from CSS
height	number	null	height of the gallery. If value is null, width will be taken from CSS
albums_init	function		Function to be triggered when a multi-album list is initialized.
autoplay	boolean	false	If true, the slideshow will start automatically.
cssID	string	"tn3"	The prefix for the CSS classes to be used. See TN3 Skinning
data	array	null	Array of album data. See TN3 Data
delay	number	7000	Slideshow delay in milliseconds.
error	function		Function to be triggered when an error occurs.
external	array	[]	An array of external sources (flickr, tn2admin or xml). see list of external parameters.
fullOnly	boolean	false	If set to true, tn3 will be visible only in fullscreen
fullscreen	function		Function to be triggered when the gallery is maximized or minimized over the available screen area.
image	object	{}	See list of imager parameters.
imageClick	string	"next"	"next", "url" or "fullscreen". Action to take when an image is clicked.
iniAlbum	number	0	The initial album index to start the gallery with.
iniImage	number	0	The index of the image to display upon initialization.
init	function		Function to be triggered when the gallery is initialized.
rebuild	function		Function to be triggered when all gallery elements exist.
skin	mixed	null	String or array with 2 values defining the name of the skin HTML file to be loaded. See TN3 Skinning.

skinDefault	string	""	The default skin html, used if the skin value is not defined. See TN3 Skinning
skinDir	string	"skins"	The name of the directory containing skin file. See TN3 Skinning
startWithAlbums	boolean	false	If true, the album list will be displayed upon initialization.
thumbnailer	object	{}	See list of thumbnailer parameters.
timer_end	function		Function to be triggered when the slideshow delay ends.
timer_start	function		Function to be triggered when the slideshow starts.
timer_stop	function		Function to be triggered when the slideshow is paused/stopped.
timer_tick	function		Function to be triggered on every slideshow time interval.
timerMode	string	"bar"	"char" or "bar"
timerSteps	number	500	The number of timer steps.
timerStepChar	string	"•"	Character used for timer step (•)
set_text	function		Function to be triggered when text of each field is set.
init_start	function		Function to be triggered at the start of tn3 initialization, when skin and album data is loaded.
image_data	function		Function to be triggered when external image data is set

Imager Parameters

Imager settings are applied using the image parameter of the main block of code. If for example you wanted to set the images to play in a random order you would use code like this:

```
<script type="text/javascript">
$(document).ready(function() {
```

```

$('.gallery').tn3({
  image:{
    random:true
  }
}); });
</script>

```

name	type	default	description
click	function		The function to be triggered when the image is clicked.
clickEvent	string	"click"	"click" or "dblclick". Name of the jQuery event that triggers a "click" event.
crop	boolean	false	If set to true, image will try to fill image area but it will never scale more then maxZoom value
defaultTransition	object	{transition:"slide"}	The default transition parameters object. See Transitions for more information.
error	function		The function to be triggered when an image file fails to load.
idleDelay	number	3000	If no mouse movement is detected for the period specified, the skin elements that have been set to autohide will hide.
load_end	function		Function to be triggered when an image is loaded.
load_start	function		The function to be triggered when an image starts loading.
maxZoom	number	2	The maximum scale index when images are smaller in size than the image area. 2 = 200%
out	function		Function to be triggered when the mouse pointer moves out of image area.
over	function		Function to be triggered when the mouse pointer is over the image area.
random	boolean	false	If true, the transition order will be random. Only applies if more than one transition is defined.
transition	function		The function to be triggered when the image transition ends.

transitions	array	null	An array of objects holding transition parameters. See Transitions for more information.
-------------	-------	------	--

Thumbnailer Parameters

Thumbnailer settings are applied using the thumbnailer parameter of the main block of code. If you wanted to decrease the inactive area to the edge of the thumbnails you'd do so as follows:

```
<script type="text/javascript">
$(document).ready(function() {
  $('.gallery').tn3({
    thumbnailer:{
      buffer:5
    }
  });
});
</script>
```

name	type	default	description
align	number	1	Sets how the thumbnails should be aligned in the event that they do not fill the full thumbnail area. 0 = left or top, 1 = center or middle, 2 = right or bottom.
buffer	number	20	The size of the 'dead area' at the edges of thumbnailer which doesn't react on mouse over.
click	function		The function to be triggered when thumbnail is clicked.
error	function		The function to be triggered when an image file fails to load.
load	function		The function to be triggered when all thumbnails are loaded.
out	function		The function to be triggered when the mouse moves out of thumbnailer area.
over	function		The function to be triggered when the mouse moves over thumbnailer area.

overMove	boolean	true	If true, the thumbnail imager will scroll on mouse over.
mode	string	"thumbs"	Options are "thumbs", "bullets" or "numbers"
shaderColor	string	"#000000"	The color of the shader.
shaderDuration	number	200	The duration for shader fadeIn.
shaderOpacity	number	0.5	The opacity of the shader.
shaderOut	number	300	Duration of the shader fade out.
slowdown	number	50	The speed of slowdown when the mouse moves out of scroll area.
speed	number	8	The mouse over movement speed.
thumbLoad	function		The function to be triggered when thumbnailer image is loaded.
thumbOut	function		The function to be triggered when mouse moves out of thumbnail image area.
thumbOver	function		The function to be triggered when the mouse moves over thumbnail image area.
useTitle	boolean	false	If true, title of the image will be used as title attribute of img tag

External Parameters

Here is the example of setting external parameters that will load a dozen albums into the gallery, all from different sources:

```
$(document).ready(function() {
    $('.gallery').tn3({
        autoplay:true,
        delay:5000,
        external:[
            {
                origin:"xml",
                url:"images/gallery_18.xml",
                image_size:2
            },
            {
```

```

origin:"flickr",
api_key:"someapikey",
user_id:"someuserid",
// sets, galleries, favorites, interestingness, photostream, search
source: "sets",
// photos specific request parameters - see flickr api
photos: {
    extras:"description",
    page:1,
    per_page:20
},
// album specific request parameters - see flickr api
albums:{},
// object with title, thumb, description properties
albumInfo:{},
thumbSize: "s",
imageSize: "z"
}
]
});
});

```

External: XML

name	type	default	description
image_size	number	0	If greater then 0, the image_src xml parameter will be automatically replaced.
origin	string	"xml"	Set to "xml" when you want to load the gallery definition from XML.
thumb_size	number	1	If the path to the thumbnail image is omitted, the TN3 will create the thumbnail automatically sizing them according to this parameter.
url	string	""	Path to the XML file.

External: tn2admin

name	type	default	description
------	------	---------	-------------

gid	mixed	null	tn2 gallery IDs. If not set, all galleries will be returned. Alternately you can set an array of gallery IDs.
image_path	string	""	Path to root image directory.
image_size	number	0	If greater than 0 the image_src xml parameter will be replaced.
origin	string	"tn2admin"	Set to "tn2admin" when you want to load the gallery definition from the tn2admin.
thumb_size	number	1	If the thumbnail image is omitted, the TN3 will create the thumbnail automatically, sizing it according to this parameter.
url	string	""	Path to XML file.

External: Flickr

Example:

name	type	default	description
albumInfo	object	{}	An object with title, thumb, description properties to be used for album button when source returns only photos.
albums	object	{}	The albums specific request parameters. See flickr API for more details.
api_key	string	""	The flickr API application key. See here for more details.
imageSize	string	"z"	The flickr image size to use for images. See here for more details.
origin	string	"flickr"	Set to "flickr" when you want to load the gallery definition from flickr.
photos	object	{ extras:"description",	The photo specific request parameters. See flickr API for more details.

		<pre> page:1, per_page:20, media:"photos" } </pre>	
source	string	"search"	Possible values are: "sets", "galleries", "favorites", "interestingness", "photostream" or "search". Flickr API method in use. See flickr API for more details.
thumbSize	string	"s"	The flickr image size to use for thumbnails. See here for more details.
user_id	string	""	The NSID of the user to get a photoset list for. If none is specified, the calling user is assumed.

Working with Transitions

Transitions control the animation of the images from one to another. Transitions are set on the gallery using the image parameter, so for example, to set the TN3 Gallery to use a slide transition you would modify the initialisation code as follows:

```
<script type="text/javascript">
$(document).ready(function() {
  $('.gallery').tn3({
    image:{
      transitions:[{
        type:"blinds",
        duration:150,
        direction:"horizontal",
        easing:"sineOut",
        parts:16,
        partDuration:300,
        partEasing:"quadIn",
        method:"fade"
      }]
    }
  });
});
</script>
```

Setting Multiple Transitions

The TN3 will accept multiple transitions, which can be set up one after the other in the transitions parameter. Multiple transitions will play through in order unless the optional 'random' parameter is set to true:

```
<script type="text/javascript">
$(document).ready(function() {
  $('.gallery').tn3({
    image:{
      random:true,
      transitions:[
        {
          type:"fade",
          duration:250
        },
        {
          type:"slide",
          duration:200
        }
      ]
    }
  });
});
</script>
```

Setting Default Transition

Not all browsers will support more advanced transitions, some features of Internet Explorer 8 for example preclude the use of some Javascript. Additionally some transition engines (such as 'blinds' and 'grid') require both images to have the same dimensions and not to be scaled. It is therefore recommended that you supplement your transition settings with a simple default transition that will play if the more complex transition cannot.

You set a default transition at the same type as you set the main transition, in the image parameter like so:

```
<script type="text/javascript">
$(document).ready(function() {
    $('.gallery').tn3({
        image:{
            transitions:[{
                type:"blinds",
                duration:150,
                direction:"horizontal",
                easing:"sineOut",
                parts:16,
                partDuration:300,
                partEasing:"quadIn",
                method:"fade"
            }],
            defaultTransition:{
                type:"fade",
                duration:250,
                easing:"swing"
            }
        }
    });
});
</script>
```

Transition Parameters

Each transition has one optional parameter “type” with which you specify the transition type and multiple other parameters which you can specify or leave at the default as you prefer. The parameters are as follows:

Fade Transition

name	type	default	description
duration	number	600	duration of transition in milliseconds

easing	string	"swing"	easing function used for opacity change
--------	--------	---------	---

Slide Transition

name	type	default	description
duration	number	300	duration of transition in milliseconds
direction	string	"auto"	"left", "right", "top", "bottom" or "auto". Direction of sliding. "auto" setting will use "left" or "right" depending on the sorting position of previous image
easing	string	"circularOut"	easing function used for previous image movement

Blinds Transition

name	type	default	description
duration	number	900	duration of transition in milliseconds
direction	string	"horizontal"	"horizontal" or "vertical". This parameter defines if horizontal or vertical blinds will be used
easing	string	"sineOut"	easing function used to define timing for which part transitions will execute
parts	number	16	number of blinds
partDuration	number	300	duration of each part transition
partEasing	string	"quadraticIn"	easing function used for each part transition
method	string	"fade"	"fade", "scale" or "slide". type of part transition
partDirection	string	"auto"	used with "slide" method only. "left", "right", "top", "bottom" or "auto". Direction of part sliding.
cross	boolean	true	used with "slide" method only. When set to "true" the slide direction will change depending on whether the part is odd or even.

Grid Transition

name	type	default	description
duration	number	900	duration of transition in milliseconds
easing	string	"sineOut"	easing function used to define timing for which part transitions will execute
gridX	number	7	number of horizontal grid parts
gridY	number	5	number of vertical grid parts
sort	string	"diagonal"	"flat", "diagonal", "circle" or "random". Sorting algorithm for grid parts
sortReverse	boolean	false	When set to "true" the order of sorted parts is reversed.
diagonalStart	string	"bl"	"tr", "tl", "br" or "bl". Defines starting part when diagonal sorting is used (top-right, top-left, bottom-right, bottom-left)
method	string	"fade"	"fade" or "scale", the type of part transition
partDuration	number	400	duration of each part transition
partEasing	string	"sineIn"	easing function used for each part transition
partDirection	string	"auto"	used with "scale" method only. "left" or "top". Direction of part scaling.

Using External Links

The TN3 Gallery can be controlled by a simple html links that you can place anywhere in your document.

Moving to the next & previous image

You can move to the next image in a gallery using this code:

```
<a href="#tn3-0-next">Next Image</a>
```

You can also move to the previous image in a gallery using this code:

```
<a href="#tn3-0-prev">Previous Image</a>
```

Javascript indexes begin at zero so the first gallery on the page is '0', the second is '1' and so forth.

Moving to a specific image

You can specify a specific image to move to using code like this:

```
<a href="#tn3-0-0-0">First image</a>
```

The numbers in the link represent the index of the gallery in the document, the index of the album within the gallery and the index of the image in the album, so to open the fifth image of the 2nd album in the 3rd TN3 instance, you'd use this link:

```
<a href="#tn3-2-1-4">Image link</a>
```

Opening in fullscreen mode

To open an image in fullscreen mode you append '-fs' to the link, for example:

```
<a href="#tn3-0-next-fs">Next Image (fullscreen)</a>
```

or

```
<a href="#tn3-0-0-0-fs">First image (fullscreen)</a>
```


TN3 Application Programming Interface

To understand how the TN3 API functions you first need to have basic understanding of jQuery. If you don't have prior knowledge of jQuery please take a few minutes to familiarise yourself with the contents of [this tutorial](#) before using the TN3 API.

Static Methods

jQuery.fn.tn3.translate(string, translation)

Changes the string used by the TN3 Gallery.

Example:

```
$(document).ready(function() {  
    $.fn.tn3.translate("Maximize", "FullScreen");  
});
```

Static Properties

jQuery.fn.tn3.config

- .data*
- .skin*
- .skinDir*
- .skinDefault*
- .cssID*

jQuery.fn.tn3.Gallery.config

- .active*
- .autoplay*
- .delay*
- .fullOnly*
- .width*
- .height*
- .iniAlbum*
- .iniImage*
- .imageClick*
- .startWithAlbums*
- .timerMode*
- .timerSteps*
- .timerStepChar*

jQuery.fn.tn3.Imager.config

- .defaultTransition*
- .clickEvent*

```
.crop  
.idleDelay  
.maxZoom  
.random  
jQuery.fn.tn3.Thumbnailer.config  
.align  
.buffer  
.mode  
.overMove  
.shaderColor  
.shaderDuration  
.shaderOpacity  
.shaderOut  
.slowdown  
.speed  
.seqLoad  
.useTitle
```

These config objects store default values for TN3 parameters. Refer to the TN3 Parameters for a detailed description of each property.

The following example will set the slideshow to always start automatically, even if the autoplay property is not set in the plug-in initialization object:

```
$(document).ready(function() {  
    $.fn.tn3.Gallery.config.autoplay = true;  
});
```

API Scope

All methods and properties (except static ones) in this reference refer to the instance of the TN3 gallery which is returned as the 'tn3' property of the jQuery 'data' object which itself is returned at tn3 plug-in initialization. In the following example, the javascript variable '\$mytn3' refers to an instance of the TN3 gallery:

```
$(document).ready(function() {  
    var $mytn3 = $('.gallery').tn3({  
        autoplay: true  
    }).data('tn3');  
    // call fullscreen method on instance of tn3 gallery  
    $mytn3.fullscreen(); });
```

Note: If the TN3 is set to load an external skin, you should only use the API after the 'init' event has been triggered.

Methods

.fullscreen()

Toggles between 'normal' and fullscreen mode.

Example:

```
$mytn3.fullscreen();
```

.resize(width, height)

Changes the width and height of the gallery.

Example:

```
$mytn3.resize(400, 300);
```

.show(n)

Shows the 'n' image in the gallery. The value of the parameter 'n' can be any integer (javascript indexes begin at 0 so use 0 for the first image, 1 for the second image and so forth). Alternately you can use the strings "next" or "prev" to move to the adjacent images in the slideshow.

The following example will display next image in the gallery:

```
$mytn3.show("next");
```

.showAlbum(albumID, imageID)

Displays and makes the album in a multi-album gallery active. The value of the parameters 'albumID' and 'imageID' can be any integer (0 for the first, 1 for second etc).

The following example will display the second image of the third album in the gallery:

```
$mytn3.showAlbum(2, 1);
```

.setAlbumData(data)

Sets data for the gallery albums. See TN3 Data for more information on data structure.

Example:

```
$mytn3.setAlbumData(dataArray);
```

.setImageData(data, albumID)

Sets data for images in the album. See TN3 Data for more information on data structure.

Example:

```
$mytn3.setImageData(dataArray, albumID);
```

Properties

.data - gallery data array

.cAlbum - index of currently active album

.n - index of currently active image

`.thumbnailer` - instance of thumbnailer

`.imager` - instance of imager

The following example will alert index of the currently active album:

```
alert($mytn3.cAlbum);
```

Skinning

The TN3 Gallery looks for its default skins in the 'skins' folder, but you can override this default behaviour if you wish by using the 'skinDir' parameter of the TN3 gallery. Name of the skin file to be loaded is defined by 'skin' parameter. Refer to the “TN3 Parameters” section of this userguide for more information. Examples:

skinDir = “mySkinDir”

skin = “mySkin”

Skin file that will be loaded has path: “/mySkinDir/mySkin/mySkin.html”

skin = “mySkin”

Skin file that will be loaded has path: “/skins/mySkin/mySkin.html”

skin = [“mySkin”, “variant”]

Skin file that will be loaded has path: “/skins/mySkin/variant.html”

Note: *TN3 Gallery will only load the HTML file of the skin and not the CSS file which is specified inside the skin's HTML file. You will need to link to the skin's CSS file manually in the html page in which the TN3 Gallery is displayed.*

Basic Skinning

For most purposes you can modify the existing skin supplied with the component to create your desired effect. To edit the default skin follow these steps:

1. Open the tn3.html (default skin) file that you'll locate inside the 'skins/tn3' folder. Check the stack order of the html elements. This is the structure of the markup and styles, particularly positioning styles depend upon it.

In most circumstances you can leave the stack order as it is. If necessary Modify the order of the html tags to alter the stack order of the HTML elements that make up the TN3 Gallery.

2. Open the tn3.css file that you'll locate inside the 'skins/tn3' folder. This dictates the styles applied to the HTML elements and any change you make here will be reflected in the Gallery's display.
3. Open the HTML file containing the TN3 Gallery with the default skin to see the gallery

with the changes that you have made.

Note: The default `tn3.html` file will **not** be loaded if you don't set the 'skin' parameter, as it is included in the `TN3Gallery` plugin. If you made changes to `tn3.html`, be sure to specify the 'skin' parameter as 'tn3' when initializing the `TN3 Gallery`.

Identifying Skin Elements

Each element in the gallery is represented by a style in the skin's CSS file. To skin the `TN3 Gallery` it's essential to understand what style definitions apply where.

[We will add interactive javascript widget here]

Creating Skins

Advanced users may find that in order to achieve their desired look, they prefer to create a custom skin rather than modify the default skin.

To create a new skin instead of modifying the default one, follow these steps:

1. Locate the default skins folder that came with your download, it's named 'tn3' and is located inside the 'skins' folder. Duplicate the folder and rename the copy, giving it whatever name you wish, for the purpose of this tutorial name it 'tn3-custom'.

As the name of the skin, 'tn3-custom' will be used in several places in this tutorial. If you use a different name, ensure that you are consistent throughout your files.

2. Inside the `tn3-custom` folder, locate the 'tn3.css' file, rename it 'tn3-custom.css'.
3. Still inside the `tn3-custom` folder, locate the 'tn3.html' file, rename it 'tn3-custom.html'.
4. Open the 'tn3-custom.css' file. Replace every class prefix '.tn3-' with '.tn3-custom-'. So for example the selector for the gallery should be changed from

```
.tn3-gallery {
```

to

```
.tn3-custom-gallery {
```

Once you've changed every selector save the file.

5. Open the 'tn3-custom.html' file. Change the link to reflect your skin name, like so:

```
<link rel="stylesheet" type="text/css" href="tn3-custom.css" />
```

6. With the 'tn3-custom.html' file still open modify all of the class names to reflect the name of your skin, So for example, the gallery class should be changed from

```
<div class="tn3-gallery">
```

to

```
<div class="tn3-custom-gallery">
```

Once you've made this change, save the file.

7. At this stage it is advisable to test the newly created skin in a browser to ensure that everything is working correctly. At this point, the TN3 Gallery should still look like the default skin. Here is a simple html markup example that uses our new skin:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>jQuery tn3gallery Plugin - Example</title>
```

```
<!-- include CSS and JavaScript files ->
```

```
<link type="text/css" rel="stylesheet" href="skins/tn3-custom/tn3-custom.css"></link>
```

```
<script type="text/javascript" src="../lib/jquery-1.5.1.js"></script>
```

```
<script type="text/javascript" src="../lib/jquery.tn3.min.js"></script>
```

```
<!-- initialize tn3gallery when DOM is ready. ->
```

```
<script type="text/javascript">
```

```
$(document).ready(function() {
```

```
    $('.gallery').tn3({
        skin: "tn3-custom",
        autoplay: true,
        delay: 5000,
        image: {
            transitions:[{
                type:"slide",
                duration:600,
                direction:"top"
            }]
        }
    });
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```

    <!-- define content →
    <div class="gallery">
        <div class="tn3 album">
            <h4>My Photo Album 1</h4>
            <div class="tn3 description">This is the first album in the
allery.</div>
            <ul>
                <li>
                    <h4>Photo 1</h4>
                    <a href="images/1.jpg">
                        
                    </a>
                </li>
                <li>
                    <h4>2nd Photo</h4>
                    <div class="tn3 description">This image has description</
div>
                    <a href="images/2.jpg">
                        
                    </a>
                </li>
                <li>
                    <h4>Photo No.3</h4>
                    <div class="tn3 description">This image has both
escription...</div>
                    <div class="tn3 somefield">...and another field defined</
div>
                    <a href="images/3.jpg">
                        
                    </a>
                </li>
            </ul>
        </div>
    </div>
</body>
</html>

```

8. Edit the newly created HTML and CSS files to style your new skin.

Special skin classes

The skin elements can have special classes. These are used for properties that cannot be defined through CSS by attaching the class to skin element. For example, to center the preloader both horizontally and vertically, we will write:


```
<div class="tn3-preloader tn3_h tn3_v"></div>
```

All special classes are prefixed with 'tn3_'. Here is a list of all of the available special classes:

tn3_h - element will be horizontally centered whenever the gallery is resized

tn3_v - element will be vertically centered whenever the gallery is resized

tn3_o - element will be shown only when mouse cursor is over the image area

tn3_rh - element will be resized horizontally

tn3_rv - element will be resized vertically

TN3 History Plugin

TN3 History plugin allows you to maintain a history of a user's gallery navigation and creates permalinks for every tn3 state.

To enable the plugin, include it in the page after the tn3 gallery:

```
<link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
<script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="js/jquery.tn3.min.js"></script>
<script type="text/javascript" src="js/tn3.history.min.js"></script>
```

History plugin parameters:

name	type	default	description
slugField	string	"title"	Field from which a slug (url friendly image name) for the permalink will be retrieved. TN3 fields can be defined in html, xml or by any external source. If this field doesn't have a value, the plugin will use the order number for the slug.
from	string	"./_,:;"	A string of characters to be replaced when creating the slug.
to	string	"-----"	A string of characters to use when replacing characters defined by the 'from' setting.
key	string	"tn3"	The hash key to be used for the permalink
delimiter	string	"/"	Character(s) to delimit the album and image slugs in the permalink.

If, for example, an album with the title "My Album" contained an image with title "My Beautiful Image", the permalink to the image — based on default values — would be:

```
http://www.example.com/gallery.html#tn3=my-album/my-beautiful-image
```

The History plugin will create the slug from the album and image title by using url friendly characters. If your slugField uses other unicode characters, that need to be replaced, use

the "from" and "to" plugin parameters to define how this should be done.

Creating title independant slugs

You can also create slugs independent of the title attributes of the album and image. To do so follow these steps:

1. Set the history plugin to use the field named 'slug' when creating slugs:

```
history: {  
  slugField: "slug" }
```

2. When defining the TN3 Gallery content (either by HTML or XML), also define values for slug fields. For example, in HTML define an image slug by adding the following line to a 'li' tag:

```
<div class="tn3 slug">my-own-image-slug</div>
```

For more information see the "Getting Started - Adding Titles and Descriptions" section of the userguide.

TN3 History plugin uses [BBQ: Back Button & Query Library](#) jQuery plugin written by Ben Alman.

TN3 Flickr Plugin

The TN3 Flickr plugin allows you to display images from Yahoo's Flickr service in your TN3 Gallery.

To enable the plugin, include it in the page after the TN3 Gallery:

```
<link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
<script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="js/jquery.tn3.min.js"></script>
<script type="text/javascript" src="js/tn3.flickr.min.js"></script>
```

Flickr plugin parameters:

name	type	default	description
origin	string	"flickr"	Set to "flickr" when you want to load the gallery definition from flickr.
api_key	string	""	The flickr API application key. See here for more details.
user_id	string	""	The NSID of the user to retrieve a photoset list for. If none is supplied, the calling user is assumed.
source	string	"search"	The Flickr API method in use. Possible values are: <ul style="list-style-type: none">- "sets"- "galleries"- "favorites"- "interestingness"- "photostream"- "search" See flickr API for more details.
photos	object	{ extras:"description", page:1, per_page:20, media:"photos" }	The photo specific request parameters. See flickr API for more details.
albums	object	{}	The albums specific request parameters. See flickr API for more details.

albumInfo	object	{}	An object with title, thumb and description properties to be used for the album button when the source returns only photos.
imageSize	string	"z"	The flickr image size to use for images. See here for more details.
thumbSize	string	"s"	The flickr image size to use for thumbnails. See here for more details.

When the source is set to "sets" or "galleries", the TN3 will first request the retrieval of the user's albums (sets or galleries). All other possible source values will result in the retrieval of one album only. You can filter the list of albums to display using the TN3's 'init_start' event.

Examples:

- Retrieve all sets for the user:

```
external: [{
  origin: "flickr",
  api_key: "your_flickr_api_key",
  user_id: "some_user_id",
  source: "sets" }]
```

- Retrieve the first 30 [Flickr interestingness images](#):

```
external: [{
  origin: "flickr",
  api_key: "your_flickr_api_key",
  source: "interestingness",
  photos: {
    per-page: 30
  } }]
```

- Retrieve all images tagged with "beach":

```
external: [{
  origin: "flickr",
  api_key: "your_flickr_api_key",
  photos: {
    tags: "beach"
  } }]
```

- Retrieve all images associated with the phrase "niagara falls":

```
external: [{  
  origin: "flickr",  
  api_key: "your_flickr_api_key",  
  photos: {  
    text: "niagara falls"  
  } }]  
}
```

- Retrieve user images by location (use [flickr.places.find](#) to identify the place_id):

```
external: [{  
  origin: "flickr",  
  api_key: "your_flickr_api_key",  
  user_id: "some_user_id",  
  photos: {  
    place_id: ".skCPTpTVr.Q3WKW"  
  } }]
```

- Link to Flickr web URL in the description field:

All data returned from the Flickr API is saved in the TN3 data 'raw' field. We can use the ID of the photo and construct a URL to the Flickr web page. By using the 'set_text' event, we can then add that URL to the description field:

```
set_text: function(e) {  
  if (e.field == "description") {  
    var furl = "http://www.flickr.com/photos/";  
    furl += e.data.raw.owner + "/" + e.data.raw.id;  
    e.text += ' <a href="' + furl + '" target="_blank">See on flickr</a>';  
  } }  
}
```

TN3 Picasa Plugin

The TN3 Picasa plugin allows you to display images from Google's Picasaweb service.

To enable the plugin, include it in the page after the TN3 gallery:

```
<link type="text/css" rel="stylesheet" href="skins/tn3/tn3.css"></link>
<script type="text/javascript" src="js/jquery-1.5.1.min.js"></script>
<script type="text/javascript" src="js/jquery.tn3.min.js"></script>
<script type="text/javascript" src="js/tn3.picasa.min.js"></script>
```

Picasa plugin parameters:

name	type	default	description
origin	string	"picasa"	Set to "picasa" when you want to load the gallery definition from Picasaweb.
userID	string	"default"	Your Google user id
source	string	"albums"	Possible values are: <ul style="list-style-type: none">- albums (all user albums)- photos (all user photos)- album (one album of user)- all (all picasaweb photos)- featured
albumID	string	""	When the source is set to 'album', this value will be used as the ID of the album.
params	object	{ "start-index": 1, "max-results": 20, thumbsize: "72c", imgmax: "1024" }	Extra parameters to be sent with each photo request. See Picasaweb API for more details.
aparams	object	{ thumbsize: "72c" }	Extra parameters to be sent with each album request. See Picasaweb API for more details.
albumInfo	object	{ }	An object with title, thumb, description properties to be used for the album button when the source returns images only.

When the source is set to “albums”, the TN3 Gallery will initially request all of the user’s albums. All other possible source values will result in the retrieval of a single album only. You can filter the list of albums to display using the TN3’s ‘init_start’ event.

Examples:

- Retrieve all of a user’s albums:

```
external: [{
  origin: "picasa",
  userID: "some_user_id" }]
```

- Retrieve the first 30 Picasa Web Featured images <https://picasaweb.google.com/lh/explore>:

```
external: [{
  origin: "picasa",
  source: "featured",
  params: {
    "max-results": 30
  } }]
```

- Retrieve images tagged with "beach":

```
external: [{
  origin: "picasa",
  source: "all",
  params: {
    tag: "beach"
  } }]
```

- Retrieve images associated with the phrase "niagara falls":

```
external: [{
  origin: "picasa",
  source: "all",
  params: {
    q: "niagara falls"
  } }]
```

- Retrieve user images by location:

```
external: [{
  origin: "picasa",
  userID: "some_user_id",
  source: "photos",
```

```
params: {  
  1: "New York"  
} }]
```