TPS Landing Constructor Developers Manual

This manual is intended for Web developers who create layouts to be imported into the Landing Constructor plugin. The manual describes the rules for layout package creation.

CONTENTS

LANDING CONSTRUCTOR OVERVIEW

The Landing Constructor plugin facilitates landing pages creation and localization. Landing pages are created for promotion and are intended to introduce the game to users and attract new players to the registration page.

Landing Page Example:

The pages with the mirror-like content can exist for different languages. Such pages are localized with the help of the Landing Constructor plugin.

To facilitate landing pages creation, the developers create landing page layouts that are used for localizations in Landing Constructor. The layout is structured as several information blocks called placeholders. Content values are added for each placeholder, and then localized into different languages. The number of placeholders and the landing page layout depends on design and is predefined in the layout package.

Prior to starting to work with Landing Constructor, the layout package is created according to the planned design and rules. After the package is uploaded into Landing Constructor, the content for each placeholder can be added or modified to create a landing page in the required languages.

For additional information on main Landing Constructor definitions, see Appendix 2. Glossary.

LAYOUT PACKAGE

To assemble the correct landing page, the layout package must be created according to particular rules for package structure.

The layout package is a .zip archive that includes one landing layout. The complete package consists of the following parts:

- Configuration file (XML) – required
- Template file (HTML) – required
- Particular layout files (CSS, JS, fonts) – optional
- Variative content (gallery images, background images, text, etc.) – optional

Landing Constructor requires the **configuration** and **template files** to be included in the package. Other content is optional and is added to the package later using Landing Constructor.

> The names of the package, configuration and template files must be the same.

The file tree example on the right shows a package directories structure.

The layout package file system includes the following directories and files:

Directory	Description
`layout_name/`	Root directory of the package
`layout_name/js|css|media|`	Template directory for JavaScript scripts, styles, static images, or other content
`layout_name/layout_name.html`	Master template file containing the layout and tags for insertion of variative content
`layout_name/layout_name.xml`	Template configuration file that describes the layout parameters and template placeholders
`layout_name/variative_content/`	Localization directory
`layout_name/variative_content/RU|EN| etc.`	Directory containing another directory appropriate for the particular placeholder

Directory	Description
`layout_name/variative_content/ru/placeholder_name1/`	Placeholder directory for directories to the placeholder elements
`layout_name/variative_content/ru/placeholder_name1/elements/element1`	Placeholder element directory for the content directories numbered in ascending order
`layout_name/variative_content/ru/placeholder_name1/elements/element1/1`	Content directory for the particular file (image, text, etc.)

The content is placed in the directory according to the [placeholder localization type](#):

Condition	Destination
Needs Translation	Localization directory of the particular language
For All Localizations	Localization directory of the master language
Managed Manually	Localization directory of the manually selected language

The directory structure corresponds to the placeholder description in the configuration file:
`variative_content/localization_name/placeholder_name/`

CONFIGURATION FILE

Configuration file is an .xml file with a specific markup. The file includes all placeholders to be used in Landing Constructor.

The names of the package, configuration and template files must be the same.

The example shows the configuration file structure detailing all main file sections with tags and placeholder rules. **See example**

Markup Tags

The configuration file structure includes several tags that comprise the main sections of the file. The following tags are required:

Tag	Description
layout	Root element in the configuration file. Defines content blocks location on a web page.
localizations	List of initially available localizations
localization	Single localization tag
localization name	Localization name to be displayed in the LC administration panel
localization master	Tag defining the master localization
placeholders	List of all placeholders in the layout
placeholder	Tag bounding a single placeholder
placeholder label	Unique placeholder qualifier
placeholder type	Data type of a placeholder
placeholder relation_type	Placeholder localization type
admintitle	Placeholder or element name to be displayed in the LC administration panel
elements	Tag bounding placeholder elements
element*	Placeholder element. Can have any name. The number of placeholder attributes depends on the element content type.
element type	Content type of a placeholder element

Placeholder Types

Layout consists of several blocks called placeholders. Placeholders describe elements and element combinations reserving fields for content input (images, quote list, video gallery, etc.). Each placeholder creates a separate layout section in Landing Constructor for adding content variants by the localization team.

Landing Constructor allows creating pages with a number of set layout blocks that remain for any localization of the particular layout.

Example: If the gallery contains 10 images, every language layout has 10 images.

Content Types

Content type of a placeholder defines the content to be included into the placeholder via Landing Constructor.

> To optimize Landing Constructor, specify the data type only for image, text, or list placeholders. Omit the type for any other compound placeholder.

The configuration file includes placeholders of several data types:
- **Simple**, containing one element.
  - **text**

```
Example:
<placeholder label="description" type="text">
        <admintitle><![CDATA[Page meta-description
settings]]></admintitle>
        <elements>
            <text type="text" max_length="200">
        <admintitle><![CDATA[Description]]></admintitle>
            </text>
        </elements>
    </placeholder>
```

  - **image**

```
Example:
<placeholder label="background" type="image">
        <admintitle><![CDATA[Page background settings]]></admintitle>
        <elements>
            <image type="image" width="3000" height="2000" size="500"
ext="jpg,png">
        <admintitle><![CDATA[Background]]></admintitle>
            </image>
        </elements>
    </placeholder>
```

- **Compound**, containing two or more connected elements.
  - **list**. Includes two related texts of variable length (quotation text and author name).

```
Example:
<placeholder label="quote-list" type="list">
        <admintitle><![CDATA[Quote list settings]]></admintitle>
            <elements>
                <text type="text-list" max_length="500">
                    <admintitle><![CDATA[Quote text]]></admintitle>
                </text>
                <author type="text-list" max_length="500">
                    <admintitle><![CDATA[Quote author]]></admintitle>
                </author>

            </elements>
    </placeholder>
```

  - All other compound placeholders excluding `list`

```
Example:
<placeholder label="slogan">
      <admintitle><![CDATA[Slogan setup]]></admintitle>
      <elements>
        <image type="image" width="500" height="500" size="500"
ext="jpg,png">
        <admintitle><![CDATA[Slogan]]></admintitle>
        </image>
        <alt type="text" max_length="150">
        <admintitle><![CDATA[Slogan alternative text]]></admintitle>
        </alt>
      </elements>
    </placeholder>
```

Each placeholder and leaf element must include the description to display in the Landing Constructor Administration panel. The description is added using the following tag structure:

```
<admintitle><![CDATA[description_here]]></admintitle>
```

```
Example:
<placeholder label="screenshots" type="text">
        <admintitle><![CDATA[Gallery title settings]]></admintitle>
        <elements>
            <text type="text" max_length="150">
                <admintitle><![CDATA[Title]]></admintitle>
            </text>
        </elements>
      </placeholder>
```

Localization Types

Starting from Landing Constructor version 2.1, the configuration file specifies a localization type for every placeholder. Localization type defines how the placeholder content is localized in Landing Constructor:

Type	Attribute	Description
Needs Translation	`one-to-one`	Localization is added via translation from the Master language. Slave localization variants are related to Master localization variants.
For All Localizations	`one-to-all`	Variative content that is common for all localizations
Managed Manually	`unrelated`	Localization variants that are not related to other variants. The content is not sent for localization, but can be added and modified manually.

Placeholder Elements

The content in a placeholder is structured by elements. Each element is intended for content of a particular type. The placeholder can have one or more elements. A placeholder includes one or several elements of the following types:

- `text`
- `image`
- `text-list`

An element can have attributes specifying content dimensions. These attributes must contain predefined measurements complying with landing design.

Element	Attribute	Description
text	`max_length`	Placeholder for a text of a specified maximum length
image	`width`	The dimensions are strict and depend on the design of a layout section where the image is located: Image width and height (in pixels)
	`height`	
	`size`	The maximum image size (in Kb)
	`ext`	Possible extensions (separated by commas)
text-list	`max_length`	Placeholder for a list item text of a specified maximum length

TEMPLATE

Template is a required part of the configuration file. The template file has the `.html` extension.

> The names of the package, configuration and template files must be the same.

Template controls the outlook and markup of a landing page and provides the framework to combine all landing page blocks.

The template file example creates the following layout of the landing page:

Placeholders

The template describes all placeholders stated in the configuration file. To include variative content, use the tags `{{ }}` corresponding to the placeholders in the configuration file.

```
Example:
<li class="b-gallery_item">
        <a rel="example_group" href="{{screenshot_gallery->image4->big_img}}"
title="" class="b-gallery_link js-gallery-link">
        <img class="b-gallery_img" src="{{screenshot_gallery->image4-
>small_img}}" alt="World of Tanks">
        </a>
</li>
```

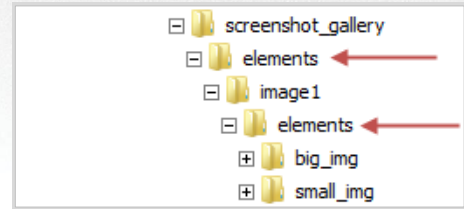The **list** placeholder type has the following labels:
```
{{#placeholder_label}}
{{/placeholder_label}}
```

To insert variable content into a template, use the unique placeholder tag.

The element path consists of all placeholder folders excluding the Elements folder.

Example:
The image path is `{{screenshot_gallery->image1->big_img}}`

Resources Paths

When Landing Constructor generates a page, the system adds the base tag to the relative resource (images, files, etc.) URL. This influences all template anchors.

Example:
If the base becomes `<base href="/wp-content/plugins/landing-constructor/layouts/2013_11_06__11_49_34_WoTX PC LP/" target="_blank">`,

then the anchor redirects to the link `href="/wp-content/plugins/landing-constructor/layouts/2013_11_06__11_49_34_WoTX PC LP/#anchor`

If resources have relative URL, <u>consider the base tag</u> in template development:

1. Acquire the current base for the dynamic file loading (for JavaScript code).
   Example:
   `document.getElementsByTagName('base')[0].href`

2. Ensure the relative path correctly opens the requested anchor. One solution is including the following script in the template:
   ```
   <script>
   $(document).ready(function(){
   $("a[href^=#]").each(function(){ var href = $(this).attr("href");
   $(this).attr("href", window.location + href); });
   });
   </script>
   ```

Example:

```html
<!DOCTYPE html>
<html>
<head>
    <title>{{page_title->element1}}</title>
    <link rel="shortcut icon" href="{{favicon->element1}}">
     <script>
       $(document).ready(function(){
        $("a[href^=#]").each(function(){ var href =
$(this).attr("href"); $(this).attr("href", window.location + href); });
       });
     </script>
 </head>
<body style="background-image: {{background->element1}}">
    <ul id="mycarousel">
        <li><a href="{{image_gallery->element1->element1}}"
class="media-screen"><img src="{{image_gallery->element1-
>element2}}"></a></li>
        <li><a href="{{image_gallery->element2->element1}}"
class="media-screen"><img src="{{image_gallery->element1-
>element2}}"></a></li>
        <li><a href="{{image_gallery->element3->element1}}"
class="media-screen"><img src="{{image_gallery->element1-
>element2}}"></a></li>
    </ul>
    <ul>
    {{#quote-list}}
        <li class="b-quote-list_item js-quote-list_item ">
            <span class="b-quote_source">{{quote-list-
>author}}</span>{{quote-list->text}}
        </li>
    {{/quote-list}}
    </ul>
</body>
</html>
```
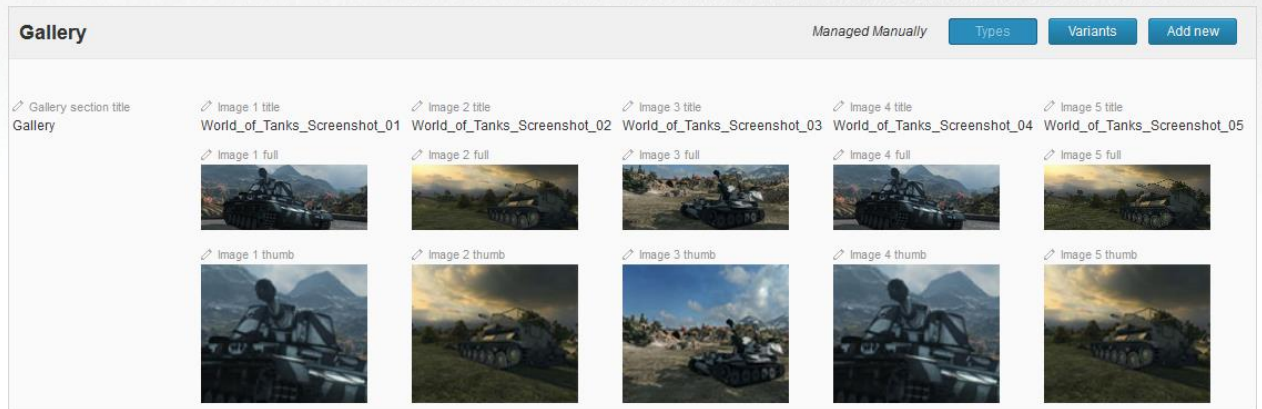
VARIATIVE CONTENT

Content is not a required part of the layout package and is added later via Landing Constructor.

When the layout package includes the default variative content, the content is stored in the `variative_content` directory of the package. The content can include gallery images, background images, text, video, etc.

> The content parameters (image size, text length, etc.) must comply with the attributes specified in the configuration file.

The content varies for different layouts, but several content values must present on any landing page. Depending on the usage frequency, the layout content can be of two types: common and layout dependable.

Common Layout Content

The following content is presented on any landing page and must be included in every layout:
- Title
- Description
- The `favicon.ico` files
- Layout URL
- CTA (Call to Action) button text

  Example:
- CTA button link. Template JavaScript includes different GET parameters for specific content display. When the landing page URL request uses SID parameter, the CTA link must also have SID parameter.

  Example: The original landing page URL is http://lc-eu.wargaming.net/landing_view/51_en3024259839, and the CTA button leads to the page https://na.wargaming.net/registration/en/?game=wot. When the GET parameter `sid` is added, the landing page URL becomes http://lc-eu.wargaming.net/landing_view/51_en3024259839?sid=test, and the CTA button leads to the page https://na.wargaming.net/registration/en/?game=wot&sid=test.

Layout Dependable Content

Most content values and their attributes (size, links, etc.) depend on the layout design. A landing layout can include the following content:

- Graphic elements
  Example: Header, background, promo images of tanks
- Animated flash elements
  Example: Promo images

- Link to the YouTube video file
  Example: Video-files, screen preview
- Screenshot files
- Award panel
- Age rating icons. The number of icons is 0 to 3.
- Text blocks grouped by paragraphs with HTML tags and pseudo tags (i.e. wiki-code) for bulleted or numbered lists, ± 1 to the text body-size, bold or italic font style, links, customized quote style, predefined colors, or tags for expanding/collapsing text on click similar to the [spoiler] forum tag.
- Text rotator
  Example: See the quote rotation in the left text block at http://eu.wargaming.net/play-wowp-1/en/
- Background color and texture of the CTA button in (on mouse over and mouse leave)
- Social network widgets

Content blocks must be of a similar size so they are interchangeable in case their positions are switched.
Example: A social network widget can be replaced by text, flash, or graphical block of the same size.

MIGRATION FROM 2.0 TO 2.1

The layout package for Landing Constructor 2.0 can be used in the version 2.1 with an addition of several modifications.

> The existing landings are not affected by the package changes.

The layout package file system:

- Delete the `no_lang` directory from the package. Place the data from `no_lang` to the Master localization directory.

Configuration file:

- Add a `relation_type` attribute to each placeholder: `one-to-one, one-to-all,` or `unrelated`. If not specified, the system assigns the `unrelated` type to the placeholder with no `relation_type`.
- Add the `master='true'` attribute to the configuration file of the particular localization.

APPENDIX 1. CODING REQUIREMENTS

TPS Requirements

- PHP Coding Standards: http://make.wordpress.org/core/handbook/coding-standards/php/
- WordPress Developer Documentation: http://codex.wordpress.org/Developer_Documentation
- Plugin Development: http://codex.wordpress.org/Writing_a_Plugin
- Theme Development: http://codex.wordpress.org/Theme_Development

Markup Requirements

- Markup validity and compliance with the following coding standards:
    - CSS: http://make.wordpress.org/core/handbook/coding-standards/css/
    - HTML: http://make.wordpress.org/core/handbook/coding-standards/html/
    - JS: http://make.wordpress.org/core/handbook/coding-standards/javascript/
- Markup is tested and debugged.
- Adaptive layout to correctly display landing pages on any device with resolution from 320 to 1980px.
- Semantic makeup where each data block is independent from background. All page blocks are independent and rearrangeable.
- Markup is stable. Any layout block is optional and can be disabled with no influence on the layout. The remaining blocks must retain their size and position.
- Analytics metrics and code are inserted before closing <\body> tag.

For the localization purpose, all descriptions must be inserted in the template as text, not images.

APPENDIX 2. GLOSSARY

base	Server base URL
compound placeholder	Placeholder containing two or more connected elements
content	Text, images, or other information, shared in landings and separate from its structural design
content properties	Layout part introduced as a placeholder with modifiable content for variants creation
CTA button	Call to Action button leading to the particular page. **Example:** Registration page link; starting page for registration via social network or other provider
default content	Content that comes in the layout package when uploaded to LC
landing (landing page, LP)	Promo site aimed to introduce the game to the users and redirect them to the registration page **Example:** http://na.wargaming.net/play2/en/
layout	Main landing template/theme without textual content
LC	Landing Constructor
leaf element	Final element of a compound placeholder
localization	Realm in a particular language **Example:** NA/EN, RU/RU
Master localization	The source localization for creating other localizations. Every layout must have a Master language. Only RU or ENG localization can be assigned as Master.
placeholder	Layout block for adding particular content via LC. Can be simple or compound. The number and types of placeholders is defined in the configuration file.
placeholder element	Placeholder consists of one or several elements. Each element describes the content type to include.
simple placeholder	Placeholder containing only one element
tag	Generic name for an XML element assumed to distinguish any XML element from the placeholder element
TPS	Typical Promo Sites
variative content (content)	Content of the landing page that depends on the layout and design and added to the placeholders of the particular layout.

APPENDIX 3. CONFIGURATION FILE EXAMPLE

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<layout>

    <localizations>
        <localization name="RU" master="true">
        </localization>
    <localization name="EN">
        </localization>
    </localizations>


    <placeholders>

    <placeholder label="description" type="text" relation_type="one-to-one">
            <admintitle><![CDATA[Page meta-description
settings]]></admintitle>
            <elements>
                <text type="text" max_length="200">
            <admintitle><![CDATA[Description]]></admintitle>
                </text>
            </elements>
        </placeholder>

    <placeholder label="title" type="text" relation_type="one-to-one">
            <admintitle><![CDATA[Page title settings]]></admintitle>
            <elements>
                <text type="text" max_length="150">
            <admintitle><![CDATA[Title]]></admintitle>
                </text>
            </elements>
        </placeholder>

        <placeholder label="background" type="image" relation_type="one-to-
all">
            <admintitle><![CDATA[Page background settings]]></admintitle>
            <elements>
                <image type="image" width="3000" height="2000" size="500"
ext="jpg,png">
            <admintitle><![CDATA[Background]]></admintitle>
                </image>
            </elements>
        </placeholder>

    <placeholder label="slogan" relation_type="one-to-one">
        <admintitle><![CDATA[Slogan settings]]></admintitle>
        <elements>
          <image type="image" width="500" height="500" size="500"
ext="jpg,png">
          <admintitle><![CDATA[Slogan]]></admintitle>
          </image>
          <alt type="text" max_length="150">
          <admintitle><![CDATA[Slogan alternative text]]></admintitle>
```

```xml
        </alt>
      </elements>
  </placeholder>

  <placeholder label="quote-list" type="list" relation_type="unrelated">
        <admintitle><![CDATA[Quote list settings]]></admintitle>
          <elements>
            <text type="text-list" max_length="500">
              <admintitle><![CDATA[Quote text]]></admintitle>
            </text>
            <author type="text-list" max_length="500">
              <admintitle><![CDATA[Quote author]]></admintitle>
            </author>
          </elements>
  </placeholder>

  <placeholder label="promotext" relation_type="one-to-one">
      <admintitle><![CDATA[Slogan alternative text]]></admintitle>
      <elements>
        <text1>
      <elements>
        <header type="text" max_length="150">
          <admintitle><![CDATA[Heading 1]]></admintitle>
        </header>
        <text type="text" max_length="150">
          <admintitle><![CDATA[Text 1]]></admintitle>
        </text>
      </elements>
        </text1>
        <text2>
      <elements>
        <header type="text" max_length="150">
          <admintitle><![CDATA[Heading 2]]></admintitle>
        </header>
        <text type="text" max_length="150">
          <admintitle><![CDATA[Text 2]]></admintitle>
        </text>
      </elements>
        </text2>
        <text3>
      <elements>
        <header type="text" max_length="150">
          <admintitle><![CDATA[Heading 3]]></admintitle>
        </header>
        <text type="text" max_length="150">
          <admintitle><![CDATA[Text 3]]></admintitle>
        </text>
      </elements>
        </text3>
      </elements>
  </placeholder>

      <placeholder label="screenshots" type="text" relation_type="one-to-
one">
          <admintitle><![CDATA[Gallery title settings]]></admintitle>
          <elements>
```

```xml
                    <text type="text" max_length="150">
                        <admintitle><![CDATA[Heading]]></admintitle>
                    </text>
                </elements>
        </placeholder>

        <placeholder label="screenshot_gallery" relation_type="one-to-all">
            <admintitle><![CDATA[Image gallery settings]]></admintitle>
            <elements>
                <image1>
                    <elements>
                        <big_img type="image" width="1500" height="2000"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Big image 1]]></admintitle>
                        </big_img>
                        <small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Preview 1]]></admintitle>
                        </small_img>
                    </elements>
                </image1>
        <image2>
                    <elements>
                        <big_img type="image" width="1500" height="2000"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Big image 2]]></admintitle>
                        </big_img>
                        <small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Preview 2]]></admintitle>
                        </small_img>
                    </elements>
                </image2>
        <image3>
                    <elements>
                        <big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Big image 3]]></admintitle>
                        </big_img>
                        <small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Preview 3]]></admintitle>
                        </small_img>
                    </elements>
                </image3>
        <image4>
                    <elements>
                        <big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Big image 4]]></admintitle>
                        </big_img>
                        <small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
                            <admintitle><![CDATA[Preview 4]]></admintitle>
                        </small_img>
                    </elements>
```

```xml
				</image4>
		<image5>
					<elements>
						<big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Big image 5]]></admintitle>
						</big_img>
						<small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Preview 5]]></admintitle>
						</small_img>
					</elements>
				</image5>
		<image6>
					<elements>
						<big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Big image 6]]></admintitle>
						</big_img>
						<small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Preview 6]]></admintitle>
						</small_img>
					</elements>
				</image6>
		<image7>
					<elements>
						<big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Big image 7]]></admintitle>
						</big_img>
						<small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Preview 7]]></admintitle>
						</small_img>
					</elements>
				</image7>
		<image8>
					<elements>
						<big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Big image 8]]></admintitle>
						</big_img>
						<small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Preview 8]]></admintitle>
						</small_img>
					</elements>
				</image8>
		<image9>
					<elements>
						<big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
							<admintitle><![CDATA[Big image 9]]></admintitle>
						</big_img>
						<small_img type="image" width="300" height="300"
```

```
size="500" ext="jpg,png">
                                <admintitle><![CDATA[Preview 9]]></admintitle>
                            </small_img>
                        </elements>
                    </image9>
        <image10>
                        <elements>
                            <big_img type="image" width="1500" height="1500"
size="500" ext="jpg,png">
                                <admintitle><![CDATA[Big image 10]]></admintitle>
                            </big_img>
                            <small_img type="image" width="300" height="300"
size="500" ext="jpg,png">
                                <admintitle><![CDATA[Preview 10]]></admintitle>
                            </small_img>
                        </elements>
                    </image10>
                </elements>
        </placeholder>

    <placeholder label="playforfree" relation_type="one-to-one">
        <admintitle><![CDATA['PlayForFree' label settings]]></admintitle>
        <elements>
          <link type="text" max_length="150">
          <admintitle><![CDATA[Link]]></admintitle>
          </link>
          <text type="text" max_length="150">
          <admintitle><![CDATA[Text]]></admintitle>
          </text>
        </elements>
    </placeholder>

    <placeholder label="cdclient" type="text" relation_type="one-to-one">
      <admintitle><![CDATA['Game Client' label settings]]></admintitle>
      <elements>
        <text type="text" max_length="150">
          <admintitle><![CDATA[Text]]></admintitle>
        </text>
      </elements>
    </placeholder>

    <placeholder label="copyright" type="text" relation_type="one-to-all">
      <admintitle><![CDATA[Copyright text settings]]></admintitle>
      <elements>
        <text type="text" max_length="150">
          <admintitle><![CDATA[copyright]]></admintitle>
        </text>
      </elements>
    </placeholder>

    <placeholder label="support" relation_type="one-to-one">
        <admintitle><![CDATA[Support link settings]]></admintitle>
        <elements>
          <link type="text" max_length="150">
          <admintitle><![CDATA[Ссылка]]></admintitle>
          </link>
```

```xml
        <text type="text" max_length="150">
        <admintitle><![CDATA[Text]]></admintitle>
        </text>
    </elements>
  </placeholder>

  </placeholders>
</layout>
```

Return to Configuration File

APPENDIX 4. TEMPLATE FILE EXAMPLE

```html
<!DOCTYPE HTML>
<html>
<head>
<meta content="width=1280px, initial-scale=1.0, user-scalable=yes"
name="viewport">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta content="{{description->text}}" name="description">
<title>{{title->text}}</title>
<script type="text/javascript" src="js/html5/html5.js"></script>
<script data-main="js/main-combined" type="text/javascript"
src="js/requirejs/require.js"></script>
<link rel="stylesheet" type="text/css" href="js/fancybox/jquery.fancybox-
1.3.4.css" media="screen" />
<link href="css/style_combined_wot.css" type="text/css" rel="stylesheet">
</head>
<body>
<div class="l-bg">
    <div class="l-main">
        <header class="l-header">
            <div class="b-logo"></div>
            <a id="openVideoPlayer" href="#video" class="b-play js-btn-play
js-tooltip" style="display: none;" data-title="{{youtube-player->alt}}"></a>
        </header>
        <div class="l-inner clearfix">
            <div class="l-content">
                <div class="l-users">
                    <span class="b-users">
                        <span class="b-users_strong">{{players-
>amount}}</span>{{players->text}}
                    </span>
                </div>
                <div class="b-content">
                    <a href="{{playforfree->link}}" class="l-button js-sid">
                        <span class="b-button">
                            <span class="b-button_right b-
button_right__big">{{playforfree->text}}
                                <span class="b-button_arrow"></span>
                            </span>
                        </span>
                    </a>
                </div>
                <div class="b-quote b-quote__bubble">
                    <div class="b-quote_container1">
                        <div class="b-quote_container2">
                            <div class="b-quote_container3">
                            <ul class="b-quote-list js-quote-list">
        {{#quote-list}}
        <li class="b-quote-list_item js-quote-list_item ">
        <span class="b-quote_source">{{quote-list->author}}</span>{{quote-
list->text}}
    </li>
      {{/quote-list}}
```

```
                    </ul>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        <div class="b-separator"></div>
        <div class="l-inner clearfix">
    <div class="b-right-col">
        <iframe
src="//www.facebook.com/plugins/likebox.php?href=http%3A%2F%2Fwww.facebook.co
m%2F{{fb-group->group-
name}}&amp;width=290&amp;height=258&amp;show_faces=true&amp;colorscheme=light
&amp;stream=false&amp;show_border=false&amp;header=false" scrolling="no"
frameborder="0" style="border:none; overflow:hidden; width:290px;
height:258px;" allowTransparency="true"></iframe>
    </div>
            <div class="b-left-col">
                    <ul class="b-gallery">
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image1-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image1-
>small_img}}" alt="World of Tanks">
            </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image2-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image2-
>small_img}}" alt="World of Tanks">
            </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image3-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image3-
>small_img}}" alt="World of Tanks">
            </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image4-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image4-
>small_img}}" alt="World of Tanks">
            </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image5-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image5-
>small_img}}" alt="World of Tanks">
            </a>
        </li>
            <li class="b-gallery_item">
```

```html
                <a rel="example_group" href="{{screenshot_gallery->image6-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image6-
>small_img}}" alt="World of Tanks">
                </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image7-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image7-
>small_img}}" alt="World of Tanks">
                </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image8-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image8-
>small_img}}" alt="World of Tanks">
                </a>
        </li>
            <li class="b-gallery_item">
            <a rel="example_group" href="{{screenshot_gallery->image9-
>big_img}}" title="" class="b-gallery_link js-gallery-link">
                    <img class="b-gallery_img" src="{{screenshot_gallery->image9-
>small_img}}" alt="World of Tanks">
                </a>
        </li>
    </ul>
        </div>
        <div class="b-center-col">
            <h1 class="b-heading1">{{promotext->text1->header}}</h1>
            <ul class="b-list">
                <li class="b-list_item">
                    <p>
                        {{promotext->text1->text}}
                    </p>
                </li>
            </ul>
            <h2 class="b-heading1">{{promotext->text2->header}}</h2>
            <ul class="b-list">
                <li class="b-list_item">
                    <p>
                        {{promotext->text2->text}}
                    </p>
                </li>
            </ul>
            <h2 class="b-heading1">{{promotext->text3->header}}</h2>
            <ul class="b-list">
                <li class="b-list_item">
                    <p>
                        {{promotext->text3->text}}
                    </p>
                </li>
            </ul>
        </div>
    </div>
```

```html
        <footer class="l-footer">
            <div class="l-inner clearfix">
                <ul class="l-rating">
            <li class="b-rating">
            <a href="http://www.esrb.org/ratings/ratings_guide.jsp" class="b-
rating_link b-rating_link__esrb">ESRB</a>
        </li>
</ul>
                <a href="{{support->link}}" class="b-support-link">{{support-
>text}}</a>
                <div class="b-footer">
                    <a href="{{logo->link}}" class="b-logo-wg">{{logo-
>text}}</a>
                    <p class="b-footer_copyright">
                        <span class="b-footer_txt">{{copyright->text}}</span>
                        <a href="{{cookies-policy->link}}" class="b-footer-
link">{{cookies-policy->text}}</a>
                    </p>
                </div>
            </div>
        </footer>
    </div>
</div>
<div style="display: none;">
    <div id="video"  class="b-overlay js-overlay" style="height: auto; width:
auto">
        <div class="js-youtube-player" data-videoId="{{video->id}}">
            <div id="youtube-player" >
                {{youtube-player->requirements}}
            </div>
        </div>
        <div class="b-overlay_panel">
            <a class="l-button l-button__indent js-sid" href="{{playforfree-
>link}}">
                <span class="b-button">
                    <span class="b-button_right b-
button_right__big">{{playforfree->text}}
                        <span class="b-button_arrow"></span>
                    </span>
                </span>
            </a>
        </div>
    </div>
</div>
    <noscript>
        <iframe
src="//www.googletagmanager.com/ns.html?id={{googletagmanager->key}}"
        height="0" width="0" style="display:none;visibility:hidden"></iframe>
    </noscript>
    <script>
        (function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
        new Date().getTime(),event:'gtm.js'});var
f=d.getElementsByTagName(s)[0],
        j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'//www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
```

```
        })(window,document,'script','dataLayer', '{{googletagmanager-
>key}}');
    </script>
</body>
</html>
```

Return to Template