QA Overview

As a company our goal is to provide our customers with the best product we can. In order to accomplish this goal, our products undergo Quality Assurance testing to ensure they meet this standard prior to being released to our customers. This document outlines the procedures, steps and tools used by QA to further this goal.

QA Systems Used

In order to accomplish the ongoing goal of QA there are systems, software and equipment that are used both in the Development environment and in the Live environment. These tools are primarily (but not limited to):

Development

- Intranet
- Dev demo
- Dev ticket system

Live

- Intranet
- Live demo
- Live ticket system

Other

- Google Drive (former Google Docs)

Software

- Internet Explorer (latest version)
- Firefox – Mac and PC
- Chrome – Mac and PC
- Safari
- Snag It
- WinMerge
- Photoshop
- FTP Software (there are several clients available, IT will set up your FTP login)

- Synergy (Allows the use of a single mouse and keyboard for both the Mac and the PC)
- Google Talk (Optional – Gtalk may also be accessed via the webmail interface)
- Notepad ++ (Optional)

QA Specialists will also utilize the following tools:

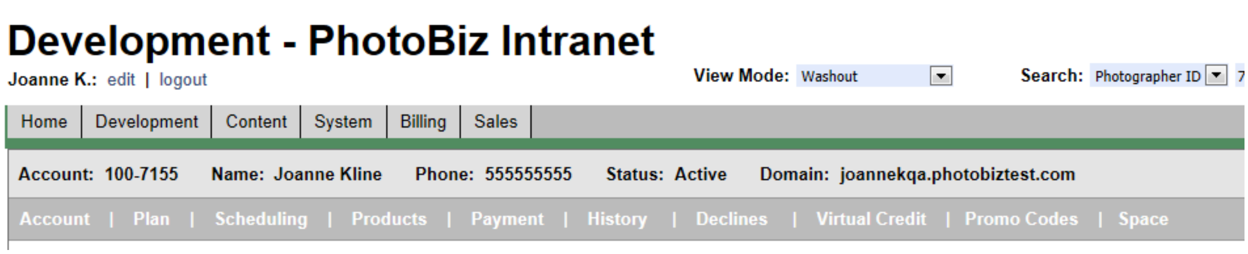
- PC set up with dual monitors
- Mac set up with a single monitor
- iPad tablet
- Android tablet
- iPhone

Some of these tools will be specifically assigned to the Specialist, whereas others may be a shared resource among the other QA Specialists and/or other departments within the Technical Services division (Design, Development, IT). Specialists may also occasionally rely on personal resources such as one's own iPhone or Android device to view products.

One important tool you will use in the QA process is other QA Specialists. Sometimes you need a set of fresh eyes on a problem. Other times you may need a different perspective on an issue, or another Specialist might have encountered a similar problem. Asking other members of the QA team and brainstorming with one another can be an important part of the QA process.

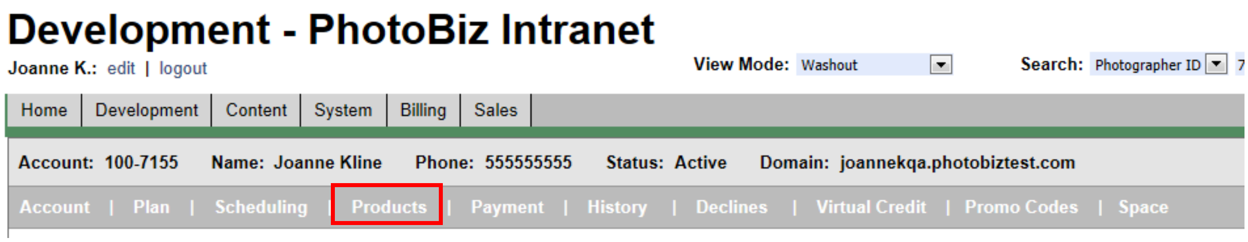
The Intranet

The Intranet is a core component for QA. The Dev intranet is used to assist in the testing environment, and the Live intranet is used for the development task system and the error log. We'll take a quick overview of these systems and touch on the most common uses for them.

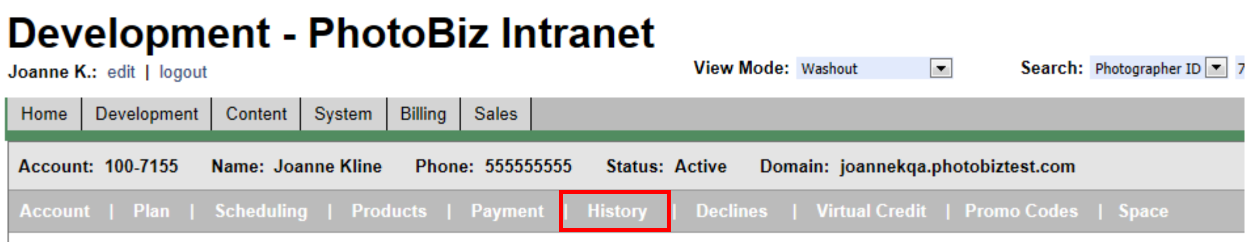


Development

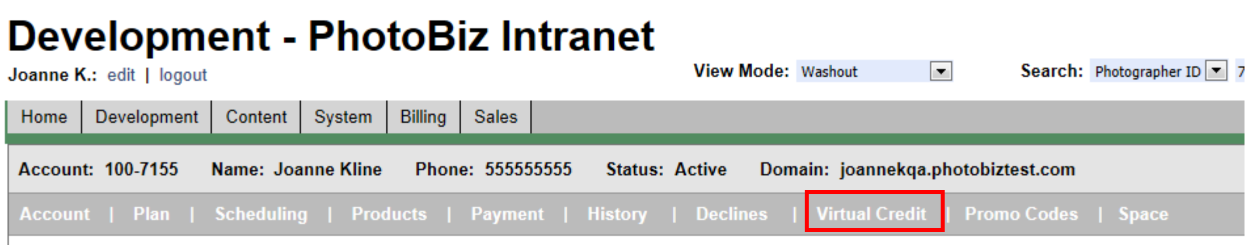
QA Specialists are able to access a wide variety of control features in the Dev intranet. Some of the areas most commonly used by QA:



Products – The Products section allows the viewing of all products on a particular account. Products may also be removed from the account in this section. This is often used during the testing process where the Specialist may add and remove products and product options. This is done to ensure the add and removal process works correctly.

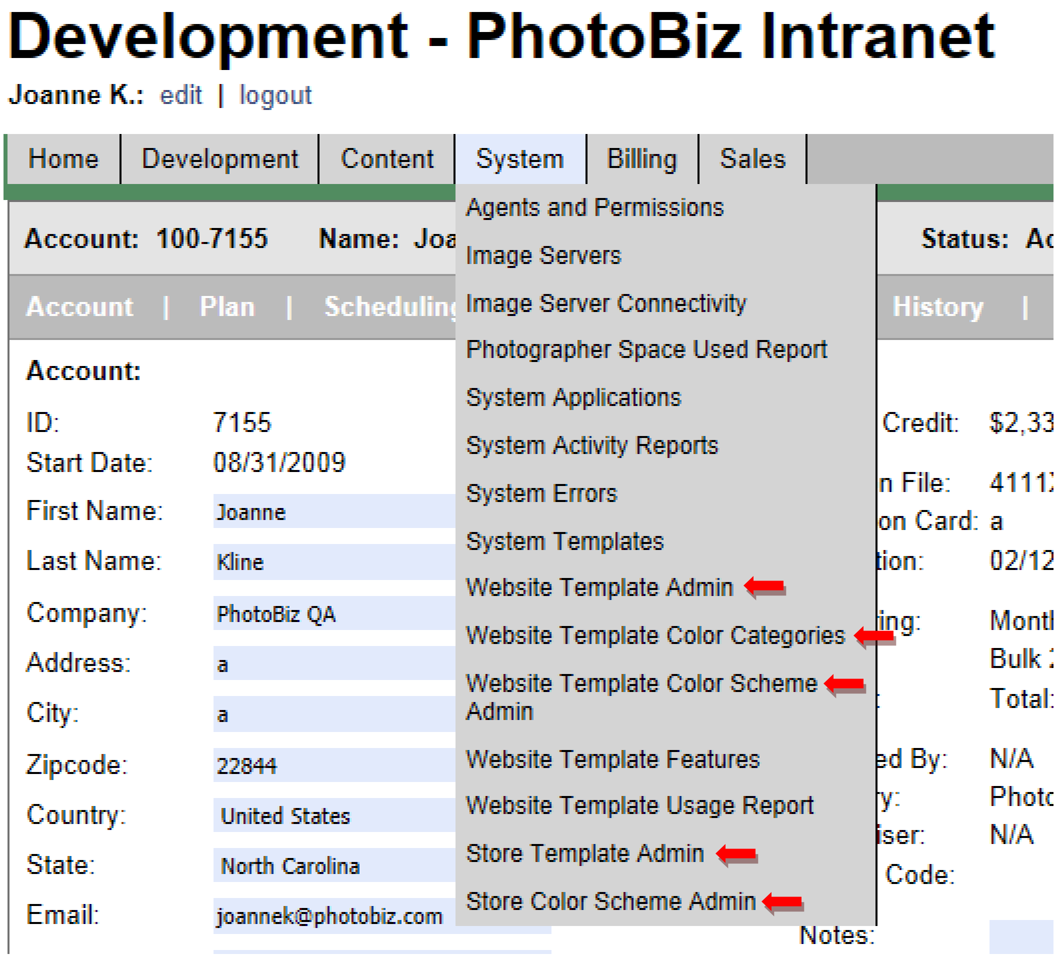


History – While this area is not used often during the QA process, it can be useful at times to look at the history of the account especially when trying to troubleshoot an issue with product removals.



Virtual Credit – This section is used to add virtual credit to a demo account which can be used to purchase products and enhancements for the testing process. QA specialists may add VC as needed to their demo accounts for testing purposes.

Additionally, the System tab is where the Flash and Store/Blog template and color scheme admin panels can be accessed. The areas most commonly used during the testing process are highlighted below and discussed in more detail within their relevant sections.

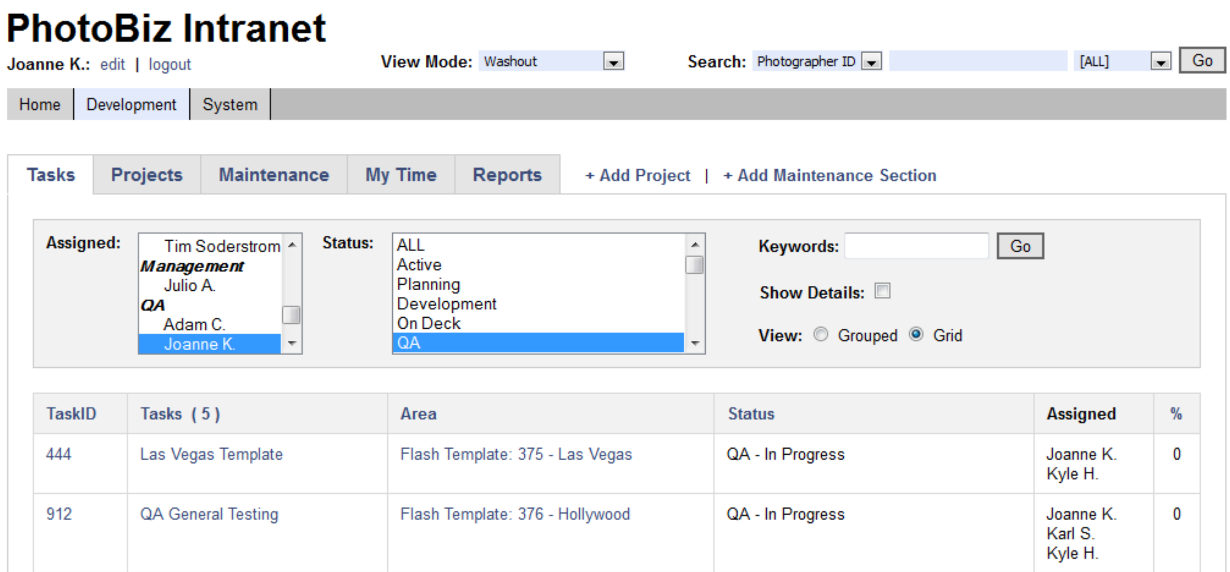


Live

The Live intranet allows access to both the Development task system and the error log. The task system is where all development-related tasks are tracked. This includes both projects and maintenance items.

Development Tab

The task system is found under the Development tab. This area allows for the creation and tracking of Development tasks.



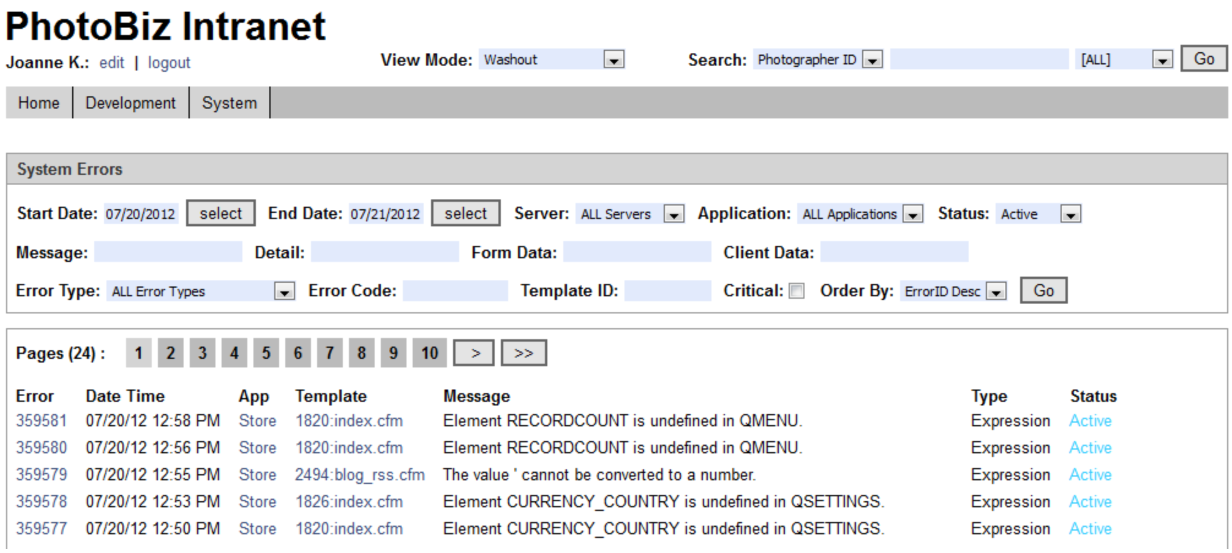
The **Tasks** tab allows you to easily keep track of tasks assigned to you. You can also use the sort and search features to view all tasks of a particular type, or search for specific tasks.

The **Projects** tab shows project categories. A project will generally have tasks within it related to that project. While the Projects tab doesn't have a specific search feature, you can narrow the viewing parameters and use the browser search feature to find specific tasks you're looking for. This system makes it easy to keep all tasks related to a project in a single place.

The **Maintenance** tab contains categories for post-release fixes, generally "bug" fixes. It may also contain items which were discovered during pre-release product testing but have been slated for post-release correction. From the maintenance tab you can click on a product sub-section to view maintenance tasks for that product. This area includes a search feature as well.

System Tab

The live error log can be accessed by clicking on System Errors, under the System tab. Whenever someone attempts to work in the control panel and receives an error message, most of the time the error is recorded in the error log. This allows QA and Development to review the error information as an aid to troubleshooting. While not all problems will generate an error log entry, many do and the error log will show the file and area where the error occurred.



The Test Environment

Setting up your testing environment

There are a couple of different types of account you will need to create in order to test items in the dev environment. Note that this is not an all-encompassing list of account types; rather it's what basics you will need to start out with. Your specific testing needs will dictate what additional account types and pages you may need to add.

For all products you will need to purchase expanded packs so that you can add and remove pages as you need. You can find information on how to add virtual credit to your account in the Development Intranet section of this manual.

Main Flash/HTML Demo

You will need multiple pages of each type to easily replicate various configurations you will be testing for. The easiest way to accomplish this is to create drop-down menus for each page type, and nest all your page type configurations into their respective DDM. Note: Some types of test configurations can be combined to cut down on the number of pages. For example, you can place your custom meta title on your color-shifting or caption page. You can review other QA Specialists' demo accounts for ideas.

- Home
  - No image
  - 1 image
  - 1 image with a color shift

- 2 images
- Multi-image (about 20 images)
- Multi-image with Color-Shifting
- Multi-image with Captions
- Custom meta titles
- Music

- Galleries
  - No image
  - 1 image
  - 2 images
  - Multi-image with left thumbnails (about 20 images)
  - Multi-image with right thumbnails (about 20 images)
  - Multi-image with Color-Shifting
  - Multi-image with captions
  - Custom meta titles
  - 100 image gallery
  - 200 image gallery
  - 300 image gallery
- Information
  - No image
  - 1 image – right aligned
  - 1 image – left aligned
  - 2 images
  - Multi-images (about 20 images)
  - Multi-images with Color-Shifting
  - Multi-images with no text
  - Custom meta titles
  - Long page title (50 chars)
  - No page title
- List/Calendar
  - Testimonials Page – Multiple pages with thumbnails
  - Testimonials Page – No Thumbnails/Custom meta titles
  - Testimonials Page – Long Title
  - Testimonials Page – No Title
  - Links – Multiple pages with thumbnails
  - Links – No Thumbnails/Custom meta titles
  - Embedded Links – Multiple pages with thumbnails

    - Embedded Links – No thumbnails/Custom meta titles
    - Calendar – Text/Custom meta titles
    - Calendar – Right aligned
    - Calendar – Long page title
    - Calendar – no page title
    - Calendar – no text
- Contact
    - No image
    - 1 image, right aligned
    - 1 image, left aligned
    - 2 images
    - Multi-image
    - Multi-image with Color-Shifting
    - Custom meta title
    - Long page title
    - No page title
    - Image, no form
    - No image, no form, right-aligned
    - No image, no form, left-aligned
- External Links
    - External Link – Lightbox
    - External Link – Embedded Video
    - External Link – New Window
    - Empty External Link

You should also create a Flash subsite in your demo account with no content, and no changes made to the default standard account creation. You will use this account when testing Flash/HTML to test new templates for account creation related problems.

Additional Demos

You may find it easier to create specific demo subsites which you can set up to allow you to quickly test certain scenarios. You can create as many subsites under your demo as you need or want to facilitate your testing environment. Below is a list of types of demo subsites you may find it useful to create:

- Testing slideshow speeds
- Testing various types of content such as HTML, 1500x865, long titles, special characters, long menus, etc.
- Testing Intrasite links

- Testing deeplinking/SWF addressing

Store/Blog Demo

When content building your store and/or blog, the main thing you want to focus on his ensuring you have sufficient content to test various scenarios, both common and unusual. Just as with the Flash, it is acceptable to combine types of test scenarios into a single item (e.g. Events with color shifting and an access code):

Store Tab

- Products, Services, Packages
  - Create multiple categories
  - Create categories both with and without category images
  - Create enough entries in at least one category to extend across multiple pages
  - Create items with long titles (50 chars)
  - Create items with long titles, all in caps (50 chars)
  - Create items with multiple images
  - Create items with and without options
  - Create items with varying amounts of description text, including none
  - Create items with HTML embedded in the description text
  - Create expired items
  - Create digitally delivered items
- Prints
  - Add enough print sizes to extend to multiple control panel pages
  - Add custom print options
- Invoices
- Gift Cards
- Events
  - Create multiple categories
  - Create categories both with and without category images
  - Create enough events in at least one category to create multiple pages
  - Create enough events in a category to extend past 10 pages of events
  - Create events with long titles (50 chars)
  - Create events with long titles, all in caps (50 chars)
  - Create events with multiple galleries
  - Create events with single images

    - Create events with custom price lists
    - Create expired events
    - Create events with color-shifted galleries
    - Create events that do not use the default event settings
    - Create events both with and without access codes
- Create multiple price lists that do not have all options
- Create shipping
- Create taxes
- Create at least one offline payment merchant option so that you can test orders
- While you may not need to set up discounts, campaigns or custom order settings up front, chances are you will need to create and delete them on the fly as you run through various types of tests.

Blog Tab

- Create some blog post entries
    - With and without images
    - With and without tags
    - With a single tag
    - With multiple tags
    - With various types of text formatting (alignment, font size, tables, HTML, etc)
    - With images used from the clipboard
- Add some comments on the front end and make sure they are visible from the Comments section.
- Post tags can be created in the Tags section, or at the time of the post.
- Check the Settings section for display time, social media icons, comment and RSS feed settings. Some of these will be changed while you test.

Site Builder Tab

Like the Flash, you can have pages on the base menu, or group in drop-downs. You can also group multiple types together such as adding an Access Code to a page with HTML content so that one page type can show you multiple scenarios.

- Setting up a Home page (Marquee/Marquee Plus) is useful as it gives you a fixed reference point to see if links you test are taking you to the main page for the Store, or to a specific page.
- Information Pages
    - With Special Characters
    - With long navigation name and page title

    - With large amounts of text
    - With HTML Content
    - Locked, using an Access Code
- Marquee Pages
- Marquee Plus
- Gallery
- Packages/Products/Services
- Workshops
- Gift Cards
- Invoices
- Contact
- RSS Feeds
- List Items
- Embedded Link
- External Link
- Embedded Video
- Blog

Creating New Demo Accounts

To create extra demo accounts on Dev (for all product types), go to http://development.photobiz.com and use the sign up form on the Development version of the corporate page, just as a new customer would on the Live version. Use the following dummy credit card information:

Name: Use any name
Card#: 4111111111111111
Expiration: Any date later than the current day.

You can also use this dummy credit card information to create new accounts when your testing needs require a fresh, new site.

Note: This dummy CC information will only work in the Development environment.

Bug Tickets

By definition, a bug is an issue with our system that cannot be solved by the customer or agent via the Control Panel, Ticket System (Level 2+) and/or DNS Management. There is a Support process in place for determining if an issue is a bug, and in reporting that issue to Quality Assurance.

Bug Ticket Lifecycle

- Rec'd from support – when support either receives a contact from a customer or in some cases notices an issue themselves, they will attempt to troubleshoot and resolve the issue directly. If they are not able to do so, the issue will be escalated to a team lead. When the team lead confirmed the issue, they will then send the issue on to QA.
- Tested and confirmed by QA – QA will attempt to recreate the problem and determine if it can be fixed by QA. If so, QA will generally return the issue with instruction on how to correct the problem. If not, QA will assign the issue to development for resolution.
- Assigned to Dev – dev will correct the issue.
- Fixed – once the issue is fixed it will be returned to QA for testing.
- Fix tested by QA – QA will retest the issue to make sure it is resolved. If the issue is not resolved or the fix creates new issues, QA will work with dev until the issue is corrected. Once the issue is corrected and the fix works properly QA will flag the issue for deployment to live.
- Deployment – uploaded to live.
- Live Review – QA will review the fix on live to ensure everything's deployed and it works correctly.
- Returned to support – once the fix is confirmed on live QA will update the task and notify support via updating the tickets and returning them to the team lead with any relevant information.

Working With Bug Tickets

In working with bug tickets your primary contact with Support will be via the Support Team Leads. In the course of working with the agent, the Team Lead may confer with QA. Occasionally, a Team Lead will instruct an agent to work directly with Quality Assurance. This is usually done to ensure that important information is not lost or misconstrued in the process of passing it back and forth through a third party. Note that in most circumstances, Support Agents should not be going directly to QA except at the direction of a Team Lead (or Level 2 if there is no Team Lead available).

Most of the time the issue will be resolved by Support either through troubleshooting or via customer education. However, if there is a problem that cannot be resolved by Support or the Team Lead, the Team Lead will escalate the issue to QA in the form of a bug ticket. You can refer to Chapter 10 of the Customer Support training manual for more information on Support's troubleshooting process.

Bug tickets will be found in the ticket system, under the Development tab. Bug tickets must be vetted by a Team Lead before being assigned to QA. If the ticket does not include notes from the Team Lead indicating the issue has been vetted, reassign it back to the Support agent under the Support tab and indicate in the notes that the issue must be vetted by a Team Lead before it can be assigned to QA. Use the status “New (Have to Call)”. This will flag the ticket as urgent in the ticket system.

When you open a bug ticket the first thing you will need to do is read through the ticket to determine the problem. Bug tickets from support must include at the very least the customer’s browser and version, OS and version, what product the problem occurred on, and if relevant the template collection and name. The ticket should also include information from the agent on what the problem is, what steps they took to try and troubleshoot, and whether they were able to recreate the issue in their own demo accounts. If they did recreate the issue, they should also include the customer ID for their demo account. If there is missing information, or the information given is inadequate for you to begin troubleshooting the issue you may at your discretion return the ticket to the Team Lead with a request for more information than was provided. The Team Lead will then get with the agent to ensure the agent obtains the requested information. Once the ticket has been updated with the requested information, the Team Lead will assign the ticket back to you.

Once you have the information you need, you can begin attempting to resolve the problem or determine the trigger, if you cannot solve the problem. This includes trying to recreate the problem in your Dev demo. It’s impossible to list every possible scenario you may encounter while attempting to locate the root of a problem, but here are some key points to keep in mind while working with an issue:

- Can you recreate the problem on your Dev demo?
- If not, can you recreate the problem on your Live demo?
- If it involves text display, check anywhere you can enter text for special characters.
- If you don’t find any, try copying text from text editors into Notepad++. Sometimes that allows you to see “hidden” characters Notepad doesn’t display.
- If you can recreate the problem, try playing around with the issue to see if you can figure out what exactly triggers it. Change settings, delete/reupload logos, try different text, etc.
- Check the error log.
- Try and reproduce the issue in Firefox, with Firebug enabled. This sometimes lets you see errors that don’t appear in the browser itself.
- Talk to other QA Specialists, they may have insights that help you.

You may determine the problem is related to something you can easily correct, such as a hidden character in a text editor. Many of these issues can be resolved by Support as well, and may have been overlooked. If you are able to resolve an issue, update the ticket with information on how you solved it, and set it back to the Team Lead. Use the status “New (Have to Call)”.

If you are unable to resolve an issue, or you determine through troubleshooting that an issue is being caused by a bug, you will need to create a task for the issue to be addressed by Development. Bug tickets should be set up as follows:

- Include the ticket ID in the title. If there is more than one ticket ID, put the rest of them in a note.
- Status should be “QA – Confirmed Issue”.
- Assign the ticket to both yourself and Adrianna N.
- Include a complete description of the issue: What is occurring, how you reproduced it, steps to reproduce, etc. If you have relevant screenshots you can upload them to the QA FTP server and link them in the Links section. It’s also a good idea to include a link to your demo, to the page the issue is occurring on.

Once Development corrects the issue, they will assign the task back to you with a status of “QA – Returned from Development” along with any informational notes. If there are no notes, that generally indicates the developer was able to correct the issue without need of further clarification. You will then re-test the issue. If the issue has been resolved, note that in the task. If the issue has not been corrected, indicate how you determined that the issue was not corrected in the notes and reassign it back to the developer with the status “Development – Returned from QA”. It is important to be very specific in your return notes so that the developer understands how you determined that the issue is still occurring.

Once you have determined that the issue is corrected and you have confirmed the issue on Dev, set the status of the task to "Deployment - Ready". After the fix is deployed, the task will be set to "Live Review". Re-test the issue on Live to ensure that the fix is working on Live correctly. If the fix is working properly, update all the support tickets referenced in the task and assign them back to the Team Lead. The Team Lead will review the information and then have the agent follow up with the customer to let them know the issue has been resolved.

If the fix does not function properly on Live, check to see if the issue is occurring on the Development servers as well. If the issue is occurring on Dev it may have been missed during testing. You will need to reopen the Maintenance task for Dev to address the issue. If the issue is not occurring on Dev, it may be due to a missed file. If you determine that a file has been missed, you will need to speak to the developer and show them what is occurring. They will need to get with Julio and determine the best way to ensure the file gets deployed properly.

Project Testing

Projects can vary widely in size and scope. A project can involve anything from a simple single enhancement to an existing product, all the way up to a completely new product that requires the entire

QA team to test. The goal of QA in project testing remains to ensure that the products we release meet our standards.

Project Lifecycle

- Planning/Design – Planning and Design determines the scope and needs for the project, and provides the guidelines for development in how the project will need to take shape and form.
- Development – The project is turned over to Dev to perform the actual coding.
- QA – First-round testing. Depending on the size and scope of the project, this may involve one or multiple QA agents.
- Design Review – At this point design will review the project, and will request any necessary changes from Development. Any changes will then need to be put through the QA process.
- QA – Second-round testing: If there is to be a second round of testing it is usually done at about this point.
- Deployment – Once all testing and review is completed, the project is deployed to the live servers. Smaller projects may be a matter of simple file uploads; larger projects can involve deploying to one live server at a time with each being tested individually, followed by overall live testing once all live servers are back on line.
- Live Review – reviewing the product on the live servers to ensure everything works properly as it did in the development environment. This ensures that we catch any potential issues that may not have shown up on the live servers, such as issues caused by missing files, etc.

Project Testing Guidelines

For projects you will generally not have a checklist. Below are general guidelines for testing both with and without a checklist – note that while not all of them may apply to every product this will give you a general idea.

Control panel

- Check long text entries in text box fields
- Check special characters in text box fields
- Check empty text box fields
- If a text box field has default information, check it with the information removed
- Check all control panel elements to make sure they function correctly, ie: button highlights, thumbnail previews, etc.
- Check to make sure the correct elements highlight relative to the section you are in.
- Ensure that elements designed to be dragged and dropped work properly

- Ensure that removable elements can be deleted properly
- Check that pages can be added and removed
- Check image uploading and deleting
- Check clipboard
- Check image count
- Check error messages
- Check layout graphics
- Check CP pagination
- Ensure fields that pre-populate do so consistently and correctly
- Create and remove items with and without images
- Ensure settings changes can be saved
- Ensure meta information can be saved
- Ensure default meta information is correct
- Check that all Settings sections work properly
- Check that reports return correct data

Front End

- Check overall page "look"
- Ensure all design elements display correctly
- Check all layouts/templates
- Check all option configurations
- Check text content to ensure links and formatting work properly
- Check limited text content areas to make sure text truncates properly
- Look for design consistency with other similar page types
- Look at template elements to make sure design matches
- Check various fonts – look for size/style interactions
- Check colors and classes
- Look at margins, borders, boxes, etc.
- Check with both logo and text headers, all alignments
- Check links and link functionality
- Make sure all visible elements display
- Make sure non-visible elements correctly remain hidden
- Check text size, alignment, formatting, links, color, etc
- Check image links for clickability

- Check access code functionality
- Check transitions for multi-images
- Check settings changes are correctly reflected on-page
- Check image sizes
- Check optional design elements such as backgrounds, music, etc

Template Testing

Part of the appeal of our products is our template system. By providing a choice of template styles and color schemes for our Flash, HTML5 and Store/Blog products we give our customers the ability to keep their sites looking fresh and interesting. Therefore a major part of the QA process involves testing new templates for these products prior to their release.

When testing Flash, HTML5 and Store/Blog templates you will be working with a checklist. The checklist will help you to focus in on the specific areas of each template type that need to be tested, however you will also want to remain mindful of the general guidelines above. It is entirely possible to discover an issue or problem with a template in a scenario not covered by the checklist. The main idea with template testing is to be as thorough as possible while remaining efficient in your use of time.

Template testing involves the use of a checklist which will guide you through the various testing scenarios. The checklists include detailed information to help you look for problem areas. We'll touch on these checklists and the overall testing process for each template type below.

Note: All checklists and information documents can be accessed via the links on the QA Docs portal: https://sites.google.com/a/photobiz.com/portal/qa-docs

Flash and HTML5 Template Testing

Flash templates begin as a design concept. Once the concept is approved, it is sent to Design where it enters the planning stage. During this stage, Design will create the project in the Intranet and create a planning task within the project which lays out the specific design parameters for Development. At this point the template is in Development's hands for coding.

Once the template coding is complete, Design will do a review and will work with Development to complete any design changes. After this process is completed, Design will assign the template to QA for first round (Black Box) testing.

Pre-Release

Template testing utilizes a checklist to help you test a full range of scenarios with template functionality. You should print out a new template checklist at the start of testing as this document is frequently updated. This insures you are working with a checklist that includes new issues to check for as well as known potential issues.

Before you begin testing, take a few moments to review the Planning, Coding and Design Review tasks as well as the QA General Testing task. These tasks often contain valuable information about the design specifications for the template along with any design changes implemented after the example screenshots were created. This will help you to know exactly what you need to see on the template. If you need clarification on any items, check with Design.

Next, fill out the information blanks at the top of the checklist form. You can obtain the Template Number and the Developer Name from the Template Features document. The Template Name will be displayed in the QA General Testing task. The Template ID can be obtained by clicking on the Preview button for the template in the Templates tab. You can find it in the URL string, right after "templateID=". Both the Template Number and the Template ID should be included on the checklist.

Once you have all the information on the form and you have reviewed the design tasks, you are ready to begin with first round testing. There are several areas you will be checking over. It is not necessary to check all the sections in the order they are printed, as long as all sections are checked in the course of testing. Let's do an overview of each section.

General

The General section deals with items that affect the overall template: Template size, template dimensions/ratio, footer, branding, music bars, design consistency, intrasite linking, overall template behavior. The template design should remain consistent across all pages, and with other similar templates. Many of the items you will be testing in this section will require you to publish your demo site. For that reason you may find it easier to test this section after testing individual pages.

Features (Music/FS/Icons)

This section deals with the way music behaves on the templates, how the icons appear and how the template handles Flash Full Screen Mode (FFSM). Pay particular attention to how the icons and music bars align. Many of the items you will be testing in this section will require you to publish your demo site. For that reason you may find it easier to test this section after testing individual pages.

Color-Shifting

This section deals with color shifting settings on all pages and how they function. Many of the items in this section can be checked as you are going through the individual pages, such as checking the color shifting on each page type, and checking each page type as a color-shifted top level page. When checking menu items with color shifting, make sure to check them moused-over as well.

Pages

Includes subsections for all page types: Home; Gallery; Information; Contact; Calendar; List (Testimonials, Links, Embedded Links); External Links. These sections deal with how text and images function on pages and how pages handle transitions, paging, slideshow speeds, forms, and other aspects of individual page behavior. These sections will include both site-wide page and page-specific behaviors; make sure to go over each checklist page section carefully during the testing process.

Menu

This section deals with menu behavior such as highlights, drop-down menus, the slider, and how menus are affected by color shifts. A note on menu terminology: “Highlight” refers to the color border or highlight that appears around the currently selected page. “Slider” refers to the animated color border or highlight that follows your pointer as you mouse over the menus. The slider generally uses the same colors as the highlight.

Fonts

This section deals with fonts and how they appear for both Global and Content font areas. You will want to check at least two different fonts, at a variety of sizes in all areas. Display type fonts and non-standard font styles tend to produce the most issues.

Color Schemes

This section deals with how color schemes affect template elements. Pay particular attention to how elements look in different color schemes; we want to make sure there are no hard-coded colors.

Control Panel

This section deals with settings and how settings function for the template. Ensure that all customizable elements of a template respond to control panel settings and changes.

Template Admin

Most of this section is currently handled by Design; however on Flash templates QA will verify the color scheme categories during the Regression testing phase.

When you find an issue with the template, you will need to create a task for Development to address the issue. The task should be assigned to the Developer handling that particular template with the status "Development – Assigned". Be as specific as you can: Describe exactly what the problem is and how it appears, list the steps to recreate the problem, provide links to the page where the problem occurred, etc. Make sure to include any relevant screenshots and links that will help the developer see what problem you are experiencing. Screenshots are particularly important with issues that occur only on the PC, as our Flash developers work primarily in a Mac environment. .

Once Development completes a task and corrects an issue with the template, they will assign the task back to you with a status of "QA – Returned from Development" along with any information notes. If there are no notes, that generally indicates the developer was able to correct the issue without need of further clarification. You will then re-test the issue. If the issue has been resolved, note that in the task and set the status to "QA – Completed". If the issue has not been corrected, indicate how you determined that the issue was not corrected in the notes and reassign it back to the developer with the status "Development – Returned from QA". It is important to be very specific in your return notes so that the developer understands how you determined that the issue is still occurring.

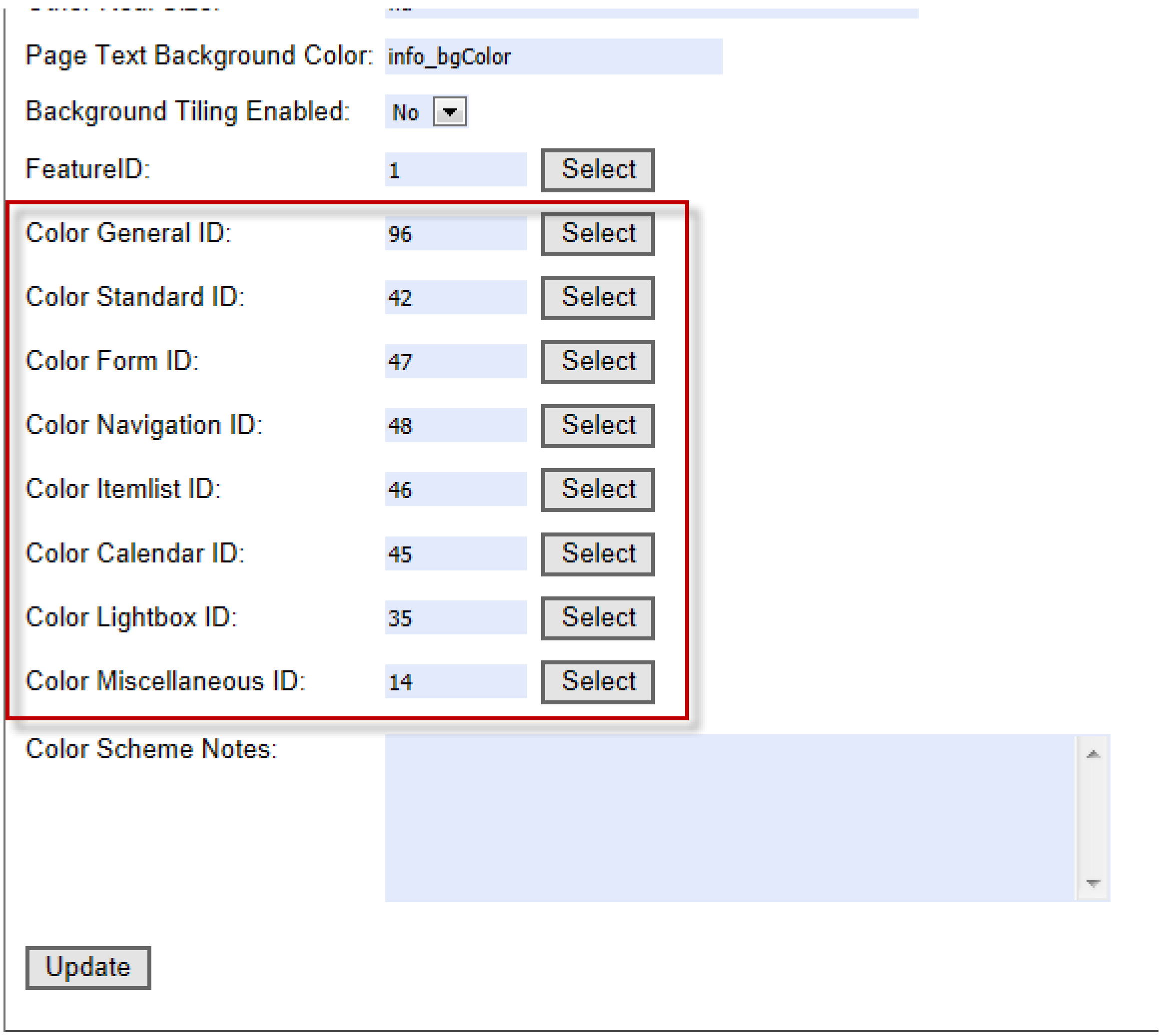
Once all of the issues you have found are corrected and all the tasks you have created have been confirmed, you can advise Design that the template is ready for the second round (Regression) testing.

Regression testing is handled by a second Specialist, and follows the same sequence as Black Box testing. As with first round testing, you will create tasks as you discover issues. Once those issues have been corrected and confirmed, the Regression tester will confirm the color schemes.

The Color IDs need to be checked to ensure all visible colors have a corresponding selector in the color picker, and that there are no selectors visible that are not used for that template. You can use the color picker to see what colors are being used in the template. The easiest way is to have the template open on one monitor, and the Control Panel with the color picker open on a different monitor. Compare the template against the color picker. Make sure that you can see all the colors in the color picker in use on the template, and that all colors on the template can be changed using the available fields in the color picker.

Next you will need to verify that all the color categories are correct for the specific template as each template is different. If the available selections need to be changed you will need to determine the correct fields to be used.

In the Dev intranet, go to **System > Website Template Admin** and select the template you are checking. Scroll down toward the bottom where the color IDs are:



In general the best pages to check each color category section with are:

General - *Check Home and Gallery pages*
Standard - *Check Contact and Info pages*
Form - *Check Forms*

Navigation - *Check Menus*

Itemlist - *Check List pages (use the Item List alternate color scheme)*

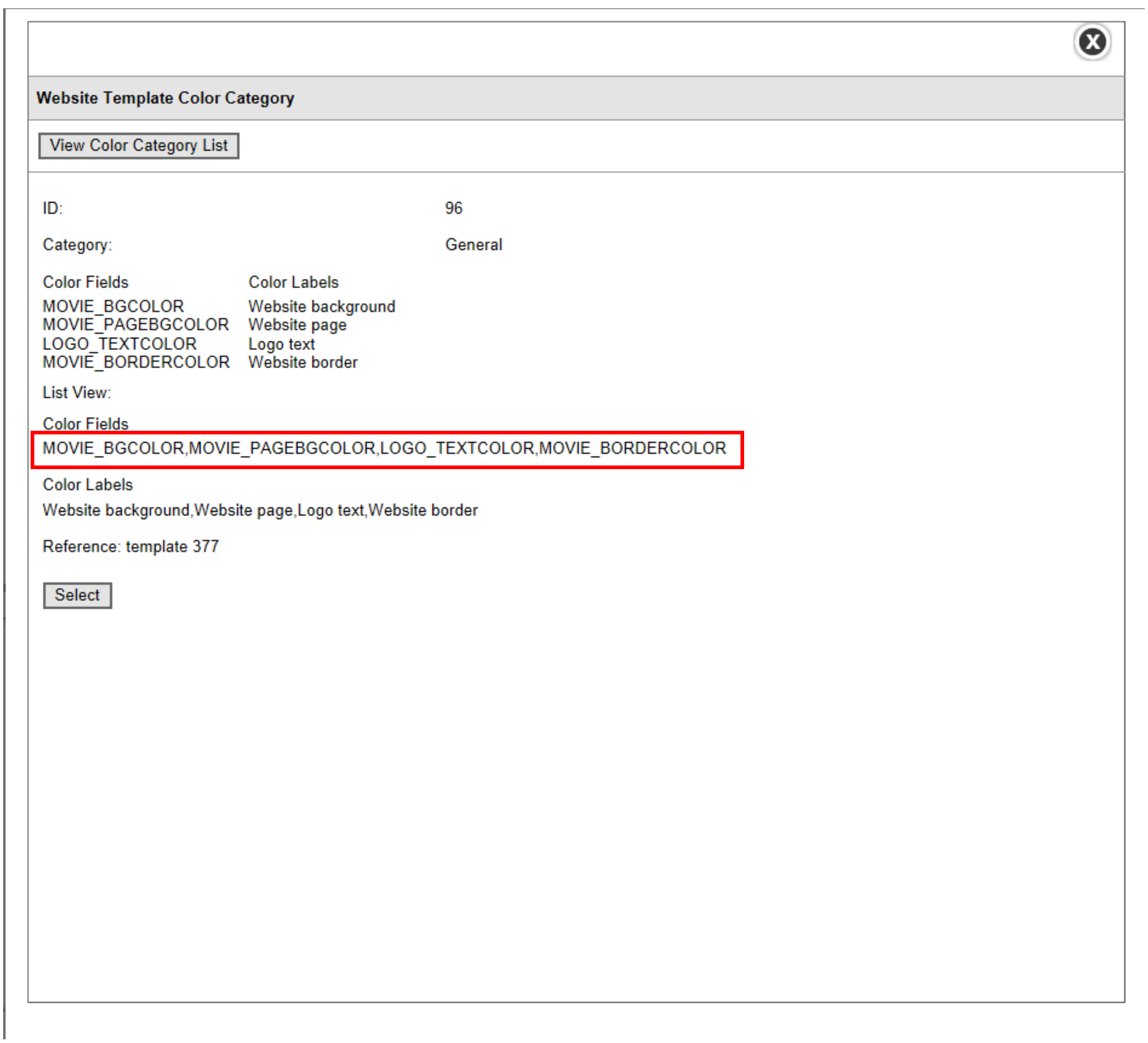
Calendar - *Check Calendar pages (use the Item List alternate color scheme)*

Lightbox – *These settings are the same on all templates, do not need to check.*

Miscellaneous - *Check loader, gallery controls, icons, music bars, etc (use the Item List alternate color scheme)*

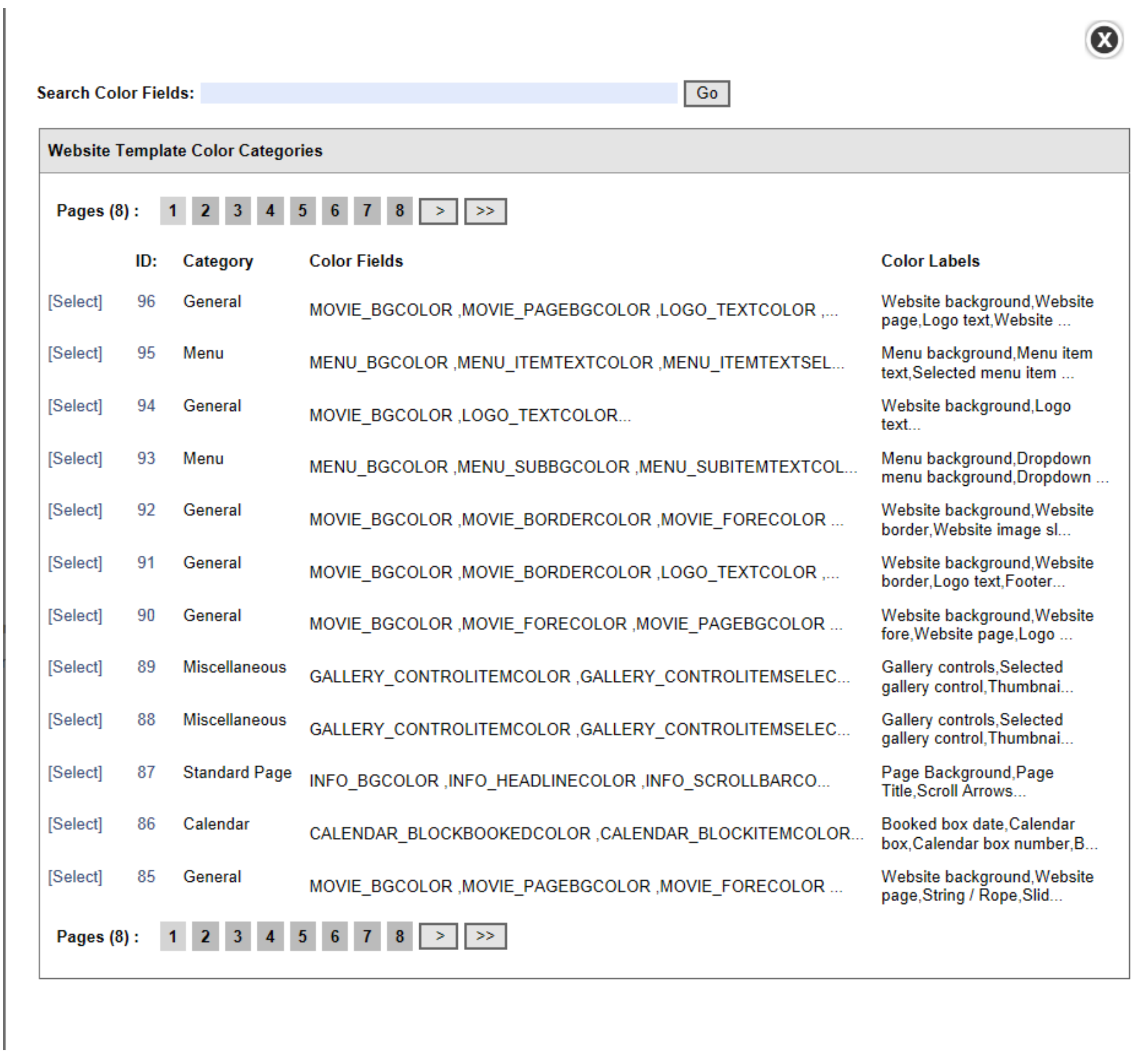
Note: You can use "Snagit" to capture the selection screen so you don't lose the settings.

Click the "**Select**" button next to the category number to view the fields currently selected for that color category. This will show you what fields are associated with that color ID, so that we can verify that all of the fields we need will be present.



If there are fields you don't need, copy the "**Color Fields**" line of text from the **Website Template Color Category** screen to a text editor such as Notepad, and remove the items you don't need. The one you want to use is the second "**Color Fields**" grouping that is a line of text, not a list of text. Make sure you don't have any extra commas, extra or trailing spaces, etc. Next, click on "**View Color Category List**" to bring up the search screen. You can search by the remaining string to find a color ID that already has the exact fields you need by copying that remaining string and pasting it into the "**Search Color Fields**" box.

If there are missing fields, you will need to look for other color IDs that include all the fields you need, and then remove any unnecessary ones to create your search string. To search for an existing color ID, click "**View Color Category List**". This will take you to the color categories list and allow you to see the full list of all existing color IDs.


Search Color Fields: Go

Website Template Color Categories

Pages (8) : 1 2 3 4 5 6 7 8 > >>

	ID:	Category	Color Fields	Color Labels
[Select]	96	General	MOVIE_BGCOLOR ,MOVIE_PAGEBGCOLOR ,LOGO_TEXTCOLOR ,...	Website background,Website page,Logo text,Website ...
[Select]	95	Menu	MENU_BGCOLOR ,MENU_ITEMTEXTCOLOR ,MENU_ITEMTEXTSEL...	Menu background,Menu item text,Selected menu item ...
[Select]	94	General	MOVIE_BGCOLOR ,LOGO_TEXTCOLOR...	Website background,Logo text...
[Select]	93	Menu	MENU_BGCOLOR ,MENU_SUBBGCOLOR ,MENU_SUBITEMTEXTCOL...	Menu background,Dropdown menu background,Dropdown ...
[Select]	92	General	MOVIE_BGCOLOR ,MOVIE_BORDERCOLOR ,MOVIE_FORECOLOR ...	Website background,Website border,Website image sl...
[Select]	91	General	MOVIE_BGCOLOR ,MOVIE_BORDERCOLOR ,LOGO_TEXTCOLOR ,...	Website background,Website border,Logo text,Footer...
[Select]	90	General	MOVIE_BGCOLOR ,MOVIE_FORECOLOR ,MOVIE_PAGEBGCOLOR ...	Website background,Website fore,Website page,Logo ...
[Select]	89	Miscellaneous	GALLERY_CONTROLITEMCOLOR ,GALLERY_CONTROLITEMSELEC...	Gallery controls,Selected gallery control,Thumbnai...
[Select]	88	Miscellaneous	GALLERY_CONTROLITEMCOLOR ,GALLERY_CONTROLITEMSELEC...	Gallery controls,Selected gallery control,Thumbnai...
[Select]	87	Standard Page	INFO_BGCOLOR ,INFO_HEADLINECOLOR ,INFO_SCROLLBARCO...	Page Background,Page Title,Scroll Arrows...
[Select]	86	Calendar	CALENDAR_BLOCKBOOKEDCOLOR ,CALENDAR_BLOCKITEMCOLOR...	Booked box date,Calendar box,Calendar box number,B...
[Select]	85	General	MOVIE_BGCOLOR ,MOVIE_PAGEBGCOLOR ,MOVIE_FORECOLOR ...	Website background,Website page,String / Rope,Slid...

Pages (8) : 1 2 3 4 5 6 7 8 > >>

When searching for an existing color ID, the older ones tend to have more items. Always use <u>ONLY</u> color IDs where all “**Color Fields**” values are in caps. Generally, the next to oldest page is the best place to start. *Do not use* color IDs with any color fields that are in lower case; they do not work.

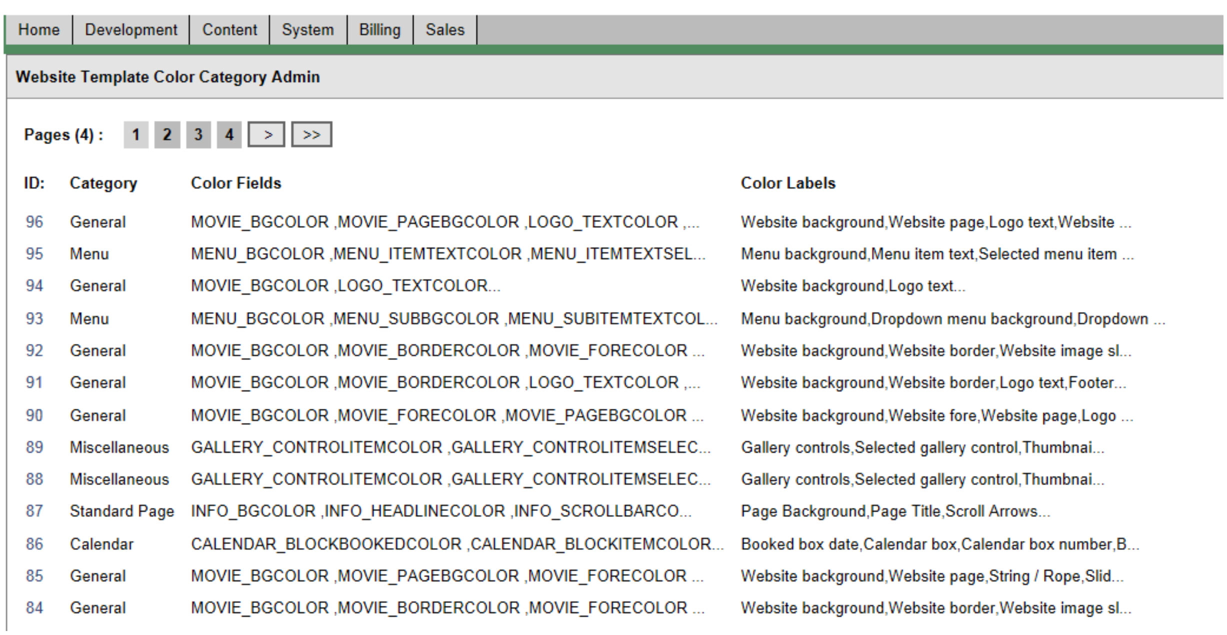
Look for pages that fall in the category you are working with. For example if you are editing **General**, make sure the items you are checking are all in the category **General**. You can click on the number under **ID** to see what fields are active in that color ID.

Once you find a color ID that includes what you need, note down the number. Close the category list window and if necessary, follow the procedure above to create a search string with the exact fields you need and search for an exact match.

Most of the time, there will already be a color ID that matches the fields you need displayed. Note the number, return to the **Website Template Admin** page and re-select the template you are working with. Enter the new **Color ID** in the appropriate **Color ID** box, and click **Update**. Now when you look at the category in your color picker you will only see the fields that correspond with the category ID you chose.

Creating A New Color ID

If your search does not turn up any categories that exactly match the fields you need, you will need to create a new category. Under **System**, you will need to navigate to **Website Template Color Categories**. This will take you to the **Website Template Color Categories Admin** screen.

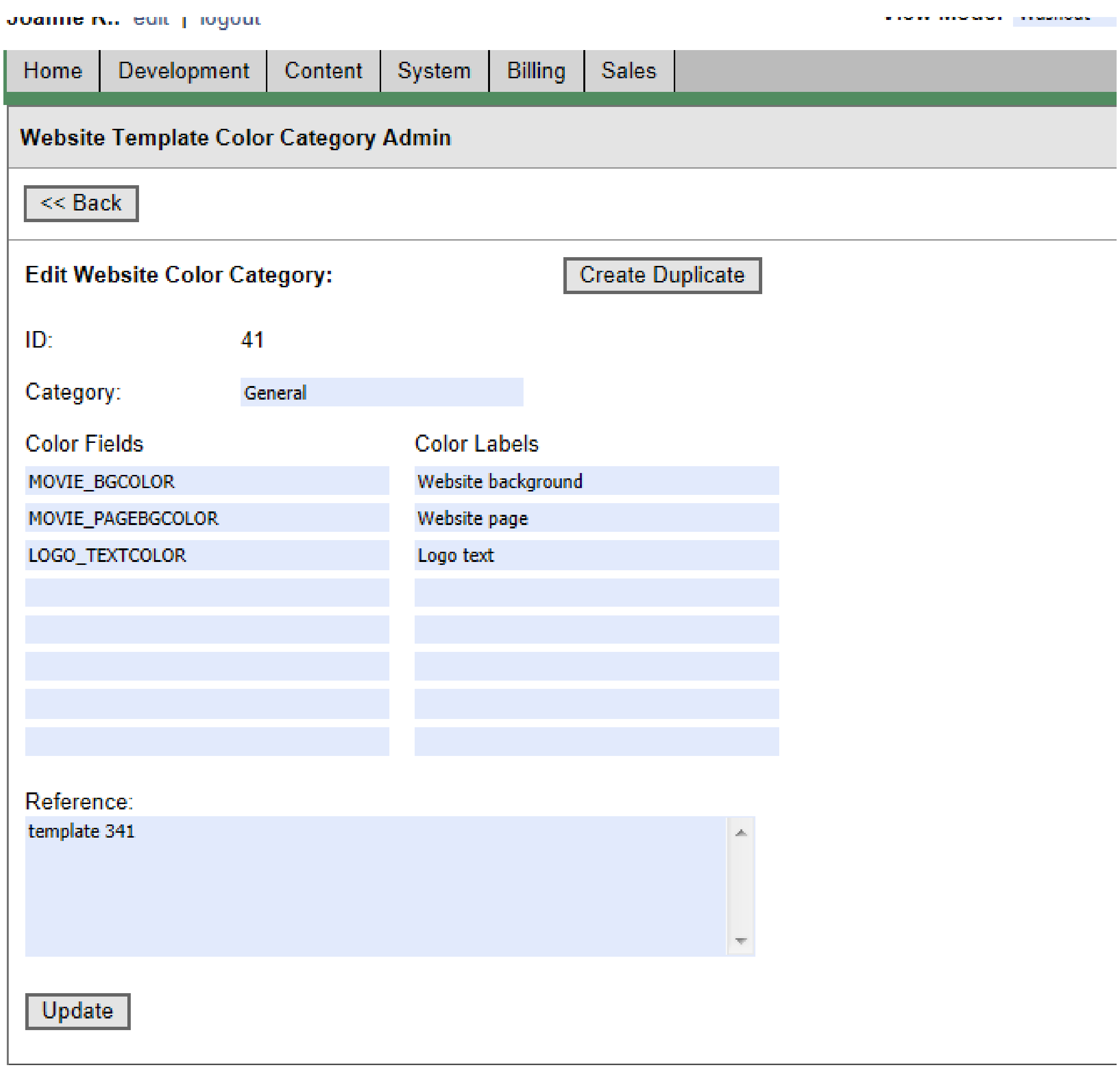

Home | Development | Content | System | Billing | Sales

Website Template Color Category Admin

Pages (4) : 1 2 3 4 > >>

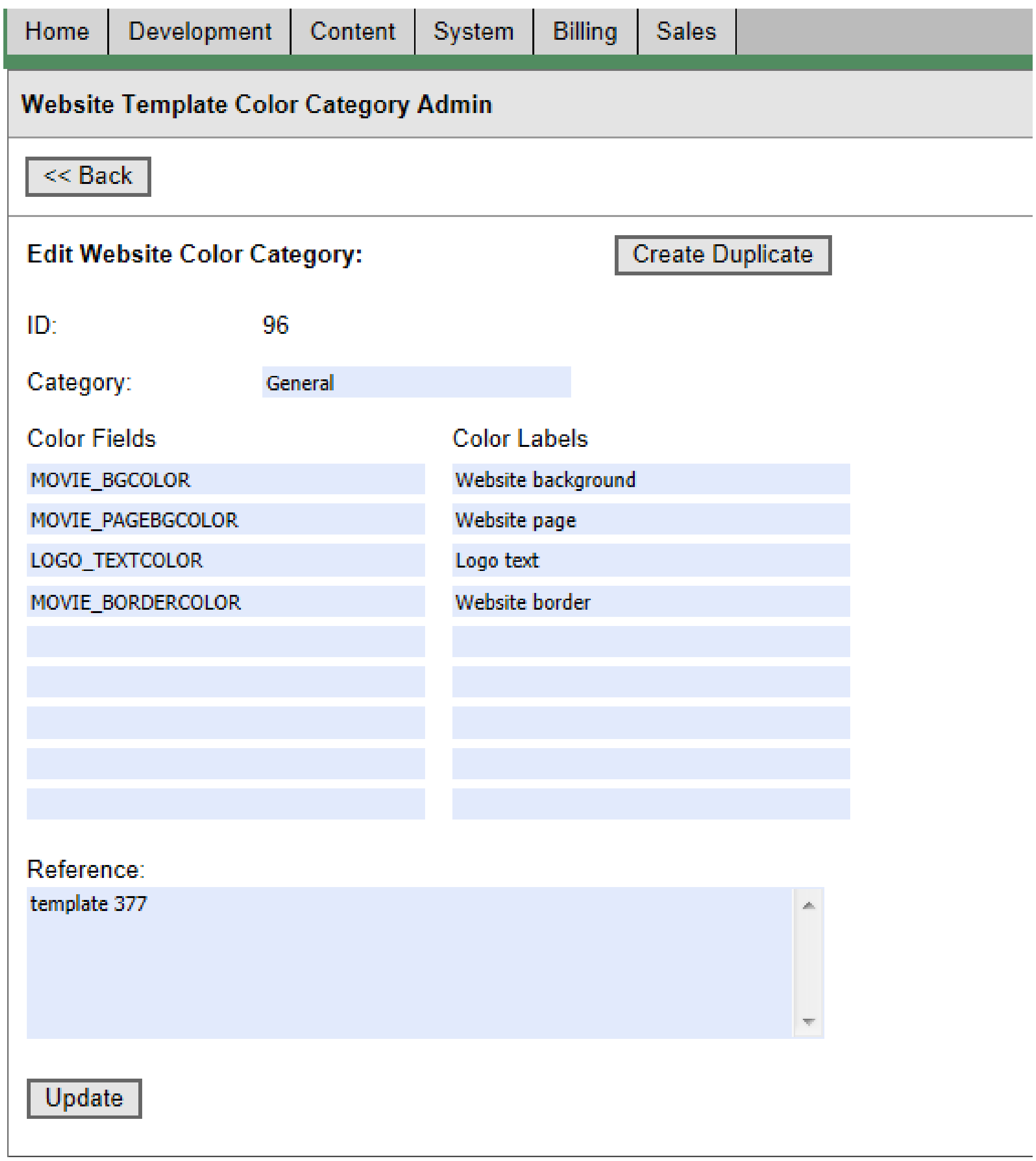
ID:	Category	Color Fields	Color Labels
96	General	MOVIE_BGCOLOR ,MOVIE_PAGEBGCOLOR ,LOGO_TEXTCOLOR ,...	Website background,Website page,Logo text,Website ...
95	Menu	MENU_BGCOLOR ,MENU_ITEMTEXTCOLOR ,MENU_ITEMTEXTSEL...	Menu background,Menu item text,Selected menu item ...
94	General	MOVIE_BGCOLOR ,LOGO_TEXTCOLOR...	Website background,Logo text...
93	Menu	MENU_BGCOLOR ,MENU_SUBBGCOLOR ,MENU_SUBITEMTEXTCOL...	Menu background,Dropdown menu background,Dropdown ...
92	General	MOVIE_BGCOLOR ,MOVIE_BORDERCOLOR ,MOVIE_FORECOLOR ...	Website background,Website border,Website image sl...
91	General	MOVIE_BGCOLOR ,MOVIE_BORDERCOLOR ,LOGO_TEXTCOLOR ,...	Website background,Website border,Logo text,Footer...
90	General	MOVIE_BGCOLOR ,MOVIE_FORECOLOR ,MOVIE_PAGEBGCOLOR ...	Website background,Website fore,Website page,Logo ...
89	Miscellaneous	GALLERY_CONTROLITEMCOLOR ,GALLERY_CONTROLITEMSELEC...	Gallery controls,Selected gallery control,Thumbnai...
88	Miscellaneous	GALLERY_CONTROLITEMCOLOR ,GALLERY_CONTROLITEMSELEC...	Gallery controls,Selected gallery control,Thumbnai...
87	Standard Page	INFO_BGCOLOR ,INFO_HEADLINECOLOR ,INFO_SCROLLBARCO...	Page Background,Page Title,Scroll Arrows...
86	Calendar	CALENDAR_BLOCKBOOKEDCOLOR ,CALENDAR_BLOCKITEMCOLOR...	Booked box date,Calendar box,Calendar box number,B...
85	General	MOVIE_BGCOLOR ,MOVIE_PAGEBGCOLOR ,MOVIE_FORECOLOR ...	Website background,Website page,String / Rope,Slid...
84	General	MOVIE_BGCOLOR ,MOVIE_BORDERCOLOR ,MOVIE_FORECOLOR ...	Website background,Website border,Website image sl...

You should have at least one color ID for the proper category which you would have noted down during your earlier search that is very close to what you need. If not, follow the same guidelines for locating a close match: Look for color IDs in the same category, with as many of the fields you need as possible (or more than you need – we will be editing these in our new color ID), and ensure all color field names are in all caps. You can click on the color ID number to view the fields. Once you have found one that meets your needs, click "**Create Duplicate**". This will create an exact duplicate of the color category, with a new color ID that will be assigned by the system. **Make certain that you are working on the duplicate, and not on the original!** You can now edit the new color category; adding or removing fields as needed to match what you need for the template color picker. You can leave deleted fields blank – the create process automatically drops them.



Note: Ensure that you remain consistent with earlier template color IDs. For example, if earlier templates that use the Website Border color use the field name "MOVIE_BORDERCOLOR" ensure that your new color category also calls Website Border "MOVIE_BORDERCOLOR". In general, it's easier to find a color category that has more than you need and remove the unneeded fields than it is to add additional fields to one that doesn't have everything you need.

Once you have edited your **Color Fields** and **Color Labels**, click on **Update**. This will return you to the top of the list where your new color category will be. You can click on the ID number to verify that your fields are all correct.



Once you have created your new color category, you can return to **Website Template Admin** and update the color category field with your new color ID. Once you save your changes, the correct fields will be reflected in the color picker screen.

Finally, Design will review the template and advise if there are any design changes which will need to be put through the QA process. Once Design has confirmed the template, pre-release testing is complete.

Post-Release

Once Design has confirmed the template, the status of the project itself will be set to "Deployment – Ready". Upon deployment, it will be set to "Live Review". Both QA Specialists should at this point take a few moments to live-test the template. This means going over each of the pages and touching on all of the areas originally examined during the pre-release testing. While it isn't necessary to delve into each individual section as deeply as was done during pre-release testing, you do want to ensure that all areas and pages function correctly and that there were no missed files during deployment. Look at each page, check the transitions, look at icons, music, color schemes, etc.

If you do find issues, check to see if the issue is occurring on the Development servers as well. If the issue is occurring on Dev it may have been missed during testing. You will need to create a Maintenance task to address the issue. If the issue is not occurring on Dev, it may be due to a missed file. If you determine that a file has been missed, you will need to speak to the developer and show them what is occurring. They will need to get with Julio and determine the best way to ensure the file gets deployed properly.

Store/Blog Template Testing

Templates for the Store, BizSites and Blog products are based on HTML5 rather than Flash. While this makes them faster and less prone to issues than Flash, it also means that there can be a greater variance of behaviors between browsers and operating systems. Therefore, it is important to test the templates across all browsers on both OS platforms. It is not uncommon to find issues that only occur on certain browsers or on one OS but not the other.

In general, there will only be one round of testing done on the Store templates. Because Store templates are designed using style sheets to work with the existing structure, most of the issues you discover will tend to be cosmetic rather than functional problems. On the other hand, you will spend more time with Design and Development helping to fine-tune the appearance and behavior of the template.

Pre-Release

Store/Blog template testing utilizes a checklist to help you test a full range of scenarios with template functionality. You should print out a new Store/Blog template checklist at the start of testing as this document is frequently updated. This insures you are working with a checklist that includes new issues to check for as well as known potential issues.

Before you begin testing, take a few moments to review the Planning, Coding and any Design related tasks as well as the QA General Testing task. These tasks often contain valuable information about the design specifications for the template along with any design changes implemented after the example screenshots were created. This will help you to know exactly what you need to see on the template. If you need clarification on any items, check with Design.

Next, fill out the information blanks at the top of the checklist form. The template Number, template Name and Dev name should all be available in the project tasks.

Once you have all the information on the form and you have reviewed the tasks, you are ready to begin testing. It is not necessary to check all the sections in the order they are printed, as long as all sections are checked in the course of testing. Let's do an overview of each section.

General

In this section you will be looking at the overall template design, mainly items that appear on or affect all pages in the template. Review page title alignments, branding, footer, social media icons; check browser dimension changes; ensure template design consistency across all pages. It is particularly important to make sure that borders, lines, item title styles, etc remain consistent across all page types.

Pages

In this section you will review individual pages to ensure all page-specific styles, layouts, settings and design elements are functioning properly. Page types include Marquee, Marquee Plus, Information, Contact, Gallery, Events, Products, Packages, Services, Registrations, Gift Cards, Invoices, External Links, Embedded Links, RSS feeds and Blog pages.

Fonts

The Font settings are fairly extensive and will require very thorough testing as font selections and sizes can have a significant impact on template design and functionality. In each font section you will need to test the full range of available size settings, on several font styles.   Pay particular attention as well to fonts that tend to create display issues such as Engravers MT (very extended), Terfens (low baseline) and Gelder Sans (very thin). A particular note on checking formatting on Macs: Safari and Chrome are known to have issues supporting bold, italic and underline formatting and may not display them on all fonts. This is a known issue with the browsers and per Design is outside of our control.

Color Schemes

Color schemes in Store/Blog templates are managed by color classes. When you review color schemes you will want to ensure that not only do colors respond to changes in the color picker, but also that design elements reference the correct color class. There will usually be notes in the design tasks about color class assignments for various design elements. You can reference a list of which style elements the various color class numbers reference in the Store/Blog HTML Color Classes document.

Control Panel

In the Control Panel section you will be reviewing settings and customizable design elements. You will want to ensure that the template responds correctly to settings and options changes, and that adding customization options such as a background does not create issues with design elements.

When you find an issue with the template, you will need to create a task for Development to address the issue. The task should be assigned to the developer handling that particular template with the status "Development – Assigned". Be as specific as you can: Describe exactly what the problem is and how it appears, list the steps to recreate the problem, provide links to the page where the problem occurred, etc. Make sure to include any relevant screenshots and links that will help the developer see what problem you are experiencing. The more detail you provide, the easier it is for Development to correct the issue.

Once Development completes a task and corrects an issue with the template, they will assign the task back to you with a status of "QA – Returned from Development" along with any informational notes. If there are no notes, that generally indicates the developer was able to correct the issue without need of further clarification. You will then re-test the issue. If the issue has been resolved, note that in the task and set the status to "QA – Completed". If the issue has not been corrected, indicate how you determined that the issue was not corrected in the notes and reassign it back to the developer with the status "Development – Returned from QA". It is important to be very specific in your return notes so that the developer understands how you determined that the issue is still occurring.

Once all of the issues you have found are corrected and all the tasks you have created are confirmed, advise Design. Design will review the template. If everything is correct and there are no design revisions, Design will mark the template as ready for deployment.

It is very common for there to be design revisions. In the event there are design revisions, Design will create a new task (or in some cases, update an existing task if there is one) with information on the desired changes, and assign it to Development. Once the revisions are completed, Development will assign it to QA for testing. After you have confirmed that the revision is correct and there are no issues, you will update the task and assign it to "Planning - Design Review" for Design to review.

Post-Release

Once Design has confirmed the template, the status of the project itself will be set to “Deployment – Ready”. Upon deployment, it will be set to “Live Review”. At this point you will take a few moments to live-test the template. This means going over each of the pages and touching on all of the areas originally examined during the pre-release testing. While it isn’t necessary to delve into each individual section as deeply as was done during pre-release testing, you do want to ensure that all areas and pages function correctly and that there were no missed files during deployment. Look at each page, check layouts, look at icons, color schemes, etc.

If you do find issues, check to see if the issue is occurring on the Development servers as well. If the issue is occurring on Dev it may have been missed during testing. You will need to create a Maintenance task to address the issue. If the issue is not occurring on Dev, it may be due to a missed file. If you determine that a file has been missed, you will need to speak to the developer and show them what is occurring. They will need to get with Julio and determine the best way to ensure the file gets deployed properly.