

This document is an overview of how to adjust the theme of your Prism 3 tenancy and how to work with the theme system. It covers the structure of Prism 3 pages, along with methods and approaches to applying your own styling in a robust manner.

Custom Designs and Default Templates

Although Prism allows you a great deal of control over the look and feel of the application, we recognise that not all organisations have the necessary in-house skills to create or maintain these. We therefore offer two options, a “Default Design” and “Design Services”.

The Default Design can be enabled at tenancy setup and provides a basic set of styles, that can be customised with minimal effort. See the Default Design section for more information.

For information on our Design Services for a full or partial design please contact libraries-sales@capita.co.uk or your account manager for a quotation.

Principles

Prism's core styles work on the following core principles:

- Be neutral
- Provide good defaults
- Allow localisation
- Client-side only

Be neutral – because Prism allows lots of customisation, we try to go with conservative and neutral choices for the core styles.

Provide good defaults – Wherever possible we look to provide the best defaults we can for labelling, structure and design. We also aim for these defaults to be usable and accessible, focusing on how people will use the application.

Allow localisation – Prism allows for both a translated interface (currently Welsh and English) as well as local text and template changes.

Client-side only – because of Prism's hosted nature, we can't allow uploading of server-side components by customers to extend the system. We therefore focus on allowing customisation client-side via HTML, CSS and Javascript within themes.

Theme Structure

At its basic level, a Prism theme consists of the following files:

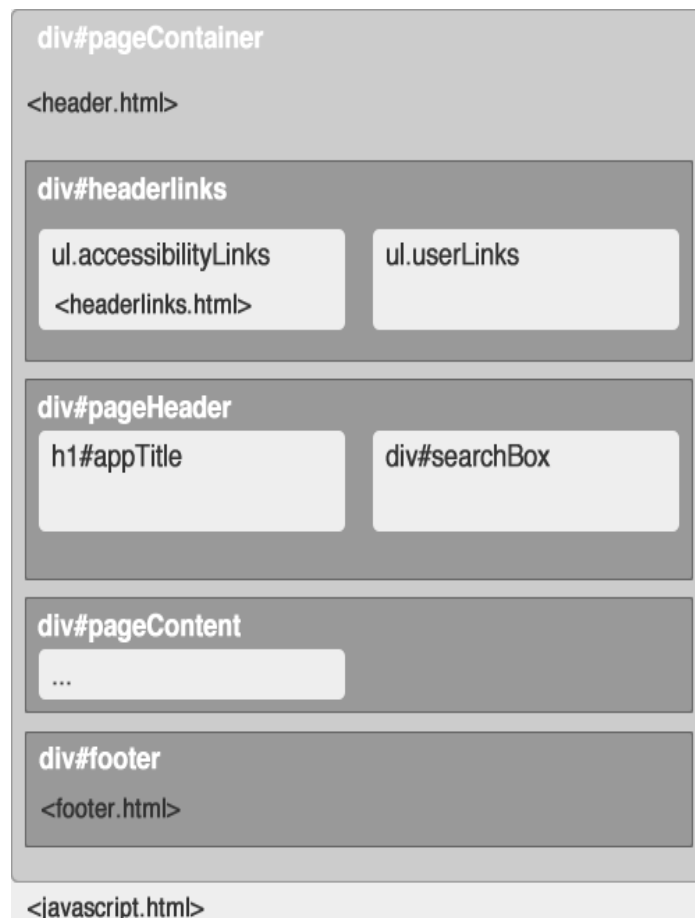
site.css
homepage.html
headerlinks.html
header.html
footer.html
javascript.html
login.html
noresults.html

Each of these files serves a particular purpose within the page's structure. They are added into the page when it is built.

The site.css file is the most important. Prism has a core stylesheet that is always applied and contains the base structure and common design elements (forms, account area and so on). You can add your own styles in the site.css file. This allows you to overwrite existing styles on a case by case basis to match your institutional site.

Note: We are in the process of improving the way stylesheets are handled as part of our mobile stylesheet work. This work will include some extra features for working with legacy Internet Explorer browsers and non-desktop devices. This guide will be updated when that work is complete.

The main page structure of a Prism page is shown below (with extra elements removed). It's important to note that this diagram is structural and not a layout diagram. Due to the nature of HTML and CSS, how a document is structured can be totally different to its display.



Each of these blocks serves a particular purpose within Prism:

#headerlinks	Links at the top of the Prism application panel, relating to core catalogue functions. Its children divide the focus into two types, application links and user links.
ul.accessibilityLinks	This element contains the application-level links. Home, Help, Accessibility and so on. The headerlinks.html fragment allows you to add your own localised links here.
ul.userLinks	This element contains user-related links and form elements. Currently this is Account/Login/Lists information, but we may add more in the future.
div#pageHeader	This block contains information that sits at the top of every page application panel.
h1#appTitle	The application title, by default this is “Prism”, but you can change by asking us to put in a custom language fragment. In the base stylesheet this element is hidden by default.
div#searchBox	The application search box. This is the most used element in Prism, so keep it prominent.
div#pageContent	The main content of the page. This will vary depending on the page of the application, search content is different to homepage content.
div#pageFooter	The footer information of the application. Includes the footer.html fragment and the Powered by Prism notice.

From a design perspective, the important thing to remember is that the header.html and footer.html allow you to “wrap” the Prism application is localised branding.

If you want to include your logo, institutional branding or similar elements, then header.html is the best place to add them.

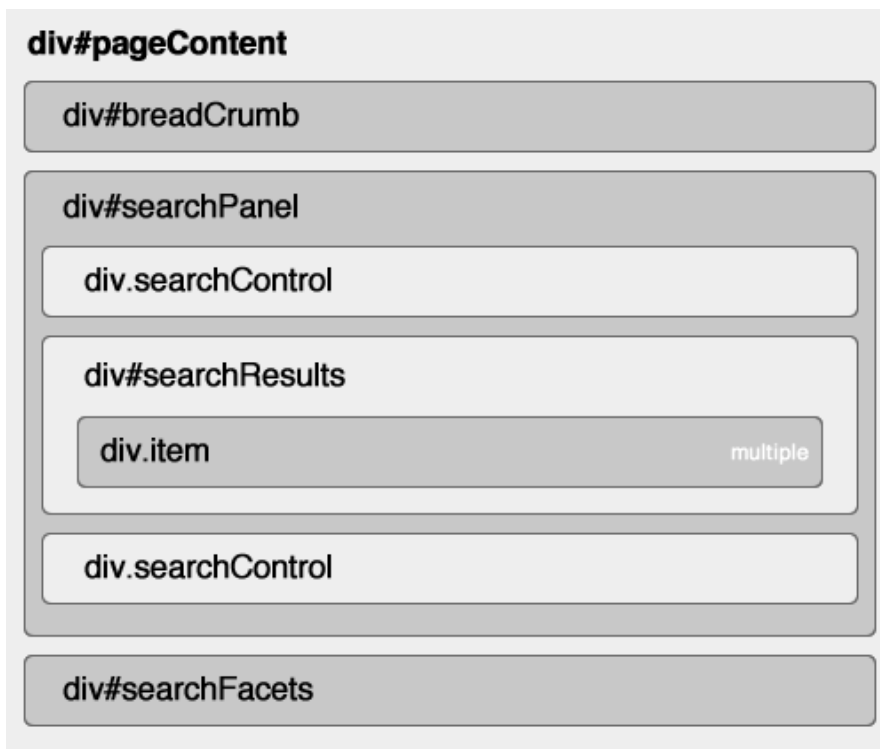
Homepage Page Structure

The homepage of your Prism tenancy differs from other parts of the application, in that the majority of its structure is available for your own content. This gives you a lot of flexibility to style the entry point as you desire by customising homepage.html and site.css.

For examples of how you can use the homepage.html fragment, see the section on Common Patterns.

Search Results Page Structure

The structure of #pageContents for search results looks like this:



The main elements are defined below:

div#breadCrumb	A breadcrumb built from search terms and facets.
div.searchControl	Controls for the search, paging, ordering and so on. Appears at both top and bottom of results.
div#searchResults	The results for a search.
div.item	Each individual item in a search.

div#searchFacets	Facets available for filtering this set of search results. Note, by default this is styled to appear before the results, even though in the document order it is after them.
------------------	--

No Results Fragment

The `noreresults.html` fragment appears on search pages with no results. It is treated a bit differently to others as it allows you to access the search query and insert it into your custom content, for example in a link to another search system such as Copac, the British Library or Google Books.

Taking advantage of this new functionality is easy, just insert the following snippet anywhere in the fragment:

```
{{ query }}
```

If you'd like to provide links to prepared searches on other discovery interfaces you'll need to change that slightly and "url encode" it; this makes sure it's properly formatted for use in a hyperlink:

```
{{ query|url_encode }}
```

So, to add a fragment linking a no results page to searches on Google Books, Copac and the British library, the code would be:

```
<p>You can also try your search for "{{ query }}" on some other catalogues:</p>
```

```
<ul>
```

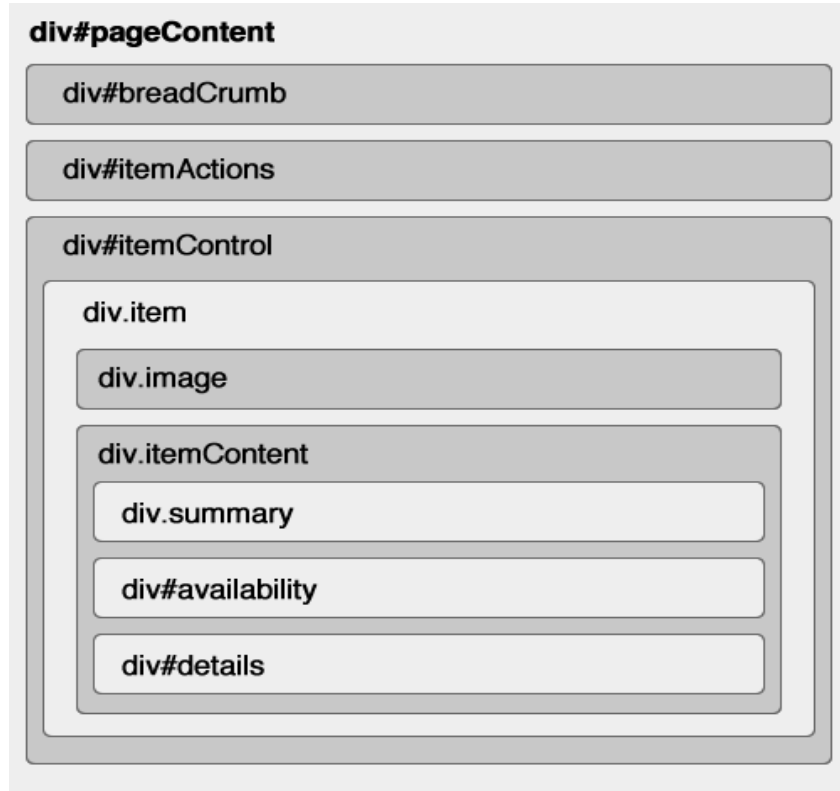
```
  <li><a href="http://copac.ac.uk/search?&any={{ query|url_encode }}">Search for "{{ query }}" on COPAC</a></li>
```

```
  <li><a href="http://explore.bl.uk/primo_library/libweb/action/search.do?mode=Basic&vid=BLVU1&vl(freeText0)={{ query|url_encode }}"&fn=search&tab=local_tab">Search for "{{ query }}" on the British Library OPAC</a></li>
```

```
  <li><a href="http://www.google.com/search?tbm=bks&q={{ query|url_encode }}">Search for "{{ query }}" on Google Books</a></li></ul>
```

Item Page Structure

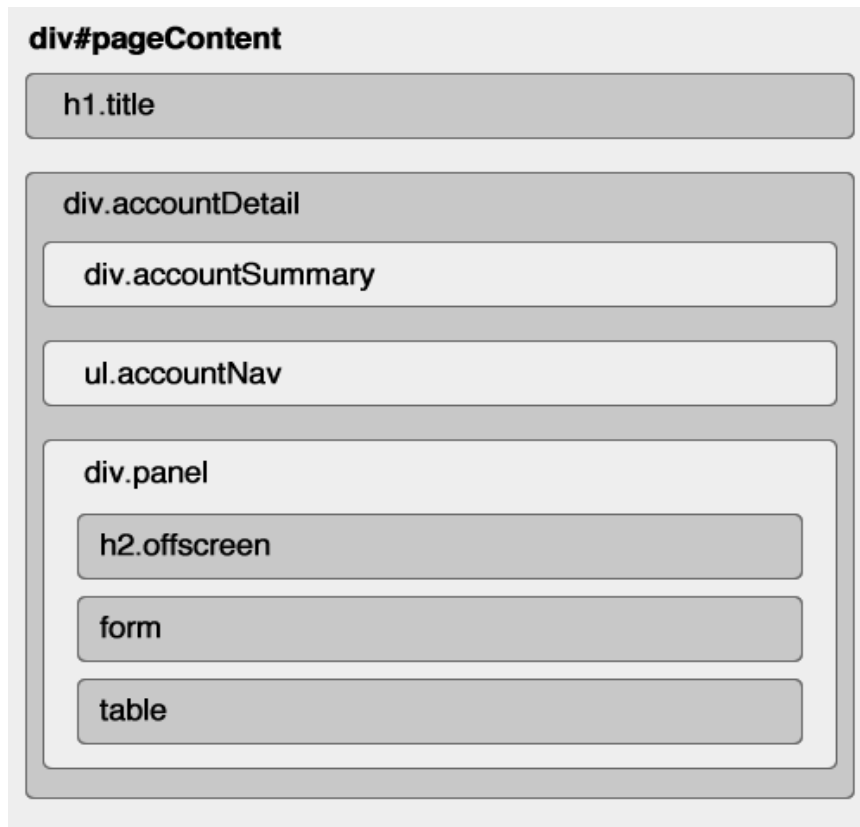
Item pages make up the bulk of Prism display. The overall structure of the page is as follows:



Each of these elements serves a distinct purpose, as detailed in the table below:

div#breadCrumb	Breadcrumb of search arriving at item.
div#itemActions	Containing find more by facets.
div#itemControl	Wrapper information related material.
div.item	An individual item - reused on search results.
div.image	Container for the cover image
div.itemContent	Container for item information.
div.summary	Summary for this item. Contains title, author and key information.
div#availability	Availability of this item at branches.
div#details	More detailed item information, notes and so on.

Account Page Structure



h1.title	Account page title
div.accountDetail	Wrapper for account information
div.accountSummary	Summary of account information statements
ul.accountNav	Tabbed navigation for the various account section. Each li has a class allowing for tabs to be hidden independently.
div.panel	Panel of account information
h2.offscreen	Sub heading for account section
form	Renew all form (renewals page only)
Table	Tabular data for this section of account information.

In addition to the account pages Prism allows you to add a login page fragment, login.html, giving more specific login instructions before the current login form.

Lists Page Structure

ul.listmenu	A menu showing all the lists a user has created
div.list	A container for a list of items
div.export	Functionality relating to the export of lists in citation formats and list deletion.

Future Pages

Prism is an evolving application, and as such we may in future add new pages for new features. Wherever possible, we will try to reuse structural elements of existing pages. We expect to add pages to cover authors, series and specific pages for differing formats in the future.

Page Identifiers

Each page type of the application has a specific identifier, allowing you to target css and javascript based on the page in question. These identifiers are applied to the body element and are shown below:

#index	Home page
#accessibility	Accessibility information page
#searchhelp	Search help page
#renderitem	Item page
#searchaction	Search results page
#login	Login page
#lists	Lists page
#account	Account page

So if you want to target the search box only on the homepage, you could use:

```
#index div#searchBox{
  color:red;
}
```

Multilingual Fragments

Each of the fragments in Prism can also be set as a translated version. So you can have header.html and header.cy-gb.html to switch the header links file if you have multiple languages.

The body of the page also has a class applied to it indicating the language. This way you can show or modify certain elements in particular language contexts. For example, setting an introductory paragraph to display depending on the language:

```
.en-gb p.intro-english{
    display:block;
}
.en-gb p.intro-welsh{
    display:none;
}
```

Interface Text

Text within the Prism interface uses a translations system. This allows us to translate the interface into a variety of languages and also allows you to over-ride default interface text based on your libraries needs.

For example, the default heading for the facets column on a search page is “Refine Your Search“, you might want it changed to “Modify your search”.

To have this change made place a service request. It is helpful if you can specify the page url you are requesting the change for, as well as the current text and the precise requested text change.

Assets

As well as the core fragment files, you can also upload additional assets for use in Prism: images, javascript or other types of file. You can even add additional web pages to your tenancy by uploading html files.

When you upload files into your site's theme they can be accessed at the web address:

```
{tenancy}/assets/-/{filename}
```

All assets also have a versioning identifier (eTag), which you will sometimes see in the url as:

```
{tenancy}/assets/ec7d42e2-e214-4b01-8dfd-3c46e224412f/{filename}
```

This is simply a versioning identifier and can mostly be ignored and replaced with – as shown above.

Scripts

The javascript.html fragment is designed to hold all your javascript and make sure it is run after the page renders. This speeds up the load time for your catalogue, by delaying script loading until after content has rendered.

You may have read that javascript files should be linked from the `<head>` of the document. This is an outdated practice that should be avoided except in a few specific cases (such as webfont loaders).

If you put your scripts in the `javascript.html` fragment, you can take advantage of Prism's local version of jQuery and therefore avoid duplicate downloads of the same library. To access this add the following script before any others:

```
<script type="text/javascript">
//use PRISM jQuery object
$=jQuery=PRISM.jQuery;
</script>
```

Prism Web Addresses

You may wish to link to particular pages from parts of your theme. You can link to any of these pages using the following form:

Homepage: {tenancyname}/

My Account: {tenancyname}/account

Accessibility Page: {tenancyname}/accessibility

Feedback Page: {tenancyname}/feedback

A search: {tenancyname}/items

An item: {tenancyname}/items/{bib_id}

A list: {tenancyname}/lists/{list_id}

{tenancyname}/searchhelp

{tenancyname}/advancedsearch

{tenancyname}/resettenantcache

Simply replace {tenancyname} with the root web address of your tenancy, for example the link to searchhelp on the Broadminster tenancy is:

<http://prism.talis.com/broadminster/searchhelp>

If you have a custom domain name enabled, the url might be:

<http://library.broadminster.ac.uk/searchhelp>

Secure Pages

Prism can work on both http and https, we switch to https when a user logs in so their data for the account is not sent over the open web. This means that your theme needs to work in both http and https. Any assets you upload to the admin console are automatically available in both states.

You may encounter problems with mixed content warnings if you include scripts, images, iframes or css from remote pages. There are several ways to address this.

Assuming you have a secure server to host files on, you can do one of two things:

Detect the protocol via Javascript

With this option, you load assets via Javascript and use a statement like the following to detect whether http or https is used:

```
var protocol = ("https:" == document.location.protocol) ? 'https://' :  
'http://';
```

And insert any elements into the page using Javascript.

Use protocol independent uris.

With this option, you adjust all asset urls to use the following form:

```

```

Note the lack of http: before the //. This will cause the asset will be loaded with the same protocol as the main page.

If you don't have a secure server to host your files on, then it is best to use uploaded assets only. In this case simply refer to any item by a relative url and the base url in Prism will handle whether it is requested securely or not.

Base Urls

All pages in prism have a <base> element set. This means that if a web address you use in a theme is not a full url, the base url of your tenancy is prepended to it.

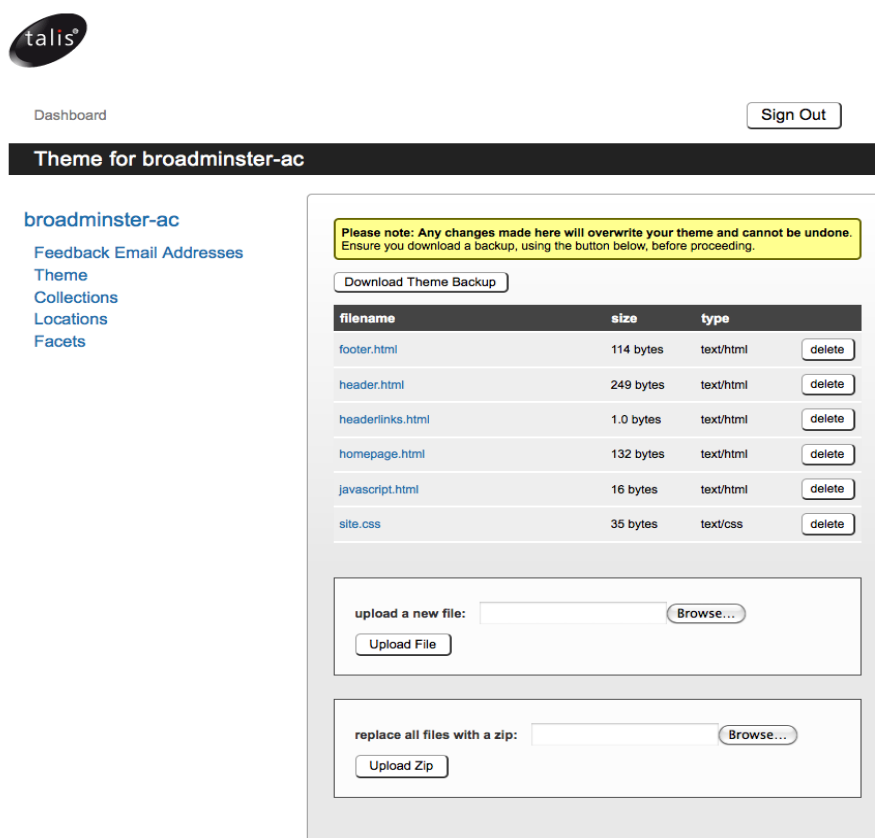
For example if you add a link: Feedback Page the link will go to <http://prism.talis.com/{yourtenancy}/feedback>, and will work whether on http or https.

Working with your Design

Admin System

All Prism tenancies are administrated via the admin.talis.com website. You will be given credentials to access this site and make amendments when your tenancy is set up.

The system allows you to configure a variety of settings for the application, one of these is the theme files.



The theme admin page allows you to upload individual files into the root folder, upload a single zip file containing a whole theme to update or download a backup of your theme.

If you do upload a whole theme, be careful to keep the file size down. Anything you upload will have to be downloaded by your users! Uploading a theme can take some time if there are many files or big files. It's easy to forget to remove script documentation or examples, which make the whole process take longer than needed. It is therefore best to only add what you're using to the theme.

Note: You should always download a copy of your theme to use when making changes and add changes to those files rather than working from an older copy. With the nature of the tenancy setup, it is possible that somebody has made changes to your theme since your last edit, so the best way to be sure you are using the up to date version is to re-download.

Resetting a theme

In order to speed up the display of Prism, assets and styles relating to your theme are aggressively cached in the application. This means when a user accesses your prism tenancy the page loads quickly.

When you load in new elements of a theme, you may find they don't display right away as the old ones are still in the application cache. To clear these out, you need to request <http://prism.talis.com/{tenancyname}/resettenantcache> to clear the cached versions.

It's worth noting that resetting the cache may make the very next request for your catalogue slower as the cache is refreshed, so it's best to avoid doing this too much during peak hours.

Using sandbox

All Prism tenancies are set up so that you have two sites to work with. Your main live site that you point the public at, and a sandbox site where you can try out new designs and features before making them live. When you log in to the admin site, both your tenancies will be visible. Your sandbox will be located at:

<http://prism.talis.com/{tenancyname}-sandbox/>

When you test design changes, take a copy of your live theme by downloading it from admin.talis.com. This ensures you have the latest version of the live theme, including any modifications made by other people. Make any modification you require and upload to the sandbox tenancy to test them.

Using Demo

Prism is an evolving piece of software. When a new version becomes available, we place it on a demo web address so that you can test any new functionality and check your themes work with the new version.

This web address is: <http://prism.talis.com/demo/{tenancyname}>

You can use the /demo/ url for both sandbox and live tenancies, allowing you to preview changes to themes for the new release. This testing time period is usually two weeks, though may be longer if issues are discovered.

Note: If you are using Prism on your own hostname then demo is not available at {ownurl}/demo and you will have to use prism.talis.com/demo/{tenancyname} to preview changes.

Building Flexible Designs & Future Proofing

Prism is an evolving product. New features are added and old features are modified and improved on a regular basis. So how to design a theme that is flexible to this?

Avoid over-styling the Prism application panels

In general, if we add new features, they will appear in the main application panel or the userlinks area. This means to avoid issues when we upgrade the system, the easiest solution is to only apply minimal changes to colour and layout to these core areas. Focus your styling on areas like header.html.

Be careful of fixed widths and positioning

If you set widths on elements or set background images that are fixed width, you may run into problems when new elements are added or removed (or simply when a user resizes the browser text).

Absolute or relative positioning based on fixed widths, rather than proportional measurements like % or ems can also cause issues when new elements are added.

Be specific in your CSS selectors

The more specific your css styles are, the less likely they are to break. This is especially true where we reuse generic marker classes like last, first, more, author or formElement.

For example. We have a class .formElement that is added to form element groups. You could target all form elements using:

```
.formElement{  
color:blue;  
}
```

If you wanted to target these on the feedback page use:

```
#feedback .formElement{
    color:blue;
}
```

if you want to target those on the item pages too use:

```
#feedback .formElement, #renderitem .formElement{
    color:blue;
}
```

Prefix your classes and ids

If you are adding classes or ids for CSS or Javascript hooks, it's best to add a local prefix to avoid potential collisions.

For example, you might decide to create a class `.panel` that conflicts with Prism's `.panel` class. By making your class `.mylibrary-panel` you can ensure future releases won't ever cause conflicts.

Common Patterns

We often get asked about how to implement certain features in a theme. Some of the common requests are discussed below.

A Typical Homepage

It's common for libraries to want two panels of content on the homepage, one for new or interesting books, one for news and attendant content. The following code will give you a basic set of panels to work with.

Add the following your homepage.html file:

```
<div id="hp-primary-content" class="hp-panel hp-wide">
  <h1>Books we Love</h1>
  <div class="hp-items">
    <!--page item blocks from search here -->
  </div>
</div>

<div id="hp-secondary-content" class="hp-panel hp-short">
  <h2>Library News</h2>
  <ul class="hp-news">
    <li><a href="http://">News Item Link</a></li>
    <li><a href="http://">News Item Link 2</a></li>
  </ul>
</div>
```

Then add the following to `site.css`:

```
#index .hp-panel{
    float: left;
    display: inline;
    padding: 1em 1%;
    border: 1px solid #eee;
}
#index .hp-short{
    width: 32%;
    padding-top: 1.5em;
}
#index .hp-wide{
    width:60%;
    margin-right: 2%;
}

#index #pageContent{
    overflow: hidden; /*clear floated elements*/
}
```

Moving the Homepage Search box

Often libraries want the focus of the homepage to be the catalogue search, and so want the search box “moved” from its normal position in the header.

The easiest way to achieve this is to add your own search box code in the `homepage.html` fragment and then hide the normal search box on the homepage.

```
<form id="homeSearchForm" method="get"
action="http://prism.talis.com/broadminster/items">
    <label for="homeSearch">Search the library catalogue</label>
    <input id="homeSearch" type="text" value="" maxlength="1000"
name="query" />
    <input type="submit" value="Search" />
    <a href="http://prism.talis.com/broadminster/advancedsearch">More search
options</a>
</form>
```

You can then style the two searchboxes with different looks and feels, by targeting using the `#index` selector to specify the homepage only. For example you could add the following to your `site.css`:

```
#index #searchForm{
    display:none;
}
#index #homeSearchForm{
    padding:1em; /*extra padding */
}
```

Linking to New Stock Searches

This feature requires the new MarcGrab software. Assuming you want to show the latest additions to the collection called "Main Catalogue", enter `collection:"Main Catalogue"` in the search box and click search.

Copy the url from the address bar and append the following to it:

```
&sort=shelveddate%3Ad&limit=10
```

This modifies the search to return in order of shelved date and show only the last 10.

More detail at <http://blogs.talis.com/prism/2010/04/15/new-stockprepared-searches/>

Javascript Extensions

Using javascript you can add extra functionality to your tenancy. We recommend using jQuery and Juice for extending Prism, two open source javascript libraries. JQuery is a Javascript library that solves the cross-browser headaches of javascript and has a variety of extensions that can be used to enhance the way information is displayed.

Juice is a jQuery extension focused on enhancing library catalogues. It uses jQuery as a base and allows you to add extensions with a range of functionality, from cross searching to embedding maps. We recommend you use at least version 0.7 of Juice, as this is the latest, fastest and most stable version.

More detail of how to add Juice to your Prism tenancy can be found online at: <http://juice-project.org/PrismJuice>

Testing Your Design

Once you have your theme finished and uploaded to your sandbox tenancy, it's best to run through a few checks.

Browser Versions

If you are not a practiced web designer, it's easy to fall into the assumption that if a design looks right on your particular setup, it works for everybody else.

For example, some institutions use the Internet Explorer 6 browser, but that browser accounts for only around 10% of Prism's traffic and less than that globally. If you only test on that older browser, with its many bugs, you may find your theme breaks on more modern systems.

When you make big changes to a design, it's best to test it with the current and previous version of popular browsers as these make up the majority of your users.

Accessibility

By default we make Prism as accessible as we possibly can, aiming for WCAG 2 level 2. However, when you make modifications to themes it is possible through adding javascript or changing CSS to hinder accessibility.

Common issues to look for:

- Making colour contrast too low in your styles
- Missing textual alternatives on forms or images
- Design breaking when resizing text due to using pixel widths
- Not providing keyboard accessibility in extensions
- Adding invalid HTML to fragments

Useful Tools

Firebug - <http://getfirebug.com/>

Provides a host of debugging tools to allow you to see why an element is styled in a particular way, detect javascript problems or missing downloads.

IE Developer Tools - www.microsoft.com/download/en/details.aspx?id=18359

Similar functionality but for the Internet Explorer browser.

Firefox Accessibility Toolbar - <https://addons.mozilla.org/en-US/firefox/addon/juicy-studio-accessibility-too/>

Tools including a colour contrast analyser.

Useful References

These guidelines can't be a guide to building with CSS, Javascript and HTML. Fortunately people have already written on these topics extensively, we highly recommend the following sources:

Books

Designing with Web Standards, Jeffrey Zeldman

Bulletproof Web Design, Dan Cederholm

DOM Scripting, Jeremy Keith

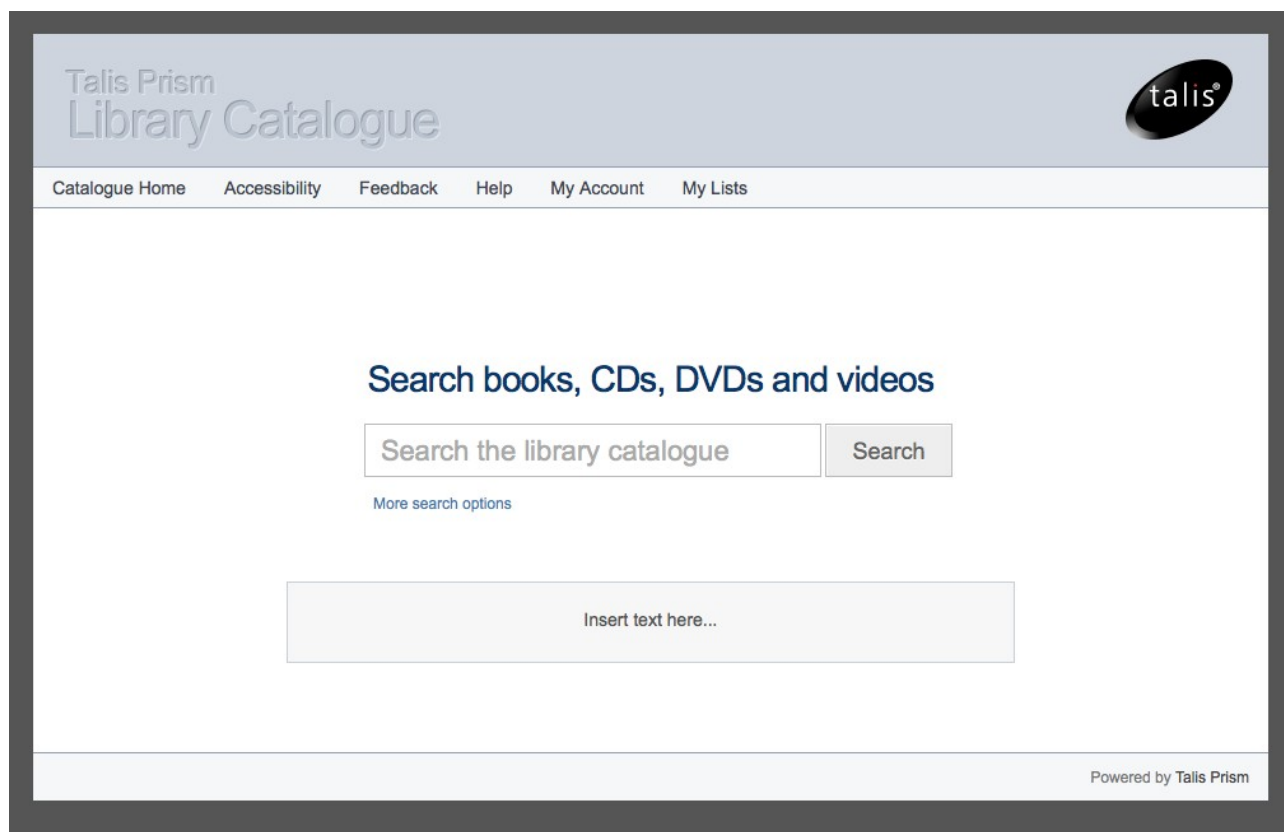
Web Accessibility: Web Standards and Regulatory Compliance, Bruce Lawson et al

Websites

We keep a wiki page at http://developer.talis.com/Prism:Main_Page where we detail many of the features and patterns for developing Prism themes as well as other topics like the Linked Data API.

The Default Design

The default design is a neutrally coloured standard theme. It's designed to be functional and to allow customisation with a minimal overhead. While it lacks the impact of a custom design, it provides a solid, usable foundation to work on.



Design Restrictions

In order to preserve the consistency and usability of the application, the following restrictions should be observed:

1. Search box

Removal of the search box, search button, more search options link from any of the pages is strictly prohibited.

2. Header fragment hyperlinks

The following header fragment links must be present on all screens:

- A link to the catalogue home page
- Accessibility Help
- Feedback
- My Account
- My Lists

Please do not remove links or duplicate the functionality with your own versions in the html fragments, this may result in functionality becoming unavailable to your users. The exception to this is replacement search boxes within homepage.html.

3. Powered by Prism link

The Prism hyperlink must appear in the footer area on all screens and hyperlink to www.talis.com/prism.

Commenting Changes

Occasionally our support staff will make changes to your tenancy theme on your request. When we do so, we'll leave a comment in place explaining the change with the date and initials.

Width: 30px /*Fixed widths - 12/03/09 MM*/

Feedback

These design guidelines will change as Prism evolves. If you have any feedback or suggestions for further coverage, please contact us at libraries-sales@capita.co.uk