The Netscape 8.0 Themes Toolkit

Advice and Resources for Working with Themes

By Kurt Cagle

and Raj Paul

The Netscape 8.0 Themes Toolkit: Advice and Resources for Working with Themes

By Kurt Cagle
and Raj Paul

Abstract

The XML Theme Kit is provided by the Netscape Corporation in order to make it easier to create themes and skins for the Netscape 8.0 browser. It is provided as is with no warranty, and Netscape assumes no liability in the use of this toolkit.

Table of Contents

List of Figures

List of Tables

List of Examples

Chapter 1. Understanding Themes

We all like to add our own "personal" touch to the world around us. Whether one's house or one's software, the ability to change the way that our world looks and acts is one of the more desired "features" that people want. The Netscape browser, based upon Mozilla Firefox, is built with this capability in mind, letting you both load in new themes from artists and designers and making it possible for you to create themes of your own.

The Netscape 8.0 Theme Kit consists of both this document, which describes both how to use and how to create themes, and a sample theme template to be used to generate a new theme that can be used personally or that can be submitted to share with others. It is expressly for the Netscape 8.0+ series of browsers.

Getting Under the Skin

Open up a typical application, and it is likely that the program looks a great deal like every other application out there - the title bar is probably blue, maybe with a gradient thrown in for good measure so that it goes from dark blue to light blue ... the height of interface fashion, to be sure. The buttons and menubars all likely have a specific light gray background, selected to be as neutral as possible, and the fonts are likely some sans-serif variation in basic black.

Given this, it's not surprising that people occasionally long to be able to dress up their applications, to make them just a little bit less .. um, corporate (or perhaps a little bit more, if your desires *really* lean that way). Themes or skins emerged as a way of customizing applications so that they were perhaps not quite so vanilla, and indeed were better able to reflect the personality of the user of those applications. Sometimes these changes are fairly minor - altering the default font or color sets, for instance - while in other situations (such as with audio or media players, which seem to have taken this phenomenon to a new level) the customization extended to shape and even component functionality.

The Netscape 8.0 browser, having been built on top of the Gecko framework used by Mozilla, offers a surprisingly robust degree of customization with respect to its themes. While certain issues prevent the creation of non-rectangular windows (at least in the current iteration), the Netscape 8.0 browser is otherwise incredibly extensible - letting you alter the background images of the title bar and status bars, replacing stodgy old rectangular buttons with much more dynamically shaped buttons, altering colors, fonts, and positions, and so forth.

What's more, because Netscape *does* use the Mozilla XUL engine, you can take advantage of the Extensible Binding Language (XBL) in order to create far more sophisticated behaviors in the components that you add. While in some respects this blurs the boundary between creating "pretty" skins and actually developing applications, this doesn't mean you can't (or shouldn't) do it.

This article covers the process of creating Netscape 8.0 themes, from inception through bundling and packaging to updating. It assumes that you've had some experience working with XML and CSS, but for the most part the underlying lesson here should be that you can create very rich, sophisticated themes with comparatively little work.

Finding New Themes

You can easily add new themes from the Internet and load it into your browser. Netscape maintains a specific site, called the Theme Park [http://browser.netscape.com/ns8/community/themes.jsp], at http://browser.netscape.com/ns8/community/themes.jsp. This site contains specific themes for use within Netscape itself - moreover, because of the potential security risks associated with any themes, the themes displayed here have been checked to insure that there is no malicious code that can affect your browser.

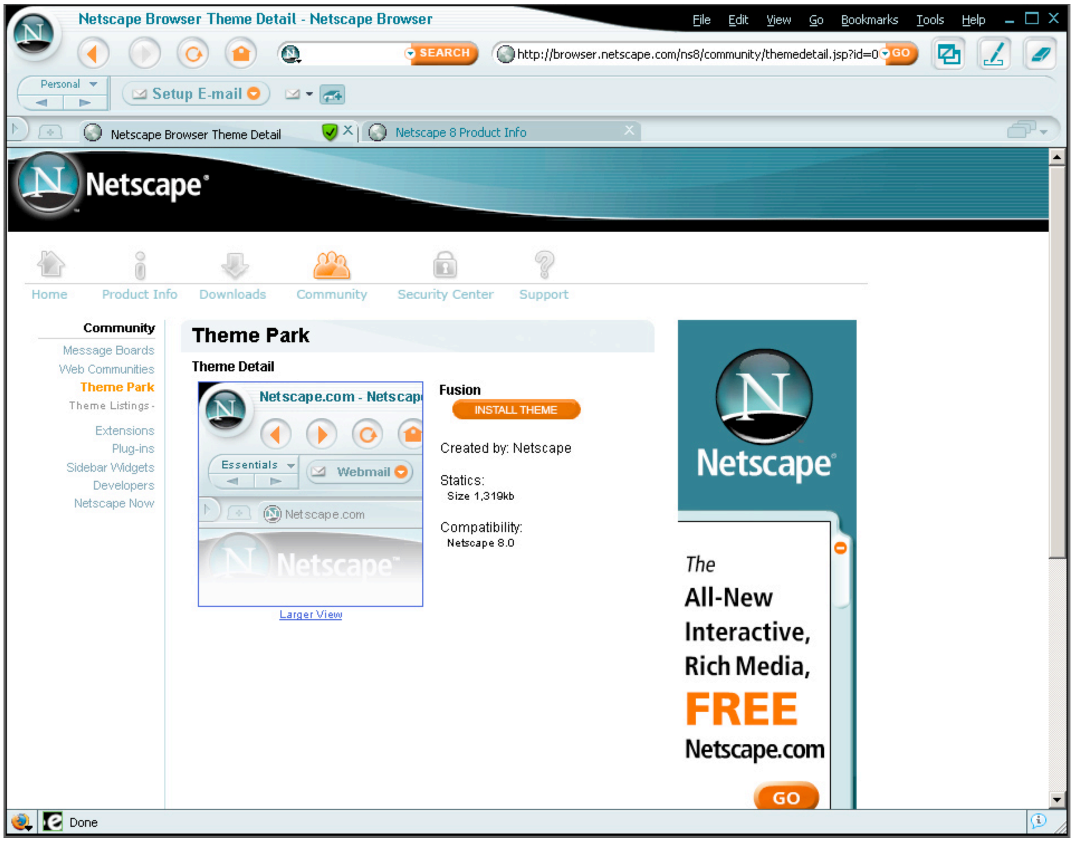
Warning

As a general security measure, you should only download themes from officially sanctioned Netscape sites. This will insure that the themes you are loading have been inspected by Netscape and determined to be clean. Unless you are sure about the integrity of the site, you should *never* download themes from unauthorized sites, as it increases the risk of viruses, spyware or other malware.

To download a theme:

1. Go to the Theme Park at http://browser.netscape.com/ns8/community/themes.jsp.
2. Preview the themes until you like one you like (there are usually screenshots that you can examine if you don't want to download the theme.
3. Click on the Install Theme button (cf. Figure 1.1, “Selecting a Theme from the Netscape Theme Park ”).

Figure 1.1. Selecting a Theme from the Netscape Theme Park



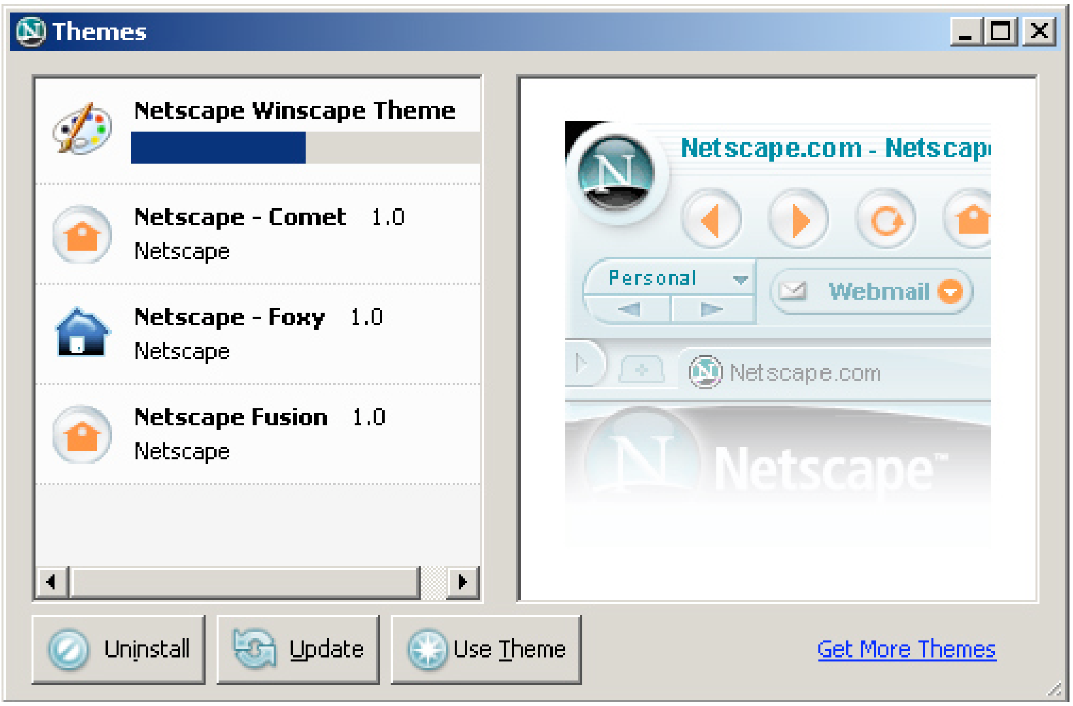
Created by: Netscape

Statics:
Size 1,319kb

Compatibility:
Netscape 8.0

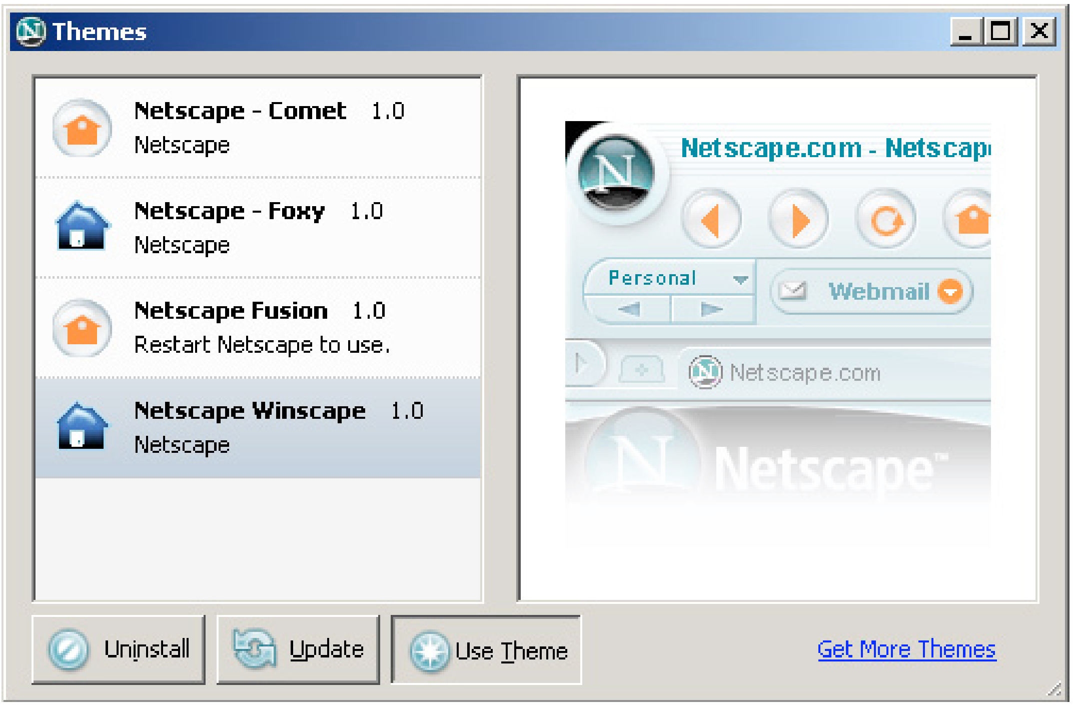
4. This will bring up a dialog box asking whether you wish to install the theme. Press the OK button to load the theme, or Cancel to stop the operation and return to the browser. Once selected, the theme will automatically load into your theme manager dialog (cf., Figure 1.2, “Downloading a Theme from the Netscape Theme Park ”)

Figure 1.2. Downloading a Theme from the Netscape Theme Park



5. After you download a theme, the theme is not yet selected. To use a particular theme, select the theme name on the upper-left hand side of the Theme Manager, then press the "Use Theme" button in order to tell the browser to switch to the new theme (Figure 1.3, “Selecting a Theme to Use”).

Figure 1.3. Selecting a Theme to Use



Troubleshooting Download Problems

Periodically you may encounter problems when downloading themes. Here are a few suggestions about what to do handle that case if it happens to you.

- **Wrong Browser.** While Netscape is built upon Firefox, Netscape and Firefox have different identifiers. As the install.rdf used by themes relies upon the identifier to ascertain whether the installing application is the correct one for the code, this means that you can't run Firefox-based themes.
- **Wrong Version.** The install.rdf also contains information about which versions of the browser are supported. Failing to include both an upper and lower limit on browser versions within the range for the current browser will launch a message indicating that the component won't work with the browser in question. Either look for a more up to date theme or upgrade your browser as appropriate.
- **Odd XML In Browser Window.** Periodically you may find with a theme that you get an odd block of XML or error code appearing within the browser itself. This usually is a symptom that either the XML (most typically the XUL in more elaborate themes) is malformed, or it indicates that there is a collision of critical ids or similar features which renders the them install incompatible with the browser or (more likely) some extension. The best solution in this case is to remove the offending theme, using the uninstall option of the Theme Manager Dialog.

More information about themes and troubleshooting can be found at the Theme Park pages at http://browser.netscape.com/ns8/community/themes.jsp .

Uploading Themes

While it is possible to host themes from your local server (as will be discussed in the next chapter), in general it is far better if you have a theme that you wish to share to upload it to the world via the Netscape Theme Park (
<authornote>URL to be determined</authornote>
). To upload your theme, fill out the form, attaching the file using the *Theme File:* input field and submit the file.

There are, however, a number of considerations that you should be aware of before submitting any theme:

- **Do Not Violate Copyright.** If you do not own the copyright to images (or entities) depicted within your theme, do not upload a theme containing them. No matter how much you want to do the Darth Vader theme, if you upload it, it won't be posted unless you are working for George Lucas.
- **No External References.** All images and related resources must be contained strictly within the theme's JAR file - there can be no references to external images or other resources sitting on a given server.
- **Identify All Bindings.** If your theme makes use of bindings (XBLs), scripts, or other executable, please note them in the uploaded document identify them and why they are included.

Chapter 2. Working with the Master.css Rules

Netscape has been defined to be customized. To do this requires the direct manipulation of a special CSS file called master.css which should be contained in each theme (it is invoked, however, by the browser itself). You can edit this file in your own theme version in order to change the styling of the browser itself.

Understanding the Master.CSS file

As mentioned in the previous section, all Netscape theme jars should contain a copy of master.css. This file is defined originally to have no direct effect upon the browser - most of the containers that the CSS lists are previously defined as part of other CSS files. However, master.css is generally invoked last, which means that any selector found within the master.css document will automatically overlay what had been defined previously, making it a known entry point for modifications that web authors can use to style the browser. Code Listing Example 2.1, “Master.css Base File” contains a listing of the "default" master.css which comes with the toolkit.

Example 2.1. Master.css Base File

```
/*************************************************************************\
 * master.css - use this file to quickly theme NS8's most visible components
 * Uncomment styles and appearances of elements you wish to modify. Refer
 * to the graphic SDK in the Photoshop and ImageReady Files and overwrite
 * graphical elements to change them.
 *
 * We recommend using a CSS editor to edit this file such as Rapid CSS
 * which has a color picker to generate the hex code for colors. Please
 * refer the theming SDK document for a How-to for this document.
 *
 *************************************************************************/
@namespace url("http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul");
/* ====================================================================
  Global application changes
  Use this section to change:
     - The main window's background color  (background-color: color)
     - The main window's font (font: font)
     - The main window's font color (color: color)
          - The main window's background
     If you would like to keep the application looking like a standard
     windows application then comment out the attributes mentioned above
     for window.
  ======================================================================= */
  /* Change the background color of the application
      explain what each one is*/
  window,
  #winInspectorMain,
  #extensionsManager,
  #downloadManager,
  #CustomizeToolbarWindow,
  #bookmark-window,
  page,
  dialog,
  wizard {
      /*background-color: black !important;*/
      /*color: white !important;*/
      /*font: arial !important;*/
                /*background-image: url("path/to/image.png") !important;*/
                /*background-repeat: no-repeat !important;*/
  }
  /* Change the font color and the font of the tabbrowser tab*/
  tabbrowser tab {
      /*color: pink !important; */
      /*font: arial !important;*/
  }
/* ======================================================================
   Main menubar and menubar items. Uncomment the background-color and color
   to change the mouse over properties of the menu and the font color
   respectively.
  ====================================================================== */
  #main-menubar > menupopup > menu[_moz-menuactive="true"],
  #main-menubar menupopup > menuitem[_moz-menuactive="true"],
  #main-menubar popup > menu[_moz-menuactive="true"],
  #main-menubar popup > menuitem[_moz-menuactive="true"],
  menupopup > menu:hover,
  #main-menubar popup > menu:hover,
  #main-menubar menupopup > menuitem:hover,
  #main-menubar popup > menuitem:hover
   {
          /*background-color: #371C14 !important;*/
        /*color: white !important;*/
  }

   /* :::: The font properties for the application menubar::: */
  #titlebar menubar > menu {
          /*color: Lime;*/
        /*font-family: Arial;*/
        /*font-size: 10pt;*/
        /*font-style: italic;*/
  }

/* ======================================================================
  Window top
  Use this section to change:
     - The appearance of top of the navigator window.
     #top-box: styles to control appearance of the top of the browser
               window.
         To have a solid color, comment the background-image line of top-box
         and supply a non-transparent background-color,
          i.e. background-color: red;
     #titlebar #titlebar-separator: image that separates the title of the
     document and the menubar (if toggled to the right).
      To have no separator
     uncomment the line background-image: none !important;

          #titlebar #titlebar-title: Font, font color and Text Decoration
          of browser
          window title.
  ====================================================================== */
  #topbox {
          /*background-color: #371C14 !important;*/
          /*background-image: none !important;*/
  }

  toolbox {
          /*background-image: none !important;*/
  }
  #titlebar {
          /*background-color: transparent;*/
  }
  #titlebar #titlebar-separator {
          /*width: 153px;*/
          /*background-image: none !important;*/
  }

  #titlebar .titlebar-menu-stack {
          /*background-color: #371C14 !important;*/
  }
  #titlebar #titlebar-title {
        /*font-family: Arial !important;*/
        /*font-weight: bold !important;*/
        /*font-size: 12pt !important;*/
        /*color: #FFFFFF !important;*/
        }

/* ====================================================================
  Statusbar
    Uncommment the background-color style or  background-image style
    to modify the appearance of the statusbar.
  ======================================================================= */
  statusbar {
          /*background-color: #FFCF94 !important;*/
        /*background-image: url("backgrounds/statusbar-bg.gif") !important;*/
        /*background-repeat: repeat-x !important;*/
  }
/* ====================================================================
  Sidebar
    Use this section to modify the appearance of the sidebar.
  ======================================================================= */

  /* :::: The background color and the top border of the sidebar::: */
  #sidebar-box {
          /* border-top: 1px solid #344c52 !important;*/
          /*background-color: green !important;*/
        /*background-image: url("path/to/image.png") !important;*/
        /*background-repeat: no-repeat !important;*/
  }
  /*::: .sidebar-header-text: sidebar header - (My Sidebar Text)::::: */
  .sidebar-header-text {
          /*color: black !important;*/
          /*font-weight: bold !important;*/
  }
  .sidebarheader-main {
          /*border-top: 1px solid white !important;*/
  }
  /* ::::Sidebar picker button::: */
  /* Default state of the panel picker button */
  #sidebar-panel-picker {
          /*color: green !important;*/
          /*border: 1px solid transparent !important;*/
  }
  /* Hover state of the panel picker button */
  #sidebar-panel-picker:hover {
          /*border: 1px outset #B1BDC9 !important;*/
  }
  /* Open state of the panel picker button... after the user
     clicks the button and the menu is open */
  #sidebar-panel-picker[open="true"] {
          /*border-width: 1px !important;*/
          /*border-style: inset !important;*/
  }
```

The master.css file contains a number of useful "containers" that can control the look and feel of the browser. These containers are also contained elsewhere in the browser application, so the effect of any changes in master.css will only be to override the existing behavior for these items. The table Table 2.1, “Description of the Selectors in Master.css” contains a summary of the classes that are affected by this document, describing what they do and providing hints about how to use them.

Table 2.1. Description of the Selectors in Master.css

Rule Name	**Rule Selector**	**Description**
Window	window	This is the general window used by the browser itself. CSS properties set here will affect everything within the browser.
Window Inspector	#winInspectorMain	This controls the window inspector dialog
Extensions Manager	#extensionsManager	This controls the extensions manager dialog.
Download Manager	#downloadManager	This controls the download manager dialog
Customize Window Dialog	#CustomizeToolbarWindow	Use this to set the primary characteristics of the Customize Toolbar dialog.
Bookmark Window	#bookmark-window	This sets the style for the bookmarks sidebar pane.
Page	page	This sets the general attributes for all pages within wizards.
Dialog	dialog	This is invoked by all dialog boxes to set up the default CSS. Can be used with others.
Wizard Tabs	wizard tabbrowser tab	In a wizard with multiple tabs, this sets thestyle of those tabs.
Active Menubar Menus		This sets the CSS properties for all menus that are accessible from the main menubar.
Active Menubar Menu Items	#main-menubar menupopup > menu-item[_moz-menuactive='true']	This sets the CSS properties for each active menu item.
Active Menubar Secondary Popup Menus	#main-menubar popup > menu[_moz-menuactive='true']	This sets the CSS for secondary menus, or for alternative popups that are launched via menus.
Active Menubar Secondary Popup Menu Items	#main-menubar popup > menu-item[_moz-menuactive='true']	This sets the CSS for items within secondary menus
Hover State for Generic Menu	menupopup > menu:hover	When a person hovers over a menu, it can change state. This lets you set the CSS to determine what that will look like for all menus.
Hover State for Main Menu	#main-menubar popup > menu:hover	Same as above, but this controls the main menubar menus only. You can subclass this with the previous entry.
Hover State for Main Menu Menu-Item	#main-menubar menupopup > menu-item:hover	This controls the hover state for an individual menu item on the main menu.

Rule Name	Rule Selector	Description
Hover State for Secondary Menu's Menu-Item	#main-menubar popup > menuitem:hover	This controls the hover state for menu items on secondary menus.
Titlebar Menubar	#titlebar menubar > menu	This lets you set menu 'buttons' when the menus are in the titlebar rather than beneath it.
Topbox	#topbox	This establishes an area that includes the titlebar and toolbar regions.
Toolbox	toolbox	The toolbox consists of the container for the menu and associated toolbars. Use this to set text characteristics
Titlebar	#titlebar	This covers the titlebar region exclusively.
Titlebar Separator	#titlebar #titlebar-separator	The titlebar separator acts as a graphic divider between the titlebar region and the menu if the menu is in fact located on the titlebar. If the menu is below the titlebar, this can be used as a secondary 'overlay' graphic.
Titlebar Menu Stack	#titlebar .titlebar-menu-stack	This defines the styles used for background and borders for the stack holding a titlebar based menu. This is not used if the menu is located below thetitlebar.
Titlebar Title	#titlebar #titlebar-title	This sets the font and text characteristics of the tilebar title.
Statusbar	statusbar	This describes the style associated with the statusbar at the bottom of the primary view window.
Sidebar Box	#sidebar-box	This provides the CSSstyling for the sidebar
Sidebar Header Text	.sidebar-header-text	This defines the header text used for sidebar content.
Sidebar Header	.sidebarheader-main	This provides the general properties of a sidebar header.
Sidebar Panel Picker	#sidebar-panel-picker	This provides the styling for the sidebar panel-picker element
Sidebar Panel Picker Hover State	#sidebar-panel-picker:hover	This gives the hover state for a sidebar picker.
Open Sidebar PanelPIcker	#sidebar-panel-picker[open='true']	This shows the open state for a sidebar picker.

This list is not exhaustive, and if you wished to you could go into the browser source code to find out the names of additional CSS selectors, but this provides a reasonably complete set.

A Short Primer on CSS

CSS is one of those languages which can be deceptively simple on the surface, but which can get very complicated once you get into the details of real world experience. While a comprehensive look at CSS can take up several hundred pages, the CSS necessary to work with Netscape can be summarized as follows:

Selectors

A selector lets you specify a certain pattern of HTML or XML elements within a document and then associates a set of style properties to that pattern. A given pattern, called a *selector*, coupled with the style properties for that pattern are together called a style rule, typically of the form:

```
selector {property1:value1;property2:value2;property3:value3;...}
```

For instance, you can set the CSS that applies to all dialog boxes within an application as:

```
dialog {background-color:white;color:black;font-family:Arial;}
```

which sets the background color to white, the text color to black,and the default font family to "Arial".

CSS is short for Cascading Stylesheets, which describes the underlying inheritance model. In general, if you apply a style to a container elements, most of the CSS properties that are assigned to the container are inherited by any child elements of that property, unless they are specifically overridden or unless it would make no sense to inherit (such as the display property, which controls how the element flows in with other elements).

This means that you can define global characteristics for a given window, dialog or wizard page then define subordinate rules that either augment or replace existing rules.

Selectors use a basic syntax to match given sets of elements:

- **Element Matches.** Matches of the form `elementName {propertyset}` with no modifiers on the name. This will match all elements that are found within the document that have that name.
- **Anonymous Matches.** Matches of the form `* {propertyset}` where the asterisk indicates that this should apply to all elements.
- **Class Matches.** Matches of the form `.classname {propertyset}` where classname is a name given as one of potentially many tokens in an element's `class` attribute.
- **Indexed Matches.** Matches of the form `.#identifier {propertyset}` where the identifier is an id attribute on a particular element. Since id-based items are unique, this will almost invariably refer to only one element in the associated document.
- **Conditional Matches.** A conditional match is of the form `selector [attr]` or `selector [attr = 'value']` where `attr` is the name of an attribute to be found on an element in the document. For instance, `menuitem[_moz-menuactive='true']` indicates that the selector matches all menuitems where the _moz-menuactive attrtibute has the value of true. If no value is given, then this rule applies to all elements that have the `_moz-menuactive` attribute, regardless of the actual value.
- **Descendent Matches.** CSS Selectors provide an set language, and by placing one selector after another, you are tellingCSS to retrieve all descendents of the first selector element that conform to the second selector. You can use the > symbol between the selectors to indicate that you only want to search the immediate children of the first element that correspond to the second. Subsequent selectors in the same rule treat all valid elements from the previous query as the base from which to search.
- **Mixed Matches.** Mixed matches combine multiple match techniques into a more restrictive selector. For instance, `#main-menubar menupopup > menuitem[_moz-menuactive='true']` indicates that CSS should find the element associated with the id "main-menubar", retrieve all **`<menupopup>`** children of that element, then retrieve all **`<menuitem>`** elements of the menupop. Finally, of those menu-items, choose only those for which the attribute `_moz-menuactive` exists and has the string value of "true".
- **Sequential Matches.** You can have more than one selector share the same property set by using a comma (,) to separate each selector prior to the property list. For instance,

```
window,dialog {background-color:lightGray,color:black;font-family:Arial;}
```

will set both the window and dialog elements to have the same light-gray background, black text color, and font-family of Arial.

- **Pseudo-Element Matches.** These are matches of the form `selector:action {properties}` A pseudo-element typically controls certain aspects of behavior of an element, such as the :hover, :active, and :visited properties which occur when the mouse hovers over or clicks on an element, or when a linked element is revisited. While the Netscape browser supports a wide number of these elements, most of the time you'll likely use only a very limited subset for XUL development. For instance,

  ```
  menuitem:hover {background-color:lightBlue;}
  ```

  will set the color of a menuitem light blue when a user places the mouse over that item.

This is, of course, not a completely comprehensive list of selector options, but should cover enough to be able to work with the master.css file. For more information about the selectors that are available within both Mozilla Firefox and the Netscape 8.0 browser, check out http://wiki.mozilla.org/Help:User_style#Css while a good set of guidelines for designing and using CSS selectors for your own applications can be found at http://www.mozilla.org/xpfe/goodcss.html.

Using CSS Properties

The original role of Cascading Stylesheets was to provide a markup independent mechanism for "presenting" HTML content, making it possible for HTML to become increasingly oriented as a "logical" style language. XUL has adopted CSS as the language it uses for specifying its own internal presentation characteristics, and as such, a good knowledge of CSS is almost mandatory to develop rich themes. Such a course of action is beyond the scope of this particular theme kit. Instead, this section will focus on a few basic principles and conventions within CSS that are especially useful for XUL theme developers.

- **CSS Specifications.** You can find the formal specification for the CSS language (CSS v. 2.0) at http://www.w3.org/TR/REC-CSS2. This link also includes download links for zipped HTML and PDF versions which can be saved to your system, especially useful if you develop from a laptop computer.

- **Mozilla CSS Properties.** The Mozilla Firefox engine on which Netscape is based also has created a number of useful CSS-like properties, most of which have been resubmitted and are in consideration for inclusion in CSS 3.0. A useful listing of some of these properties can be found at http://www.blooberry.com/indexdot/css/properties/extensions/nsextensions.htm or at http://www.xulplanet.com/references/elemref/ref_StyleProperties.html. These properties are typically prefixed by a "-moz-" extension(such as `-moz-opacity`)that indicates that they are experimental and currently Mozilla specific

- **Include Units.** The use of units within Mozilla CSS is mandatory - the CSS engine will not not automatically assume that a CSS property is the same as `{width:25px;}`. The typical response in such cases is for the rendering engine to ignore that property and all properties after it, making debugging especially frustrating.

- **Use the chrome protocol.** The reflection of languages, themes, overlays and bindings together conspire to make specifying the location of resources especially tough. For this reason, you should never use relative references to graphics or images, but instead should specify the chrome location of those resources. More information about chrome is covered in the next section.

- **Employ Transparencies.** The Netscape application supports transparent XUL containers, either through the use of PNG files as backgrounds for such containers or using the -moz-opacity property. PNG files with alpha-channel support can appear partially translucent, making them ideal for background images in popups.

- **Deploying Image Maps.** On a related note, the Netscape browser makes extensive use of "image maps", alpha-channel supporting PNG or GIF images that contain collections of buttons and related interface graphics. Rather than retaining the hundreds of individual graphics, Netscape utilizes

  The Netscape 8.0 Theme SDK includes several graphics files that hold the critical buttons and backgrounds for the

- **Important!** For ease of use, many of the more common CSS rules (especially from browser.css) have been placed in the master.css file. You can use either stylesheet, but if you do decide to use master.css, you must make sure that every rule contains the `important!` directive. This instructs the browser to use this rule in preference to an others declared, even if the others are declared at a later point in the build cycle.

  The Netscape 8.0 Theme SDK includes several graphics files that hold the critical buttons and backgrounds for the

Understanding Chrome

The Netscape 8.0 browser is considered to be one of the family of Mozilla browsers (specifically Mozilla 1.7) which also forms the foundation of such applications as the Firefox web browser, Thunderbird mail application, the Mac OSX Camino, and others. This means that it is built upon the XML User-interface language (XUL) and utilizes (for the most part) services built upon the Mozilla API (known by the more colorful sobriquet "Gecko"). The principal difference between the Netscape and Firefox browsers is the fact that Netscape also includes the capability of running Internet Explorer as part of its internal architecture, though this is hosted within the Gecko environment. From a themability standpoint, the Internet Explorer component is irrelevent.

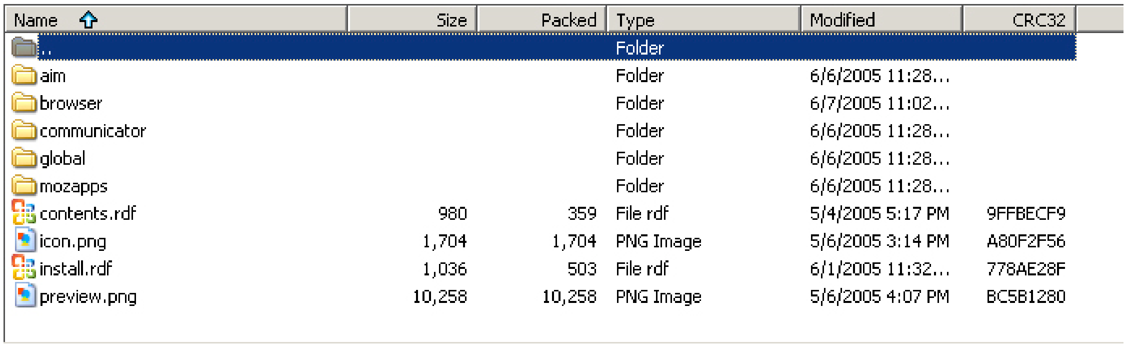
Under both the original Mozilla engine and the current design used by Netscape, the Chrome is both a generic term that describes the various pieces that make up a theme (and, for that matter, that make up most applications) as well as having a more specific meaning. In the latter case, the *chrome* is a protocol, in much the same way that http: is a protocol, that is used to describe the location of resources within the application in a platform neutral, theme and language aware environment.

The reason for such a system should become evident with a little bit of thought. For instance, consider the need to access a particular background image called background.jpg used for the titlebar (assuming the background has an image in the first place). A URL , even a file URL, will give an absolute location for `background.js`, something that is very useful for web content in general. However, within Netscape, each particular theme that is employed in the browser may (and likely will) have a different graphic for the same general position, quite probably with the same name.

Given this, the Netscape browser needed to have some way of being able to uniquely identify one such resource among many possible themes, and to do so it used the concept of the Mozilla chrome: protocol. Within chrome:, a file reference first must get resolved by the chrome: parser by looking up the current theme and language versions, which in turn then points to the actual file within the theme itself. This system, while a little awkward at first blush, works surprisingly well for the use-cases which Netscape has to deal with.

The chrome usually does not deal directly with file systems per se. Instead, it treats the theme jar file as a distinct file system, and usually relies upon a specific set of folders to be within the JAR. The root directory of the jar then acts as the root for the file system, containing general installation scripts (install.rdf) and other system wide components (see Figure 2.1, “Root Directory of the Generic Theme”)

Figure 2.1. Root Directory of the Generic Theme



Name	Size	Packed	Type	Modified	CRC32
..			Folder		
aim			Folder	6/6/2005 11:28...	
browser			Folder	6/7/2005 11:02...	
communicator			Folder	6/6/2005 11:28...	
global			Folder	6/6/2005 11:28...	
mozapps			Folder	6/6/2005 11:28...	
contents.rdf	980	359	File rdf	5/4/2005 5:17 PM	9FFBECF9
icon.png	1,704	1,704	PNG Image	5/6/2005 3:14 PM	A80F2F56
install.rdf	1,036	503	File rdf	6/1/2005 11:32...	778AE28F
preview.png	10,258	10,258	PNG Image	5/6/2005 4:07 PM	BC5B1280

For instance, a reference such as:

`chrome://browser/skin/background.png"`

points not to a specific file, but to a particular file to be found based upon the theme that is currently selected. The default theme for Netscape, called the Winscape theme, indicates that a chrome reference should look in the Winscape.jar file in the Netscape application directory for the resource, based upon a special resource called install.rdf.

The install.rdf file contains the metadata that uniquely identifies a given resource, and should always be in the root directory of any theme Jar file. It replaces the older install.js file which was used by earlier Mozilla builds, and is specifically searched for by the theme installer in order to help it perform the installation.

Example 2.2. The Generic Theme install.rdf file

```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:em="http://www.mozilla.org/2004/em-rdf#">
  <Description about="urn:mozilla:install-manifest">
    <em:id>{7ace5bd0-bd83-11d9-9669-0800200c9a66}</em:id>
    <em:version>8.0</em:version>
    <!-- Target Application this extension can install into,
         with minimum and maximum supported versions. -->
    <em:targetApplication>
      <Description>
        <em:name>Netscape</en:name>
        <em:id>{3db10fab-e461-4c80-8b97-957ad5f8ea47}</em:id>
        <em:minVersion>8.0</em:minVersion>
        <em:maxVersion>8.1</em:maxVersion>
      </Description>
    </em:targetApplication>
    <!-- Front End MetaData -->
    <em:name>Generic Theme</em:name>
    <em:description>A generic theme to be used as a base for building
          other themes for the Netscape browser.</em:description>
    <em:creator>Kurt Cagle</em:creator>
    <em:creator>Raj Paul</em:creator>
    <em:homepageURL>http://www.netscape.com</em:homepageURL>
    <!-- Front End Integration Hooks (used by Theme Manager)-->
    <em:internalName>Generic Theme</em:internalName>
  </Description>
</RDF>
```

The install.rdf file can look a little daunting at first, especially with regard to the <em:id> elements. These particular elements are UUIDs (or GUIDs, if you come from a Windows background) - time-stamp derived "numbers" which are statistically unique with a one in 3,402,823,669,209,385,000,000,000,000,000,000,000,000 chance of generating the same ID. They consequently serve very nicely as a way of uniquely identifying a given component over the web.

Note

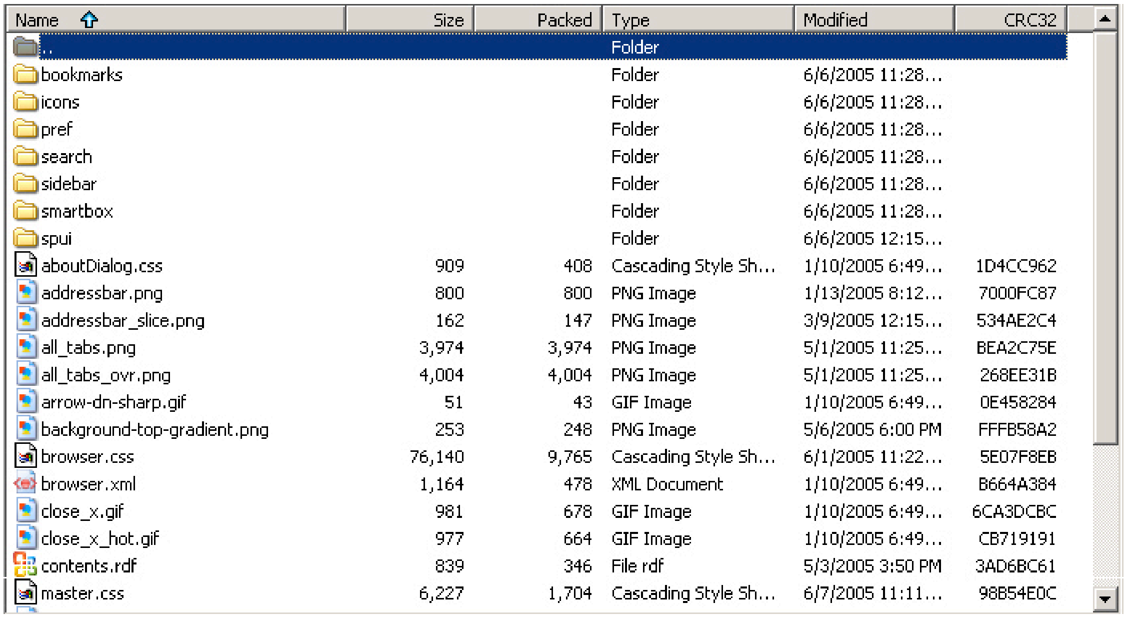
There are a number of free GUID generators on the web, including the one which generated this UUID at http://kruithof.xs4all.nl/uuid/uuidgen. When you create a new theme (or other XUL resource, such as an extension), generate the GUID and paste it into the primary <em:id> element.

The <em:targetApplication> indicates which application this theme is target to, and should be kept as shown above. The <em:id> again is perhaps the most critical property, as this, rather than the name, uniquely defines the Netscape browser itself. The version information indicates which versions of the browser this particular theme can be used with - if the theme mechanism changes, you would use these properties to insure that someone doesn't attempt to install a theme into a version which doesn't have the means to be able to accept it.

You should also set the <em:version to provide for a version control for the theme. Keep in mind that themes can have programmatic components, can introduce bugs, and may need to be updated with critical information, so version control is very important when dealing with such resources.>

Perhaps the most significant directory within the JAR file itself (at least with respect to themes) is the Browser folder. This folder will typically contain most, if not all, of the critical files for the theme itself, including the Master.css file referenced within this SDK. It will also usually have most of the primary graphics that are used for buttons or backgrounds. A partial listing of the contents of the generic theme browser folder is shown in Figure 2.2, “Browser Directory of the Generic Theme”

Figure 2.2. Browser Directory of the Generic Theme



Name	Size	Packed	Type	Modified	CRC32
..			Folder		
bookmarks			Folder	6/6/2005 11:28...	
icons			Folder	6/6/2005 11:28...	
pref			Folder	6/6/2005 11:28...	
search			Folder	6/6/2005 11:28...	
sidebar			Folder	6/6/2005 11:28...	
smartbox			Folder	6/6/2005 11:28...	
spui			Folder	6/6/2005 12:15...	
aboutDialog.css	909	408	Cascading Style Sh...	1/10/2005 6:49...	1D4CC962
addressbar.png	800	800	PNG Image	1/13/2005 8:12...	7000FC87
addressbar_slice.png	162	147	PNG Image	3/9/2005 12:15...	534AE2C4
all_tabs.png	3,974	3,974	PNG Image	5/1/2005 11:25...	BEA2C75E
all_tabs_ovr.png	4,004	4,004	PNG Image	5/1/2005 11:25...	268EE31B
arrow-dn-sharp.gif	51	43	GIF Image	1/10/2005 6:49...	0E458284
background-top-gradient.png	253	248	PNG Image	5/6/2005 6:00 PM	FFFB58A2
browser.css	76,140	9,765	Cascading Style Sh...	6/1/2005 11:22...	5E07F8EB
browser.xml	1,164	478	XML Document	1/10/2005 6:49...	B664A384
close_x.gif	981	678	GIF Image	1/10/2005 6:49...	6CA3DCBC
close_x_hot.gif	977	664	GIF Image	1/10/2005 6:49...	CB719191
contents.rdf	839	346	File rdf	5/3/2005 3:50 PM	3AD6BC61
master.css	6,227	1,704	Cascading Style Sh...	6/7/2005 11:11...	98B54E0C

To accesss specific files within the theme via the chrome you have to rely upon the folder structure, though a specific path `$THEME_JAR/browser/myFile.jpg` doesn't correspond directly to the URL. Instead, your chrome reference would look like `chrome://browser/skin/myFile.jpg`, where the browser folder corresponded to $THEME_JAR/browser, the /skin/ indicated that it was referenced as the current theme, and the filename maps to the filename.

It's worth noting here that this can also be used to good effect for additional libraries. For instance, if you had an extension for Netscape called myLib.xpi with global objects contained in a /global/ directory in that JAR file (XPI's are also JARs), you could reference it from within your CSS as `chrome://mylib/global/myFile.jpg.`

Warning

The chrome provides a measure of security for your users, and as a theme developer you should respect this, as you have heightened access into a user's environment. For this reason, you should only include chrome references into your CSS, not external URL calls (Netscape specifically removes any such references when evaluating and preparing themes for common download, but if you host a theme yourself you should also keep this warning under consideration.)

Chapter 3. Building and Modifying Themes

Creating new themes in general is not complicated, though it does involve a certain amount of work up front to make things happen. This chapter focuses on the creation process, walking you through the steps to make a simple theme from the generic theme JAR and the Toolkit.

Designing the Theme

Creating a theme is obviously an artistic endeavor - your are seeking a way to be able to present your visions and artistic sensibility to others. While it is possible to create themes one piece at a time, such a process is usually fraught with peril. Thus, the best solution as a theme designer is to take a screen shot of an existing theme as deployed in Netscape and use that as a starting point. As you are designing, you should keep a number of factors in mind.

Protect Yourself!

Before creating any new theme, protect yourself from grief later by creating a folder with the theme's name (for this example, you can always change it later), and drag a copy of the default theme jar file and the image templates into this new folder. Always work on the copy. For purposes of discussion, the theme being generated here is the Rustic Theme.

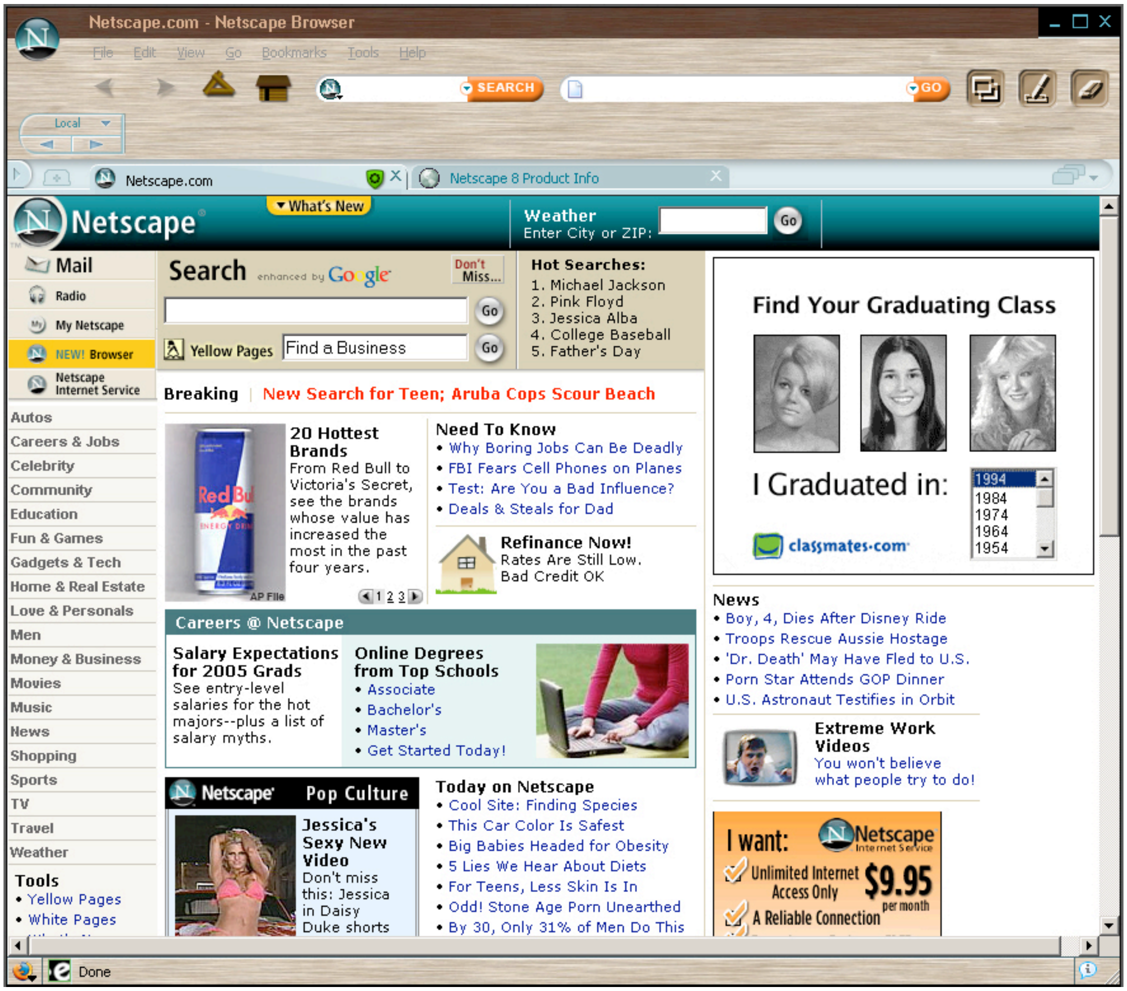
Creating a Prototype

Just as no sane architect would create a blue-print of a house without first putting together both drawings and scale models of a given building, so too should you plan ahead in establishing how you envision the final theme will look like. One good technique is to take a screen shot of an existing browser theme similar to your own, then edit it in your favorite graphics editor by adding new backgrounds, changing the general shape and functionality of buttons and so forth. By making each change on a different layer, you can also generate many of the graphics that you will likely develop anyway.

Inspirations for themes can come from anywhere, and with the use of photographic images and high quality gradients you can generally build themes for just about any topic. Please keep in mind, however, that if you wish to distribute themes (either through your own site or through the Netscape Theme Park) you can only use a given graphic or image (or piece thereof) if you created it, or you have the express written permission of the creator under some form of license agreement.

In order to illustrate the point, this paper looks at the Rustic Theme (see Figure 3.1, “Rustic Theme Prototype”), included as part of the Netscape Theme SDK. Designed to give a "woodsy" feel to the browser, this particular theme is both relatively simple to implement yet sufficiently rich to show how themes are built.

Figure 3.1. Rustic Theme Prototype


Netscape.com - Netscape Browser
File Edit View Go Bookmarks Tools Help
SEARCH
GO
Local
Netscape.com
Netscape 8 Product Info
Done

Netscape

What's New

Weather
Enter City or ZIP:
Go

Mail
Radio
My Netscape
NEW! Browser
Netscape Internet Service

Search enhanced by Google
Go
Yellow Pages
Find a Business
Go

Don't Miss...

Hot Searches:
1. Michael Jackson
2. Pink Floyd
3. Jessica Alba
4. College Baseball
5. Father's Day

Find Your Graduating Class
I Graduated in:
1994
1984
1974
1964
1954
classmates.com

Breaking | New Search for Teen; Aruba Cops Scour Beach

Autos
Careers & Jobs
Celebrity
Community
Education
Fun & Games
Gadgets & Tech
Home & Real Estate
Love & Personals
Men
Money & Business
Movies
Music
News
Shopping
Sports
TV
Travel
Weather

Tools
- Yellow Pages
- White Pages

20 Hottest Brands
From Red Bull to Victoria's Secret, see the brands whose value has increased the most in the past four years.
AP File
1 2 3

Need To Know
- Why Boring Jobs Can Be Deadly
- FBI Fears Cell Phones on Planes
- Test: Are You a Bad Influence?
- Deals & Steals for Dad

Refinance Now!
Rates Are Still Low.
Bad Credit OK

News
- Boy, 4, Dies After Disney Ride
- Troops Rescue Aussie Hostage
- 'Dr. Death' May Have Fled to U.S.
- Porn Star Attends GOP Dinner
- U.S. Astronaut Testifies in Orbit

Careers @ Netscape

Salary Expectations for 2005 Grads
See entry-level salaries for the hot majors--plus a list of salary myths.

Online Degrees from Top Schools
- Associate
- Bachelor's
- Master's
- Get Started Today!

Extreme Work Videos
You won't believe what people try to do!

Netscape Pop Culture
Jessica's Sexy New Video
Don't miss this: Jessica in Daisy Duke shorts

Today on Netscape
- Cool Site: Finding Species
- This Car Color Is Safest
- Big Babies Headed for Obesity
- 5 Lies We Hear About Diets
- For Teens, Less Skin Is In
- Odd! Stone Age Porn Unearthed
- By 30, Only 31% of Men Do This

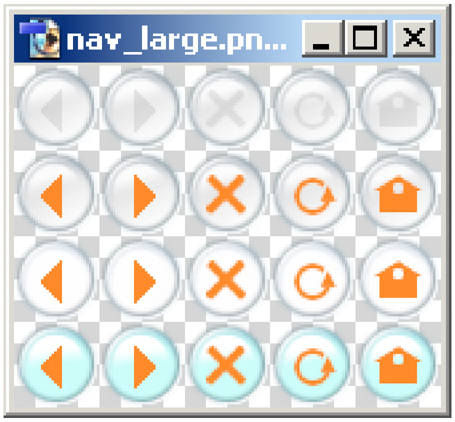
I want: Netscape Internet Service
Unlimited Internet Access Only $9.95 per month
A Reliable Connection

Standard Button Sets

It is not yet possible to create a translucent background for the browser itself (at least not yet), you can create translucent backgrounds for any contained element within the browser. This principle is utilized heavily for the production of buttons and related interactive graphics.

The current Netscape themes make use of a single PNG for collections of common buttons, then internally segments the buttons using a clipping rectangle to isolate the specific buttons. Most of these are containered in the $THEMEJAR/browser/icons/ folder. For instance, the larger navigation buttons are contained in a single file - $THEMEJAR/browser/icons/nav_large.png (see Figure 3.2, “Large Navigation Buttons File nav_large.png”).

Figure 3.2. Large Navigation Buttons File nav_large.png

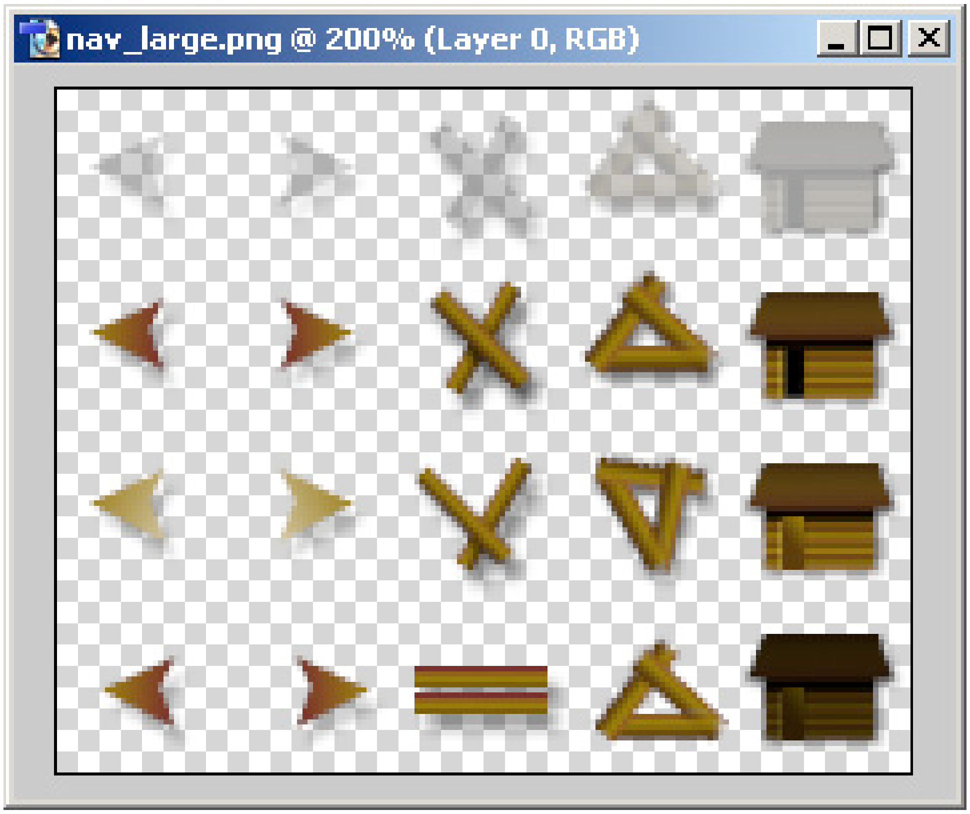


Templates

While you can resize such button sets using CSS, this can be a complex process at best as you will need to handle positioning each button from CSS separately. in general you're better off replacing the graphics with your own at the same size. One useful technique for doing this in applications such as Photoshop or GIMP is to place the original graphics in one layer, set the opacity of that layer to 25%, create a second layer on top of the first, then draw your new buttons over the old ones on the second layer. Prior to saving, remove the first layer and flatten as a PNG file.

You can replace the graphics with your own graphics, though you should make sure that you replace them exactly where they were in the original jar file (see Figure 3.3, “Creating the Rustic Theme nav_large.png icons”. It should also be noted that replacing button and system graphics in this way does not necessitate changes in the CSS stylesheets directly.

Figure 3.3. Creating the Rustic Theme nav_large.png icons



Large icons are 32x32 pixels in size, while small icons are 16x16. The grid of images left to right for the navigational elements provide the graphics as follows:

1. Previous web page in viewing history
2. Next web page in viewing history
3. Stop the current download process.
4. Refresh the current web page.
5. Go to the user-defined home page.

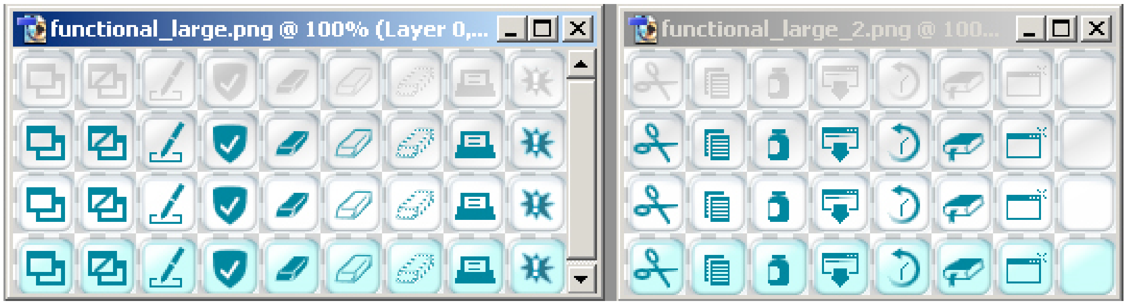
The rows in the diagram represent different states of each button, as follows:

1. Disabled state (button is inactive)
2. Default state (button is active but has not received focus)
3. Hover state (button is active and the mouse is hovering over the button)
4. Active state (button is active and being pressed/activated)

while these buttons represent the primary means of controlling the browser itself, Netscape also has a number of secondary functions which can be added via "customized" buttons. As with the navigational buttons, these can be found within the /browser/icon folder as `functional_large.png` and `func-`

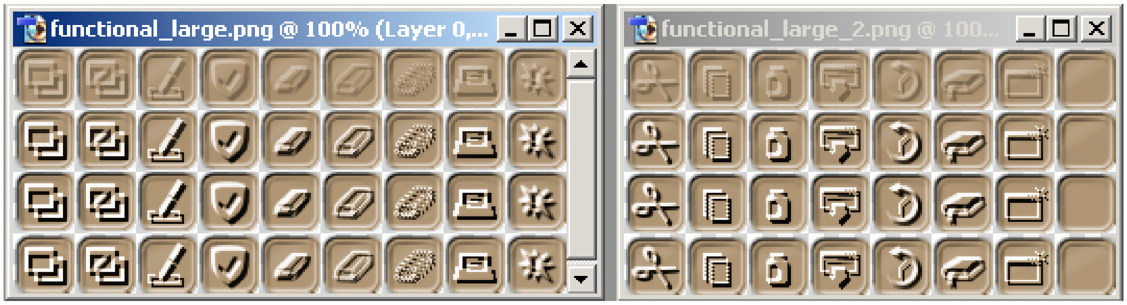
`tional_large_2.png`. (see Figure 3.4, “Generic Functional Buttons”).

Figure 3.4. Generic Functional Buttons


functional_large.png @ 100% (Layer 0,...
functional_large_2.png @ 100...

Modifying these buttons for the Rustic theme was quite simple - using Photoshop, the entire file full of buttons was first run through an emboss filter to make them appear as if carved, then run through the Hue/Saturation dialog and "colorized" to turn them into a wood-brown color. These were then saved back into the RusticTheme.jar file under their old name (see ???).

Figure 3.5. Rustic Functional Buttons


functional_large.png @ 100% (Layer 0,...
functional_large_2.png @ 100...

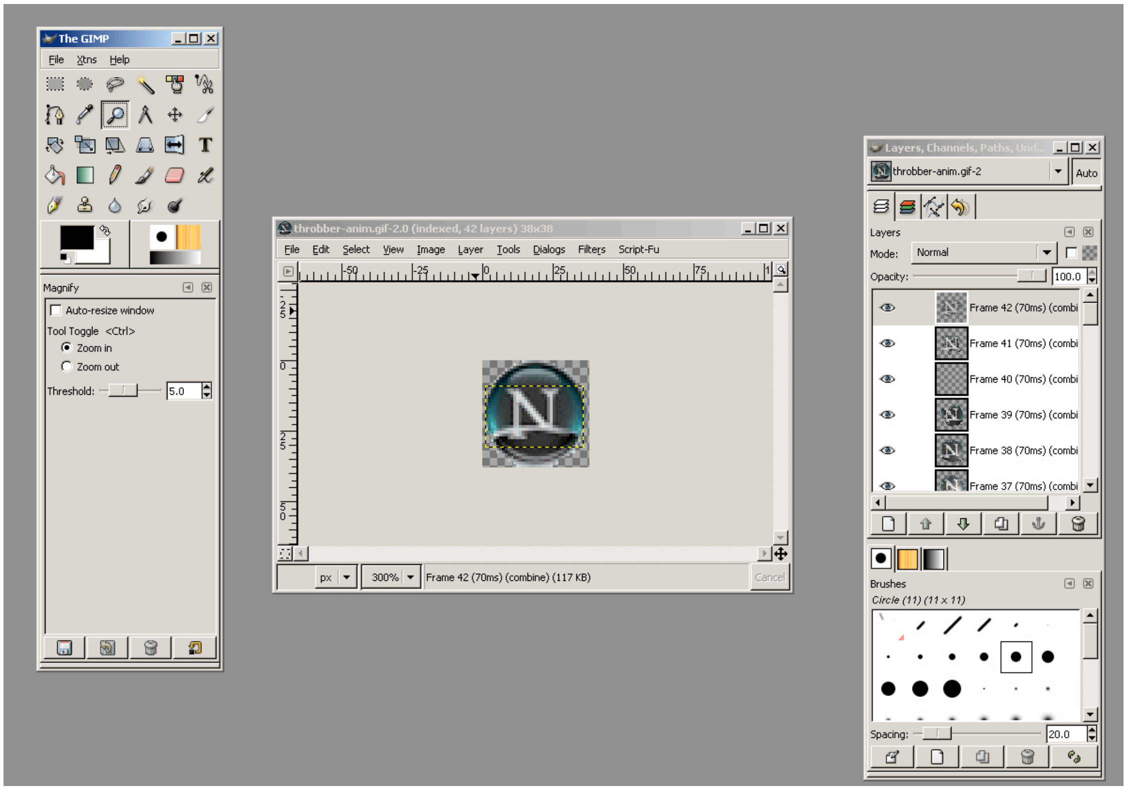
Throbber

The throbber is the name of the animated graphic that appears whenever a web page is being loaded, the trademark Netscape "N". It in fact consists of two distinct images, a static PNG file that shows the throbber in its "rest" state and an animated GIF file that contains the sequence shown while a page is being requested or refreshed.

These particular graphics are contained in the theme's /browser subfolder, as `throbber-bg.png` and `throbber-anim.gif`. The throbber-bg.png file is sized to fit exactly within the upper left corner of the application, while the GIF animation is specifcally positioned so that it appears on top of the throbber background element.

While editing the throbber-bg.png file is reasonably straight-forward and can be done in Adobe Photoshop, creating and editing animated GIFs is a little more problematic, as this capability is not supported by Photoshop except via plug-ins (such as BoxTop Software's GIFmation), though it is supported by Adobe ImageReady. If you want to go the open source route, GIMP 2.0 (http://www.gimp.org/windows/) provides a free alternative to Photoshop that lets you edit the frames of an animation as successive layers and control the timing of each framee of the animation Figure 3.6, “Creating Animations using Gimp 2”). The throbber animations continues to run until a file is downloaded or the Nescape browser application times out, so the animation duration is immaterial to the throbber's primary role.

Figure 3.6. Creating Animations using Gimp 2


The GIMP
File Xtns Help
Magnify
Tool Toggle <Ctrl>
throbber-anim.gif-2.0 (indexed, 42 layers) 38x38
File Edit Select View Image Layer Tools Dialogs Filters Script-Fu
px
Cancel
Layers, Channels, Paths, Und...
throbber-anim.gif-2
Auto
Layers
Brushes
Circle (11) (11 x 11)

☐ Auto-resize window
● Zoom in
○ Zoom out
Threshold: 5.0
300%
Frame 42 (70ms) (combine) (117 KB)
Mode: Normal
Opacity: 100.0
Frame 42 (70ms) (combi
Frame 41 (70ms) (combi
Frame 40 (70ms) (combi
Frame 39 (70ms) (combi
Frame 38 (70ms) (combi
Frame 37 (70ms) (combi
Spacing: 20.0

The throbber graphic and animation sequences are considered to be trademarks of Netscape corporations, and must be included on any Netscape Theme package unless the authors of the theme have the express written consent from Netscape Corporation or its agents to alter or replace the graphics.

Background Graphics

Backgrounds provide the foundation upon which your themes ride. The Generic background, while seeming to be fairly complex, relies upon a few basic tricks to minimize filesize - creating 20 pixel high/ one pixel wide graphics that can be repeated along one axis or another, for instance. Chances are good, however, that your own background may be more complex, and as such a slightly different approach should be taken in designing these backgrounds.

For Rustic, the basic background piece was created as a texture in Photoshop by using a combination of the Noise filter and a motion blur, coupled with colorizing the resulting gray streaks to make them look like wood, as shown in Figure 3.7, “The principle background pattern, woodBackground.jpg.” and in a darker version for the titlebar (Figure 3.8, “A darker version of the background, darkWoodBackground.jpg, for the titlebar.”).

Figure 3.7. The principle background pattern, woodBackground.jpg.

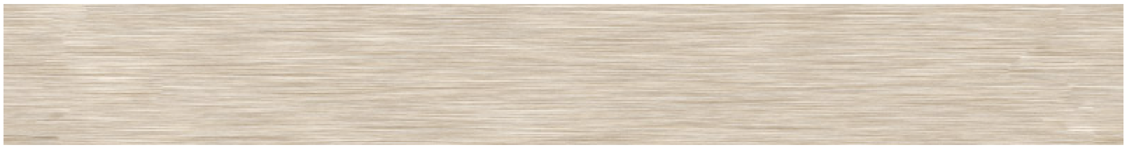


Figure 3.8. A darker version of the background, darkWoodBackground.jpg, for the titlebar.

Both of these should then be placed in the /browser folder of the RusticTheme.jar file.

Working with Theme Jars

In Netscape 8.0, a theme is an extension, a compressed resource that contains sets of CSS documents, XBL code, graphics, and similar resources. Each theme is wholly self-contained, and is identified by a specific name. When you download the stock Netscape 8.0 browser, the application ships with two distinct XPI files: Winscape, which provides a visual interface similar to that of the Mozilla Firefox browser, while the Generic them provides a sleeker "starship navigator" look and feel.

Note

This particular Theme SDK comes with its own generic theme (called Generic.jar), which includes a basic configuration for use in developing your own themes.

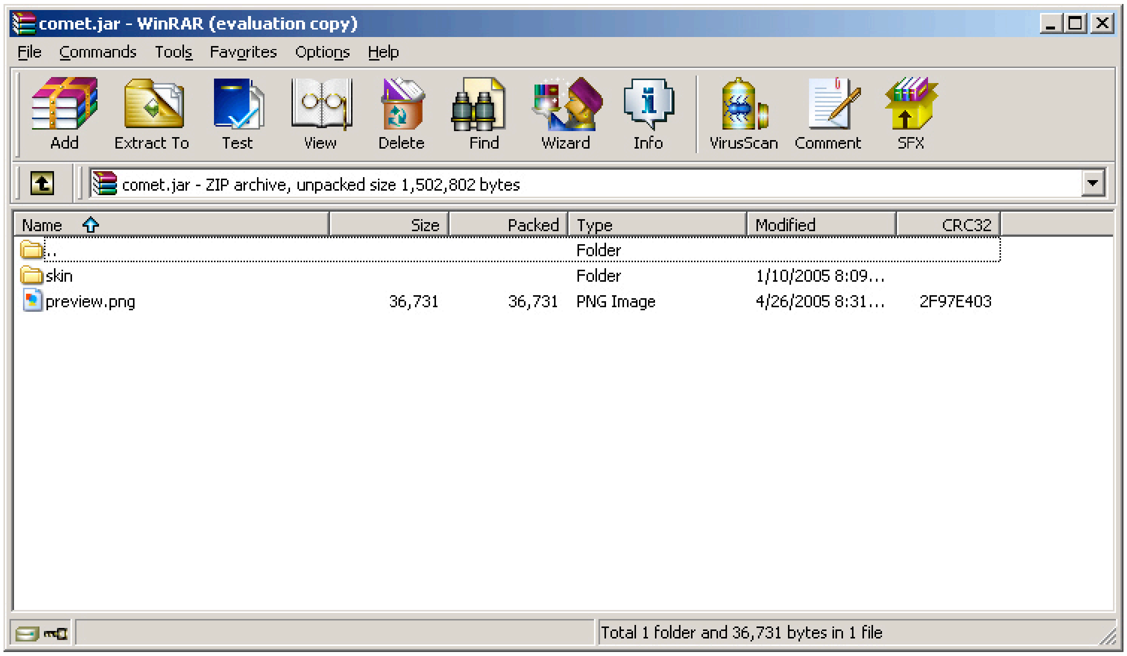
Each theme is a JAR file. If $NETSCAPE is the folder where your netscape.exe file is kept (by default, $NETSCAPE is set to `c:\Program Files\Netscape`) then you should be able to find all of the relevant them JAR files in $NETSCAPE\chrome.

A theme JAR file is a compressed file that uses the ZIP protocol for compression. This means that any application that lets you open up and examine a ZIP file (including several that are built into the most recent versions of Windows) should let you look at it. If you have such an application installed and you're still having trouble opening the JAR file, change the extension to ZIP and try opening it this way..

The best way to learn how to create a theme is to start with an existing one, such as the Generic theme. *Because it is very easy to make changes to a theme that leave it in an unstable (i.e., broken) state, you are strongly recommended to make a backup of the $NETSCAPE\chrome\Generic.jar file before making any changes*. Copy the jar file to someplace safe, then work on the original version within the $NETSCAPE\chrome folder. If you have a goof (or simply don't like the theme you've produced), then copy the newly created copy back over the one one in the chrome folder and restart your browser.

Open up the Generic.jar file using your ZIP tools, and you will see something that looks roughly like Figure 3.9, “Generic jar file root.”.

Figure 3.9. Generic jar file root.



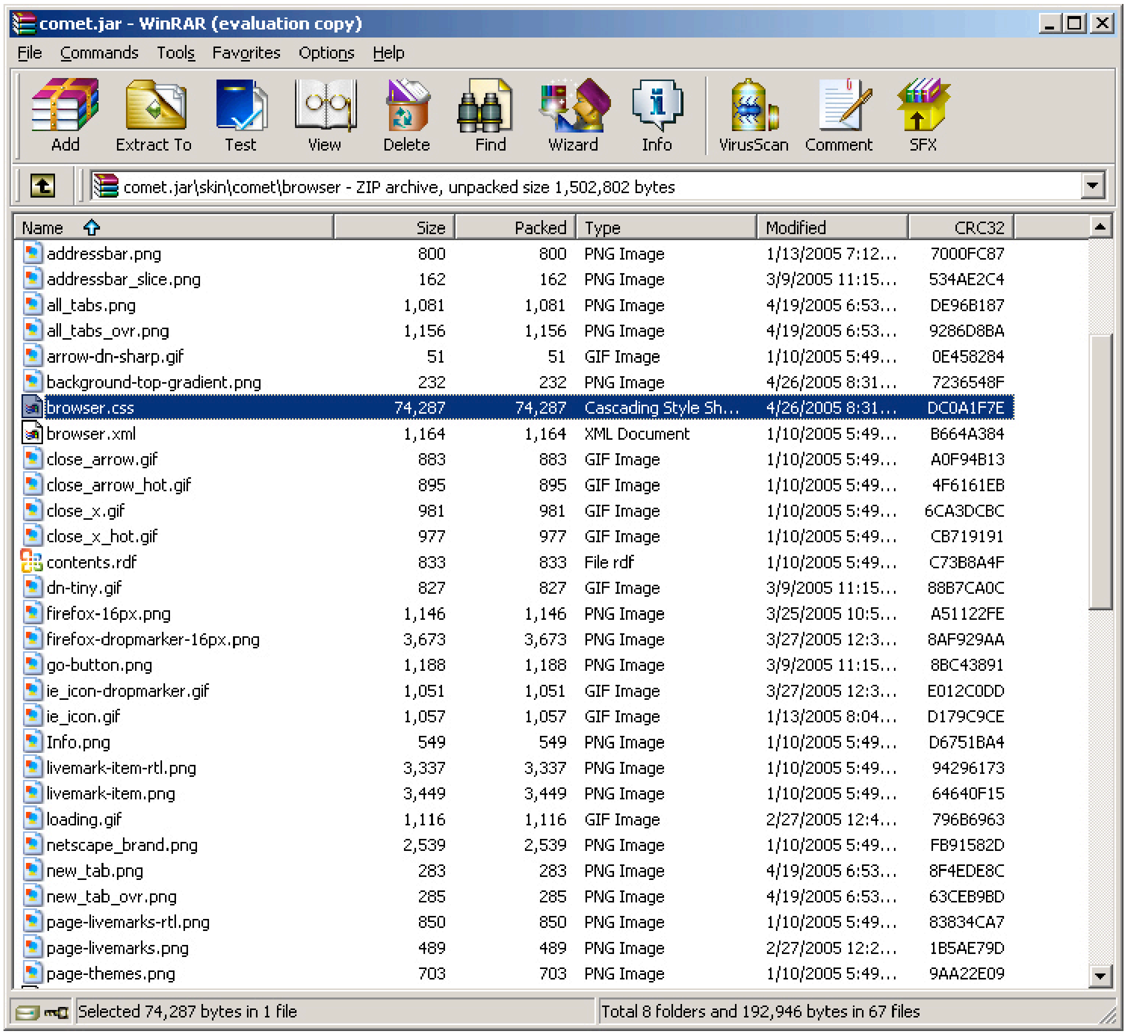
comet.jar - ZIP archive, unpacked size 1,502,802 bytes

Name	Size	Packed	Type	Modified	CRC32
..			Folder		
skin			Folder	1/10/2005 8:09…	
preview.png	36,731	36,731	PNG Image	4/26/2005 8:31…	2F97E403

Total 1 folder and 36,731 bytes in 1 file

Once here, select the `skin` folder, then `Generic\browser\` in order to get to the interesting bits (Figure 3.10, “Generic jar interesting bits.”). This is where the theme that the browser uses resides, regardless of the theme in question. You should make use of the same structure if you create your themes from scratch.

Figure 3.10. Generic jar interesting bits.



comet.jar\skin\comet\browser - ZIP archive, unpacked size 1,502,802 bytes

Name	Size	Packed	Type	Modified	CRC32
addressbar.png	800	800	PNG Image	1/13/2005 7:12...	7000FC87
addressbar_slice.png	162	162	PNG Image	3/9/2005 11:15...	534AE2C4
all_tabs.png	1,081	1,081	PNG Image	4/19/2005 6:53...	DE96B187
all_tabs_ovr.png	1,156	1,156	PNG Image	4/19/2005 6:53...	9286D8BA
arrow-dn-sharp.gif	51	51	GIF Image	1/10/2005 5:49...	0E458284
background-top-gradient.png	232	232	PNG Image	4/26/2005 8:31...	7236548F
browser.css	74,287	74,287	Cascading Style Sh...	4/26/2005 8:31...	DC0A1F7E
browser.xml	1,164	1,164	XML Document	1/10/2005 5:49...	B664A384
close_arrow.gif	883	883	GIF Image	1/10/2005 5:49...	A0F94B13
close_arrow_hot.gif	895	895	GIF Image	1/10/2005 5:49...	4F6161EB
close_x.gif	981	981	GIF Image	1/10/2005 5:49...	6CA3DCBC
close_x_hot.gif	977	977	GIF Image	1/10/2005 5:49...	CB719191
contents.rdf	833	833	File rdf	1/10/2005 5:49...	C73B8A4F
dn-tiny.gif	827	827	GIF Image	3/9/2005 11:15...	88B7CA0C
firefox-16px.png	1,146	1,146	PNG Image	3/25/2005 10:5...	A51122FE
firefox-dropmarker-16px.png	3,673	3,673	PNG Image	3/27/2005 12:3...	8AF929AA
go-button.png	1,188	1,188	PNG Image	3/9/2005 11:15...	8BC43891
ie_icon-dropmarker.gif	1,051	1,051	GIF Image	3/27/2005 12:3...	E012C0DD
ie_icon.gif	1,057	1,057	GIF Image	1/13/2005 8:04...	D179C9CE
Info.png	549	549	PNG Image	1/10/2005 5:49...	D6751BA4
livemark-item-rtl.png	3,337	3,337	PNG Image	1/10/2005 5:49...	94296173
livemark-item.png	3,449	3,449	PNG Image	1/10/2005 5:49...	64640F15
loading.gif	1,116	1,116	GIF Image	2/27/2005 12:4...	796B6963
netscape_brand.png	2,539	2,539	PNG Image	1/10/2005 5:49...	FB91582D
new_tab.png	283	283	PNG Image	4/19/2005 6:53...	8F4EDE8C
new_tab_ovr.png	285	285	PNG Image	4/19/2005 6:53...	63CEB9BD
page-livemarks-rtl.png	850	850	PNG Image	1/10/2005 5:49...	83834CA7
page-livemarks.png	489	489	PNG Image	2/27/2005 12:2...	1B5AE79D
page-themes.png	703	703	PNG Image	1/10/2005 5:49...	9AA22E09

Selected 74,287 bytes in 1 file

Total 8 folders and 192,946 bytes in 67 files

The theme contains a number of graphics files - though with GIF and PNG files predominating. The PNG format (for Portable Network Graphics) makes it possible to create partially transparent images in 24-bit color, and as such, is increasingly replacing the use of 8-bit GIF files for handling transparency-containing images.

Additionally, the theme at this level includes a number of cascading stylesheet (CSS) files, most significant of which being the file `browser.css`. This document serves to map XUL elements (referenced either through ids, in the form `#myItem`, or CSS classes of the form `.myClass`) to specific CSS style properties. This format does not differ in any significant way from that what is used for providing CSS for normal web development, although there are a few additional Mozilla/Netscape extensions which are specific to the XUL environment (more information on these can be found at http://xulplanet.mozdev.org/references/elemref/ref_StyleProperties.html .

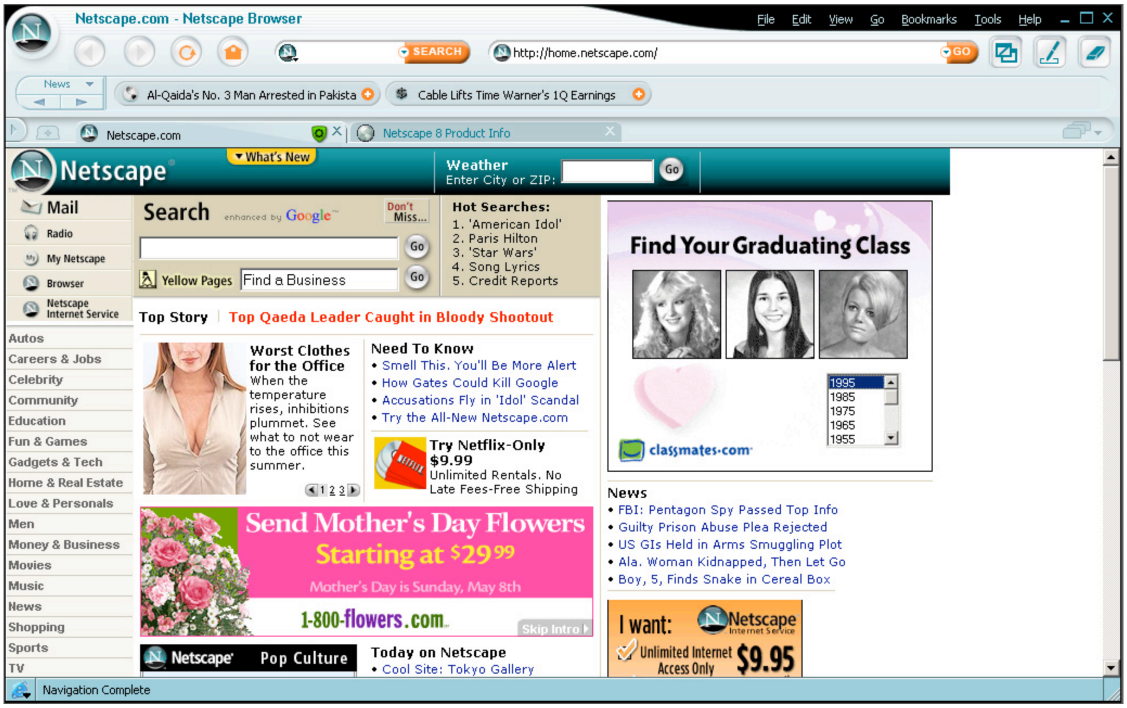
Modifying an Existing Theme

The browser.css document provides the specific CSS that will be used in changing the most significant aspects of the theme, though other CSS files are used to change the presentation on lower level control elements. There are a number of different style rules within browser.css, and in general the best way to learn how to work with creating new themes is to try altering different rules.

The Generic theme was intended to evoke a "starship" feel to it. By adapting it, you can change the theme to something with more of a "deep space" visualization, which will eventually be the Rustic Theme. In order to make such a theme work, its generally worthwhile first to mentally plan ahead of time how you want the theme to look. In the case of a Rustic theme, one possibility is to pull images from the Hubble telescope (many superb ones of which can be found at http://www.hubblesite.org) and use these as the basis for new graphics and related resources .

As a benchmark, it's worth considering the original Generic theme, then provide a screenshot of the new Rustic theme to show what's possible with simple modifications. The Generic Theme (Figure 3.11, “Netscape with the Generic Theme”), takes advantage of PNG transparancies to give the illusion of rounded corners and a three dimension appearance.

Figure 3.11. Netscape with the Generic Theme



Hot Searches:
1. 'American Idol'
2. Paris Hilton
3. 'Star Wars'
4. Song Lyrics
5. Credit Reports

Top Story | Top Qaeda Leader Caught in Bloody Shootout

Worst Clothes for the Office When the temperature rises, inhibitions plummet. See what to not wear to the office this summer.

Need To Know
- Smell This. You'll Be More Alert
- How Gates Could Kill Google
- Accusations Fly in 'Idol' Scandal
- Try the All-New Netscape.com

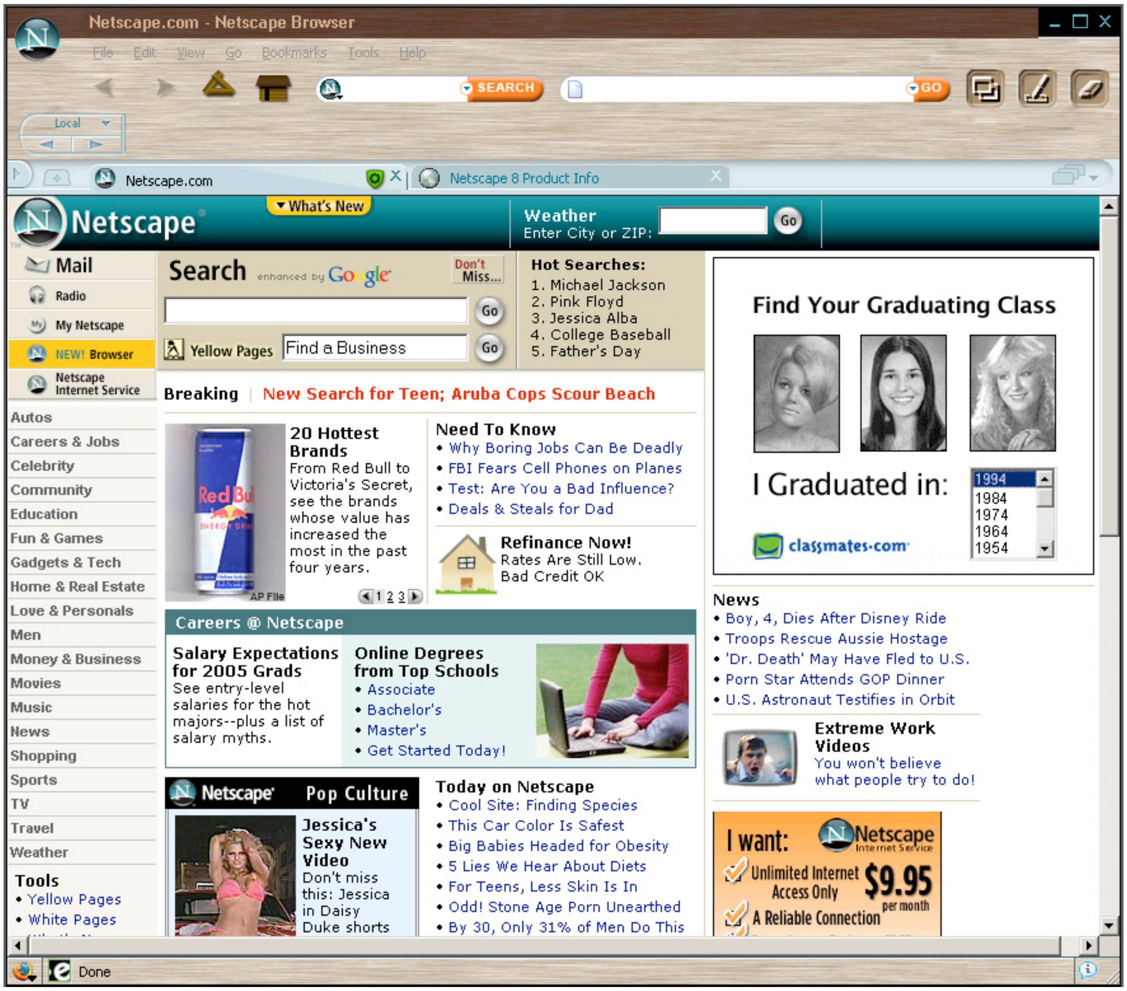
Try Netflix-Only $9.99 Unlimited Rentals. No Late Fees-Free Shipping

News
- FBI: Pentagon Spy Passed Top Info
- Guilty Prison Abuse Plea Rejected
- US GIs Held in Arms Smuggling Plot
- Ala. Woman Kidnapped, Then Let Go
- Boy, 5, Finds Snake in Cereal Box

Today on Netscape
- Cool Site: Tokyo Gallery

By swapping out a few graphics and changing a few lines of CSS, this can be changed to reflect the Rustic theme (Figure 3.12, “Netscape with the Rustic Theme”):

Figure 3.12. Netscape with the Rustic Theme


Netscape.com - Netscape Browser
File Edit View Go Bookmarks Tools Help
SEARCH
GO
Local
Netscape.com
Netscape 8 Product Info
Done

Netscape
What's New
Weather
Enter City or ZIP:
Go

Mail
Radio
My Netscape
NEW! Browser
Netscape Internet Service

Search enhanced by Google
Don't Miss...
Yellow Pages Find a Business

Hot Searches:
1. Michael Jackson
2. Pink Floyd
3. Jessica Alba
4. College Baseball
5. Father's Day

Breaking | New Search for Teen; Aruba Cops Scour Beach

Autos
Careers & Jobs
Celebrity
Community
Education
Fun & Games
Gadgets & Tech
Home & Real Estate
Love & Personals
Men
Money & Business
Movies
Music
News
Shopping
Sports
TV
Travel
Weather

Tools
- Yellow Pages
- White Pages

20 Hottest Brands
From Red Bull to Victoria's Secret, see the brands whose value has increased the most in the past four years.

AP File

Need To Know
- Why Boring Jobs Can Be Deadly
- FBI Fears Cell Phones on Planes
- Test: Are You a Bad Influence?
- Deals & Steals for Dad

Refinance Now!
Rates Are Still Low. Bad Credit OK

Careers @ Netscape

Salary Expectations for 2005 Grads
See entry-level salaries for the hot majors--plus a list of salary myths.

Online Degrees from Top Schools
- Associate
- Bachelor's
- Master's
- Get Started Today!

Netscape Pop Culture

Jessica's Sexy New Video
Don't miss this: Jessica in Daisy Duke shorts

Today on Netscape
- Cool Site: Finding Species
- This Car Color Is Safest
- Big Babies Headed for Obesity
- 5 Lies We Hear About Diets
- For Teens, Less Skin Is In
- Odd! Stone Age Porn Unearthed
- By 30, Only 31% of Men Do This

Find Your Graduating Class
I Graduated in: 1994 1984 1974 1964 1954
classmates.com

News
- Boy, 4, Dies After Disney Ride
- Troops Rescue Aussie Hostage
- 'Dr. Death' May Have Fled to U.S.
- Porn Star Attends GOP Dinner
- U.S. Astronaut Testifies in Orbit

Extreme Work Videos
You won't believe what people try to do!

I want: Netscape Internet Service
Unlimited Internet Access Only $9.95 per month
A Reliable Connection

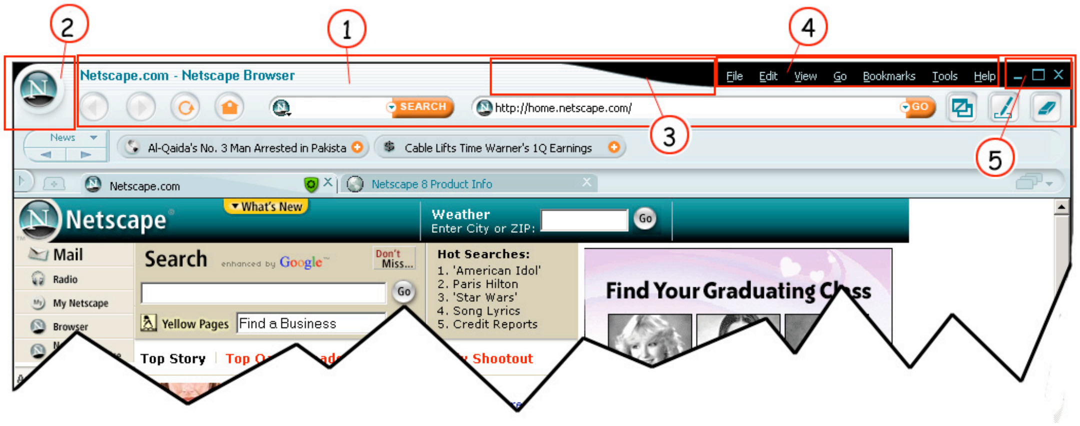
The changes in the theme are fairly minor in terms of the overall look and feel - a full theme would likely swap out buttons, alter placement of tabs, change default colors and modify font families and similar resources, but most of these changes utilize the same techniques as are discussed for the simple title bar change.

Themes are generally built up working on specific XUL regions, which are usually rectangular in nature. What this means in general is that you will often need to be cognizant of multiple overlapping areas within the web browser, which usually translates into being conscious about how transparencies are used. The principal areas to be changed in our Generic to Rustic conversion are shown in Figure 3.13, “Regions to edit in the Rustic theme”, with callouts describing the regions in detail.

Note

Once you save the changes you make back into the JAR file, you can actually see the changes within the web browser. However, as the themes are loaded into the browser at start-up time, you should generally close your browser before modifying the jar, then reopening it to reinitialize the themes engine.

Figure 3.13. Regions to edit in the Rustic theme



1. The topbox. This is the foundation upon which the titlebar region is built. In the Generic topbox, a background image (1-pixel wide) is used to create the lined pattern that runs the length of the titlebar. The CSS for this tobbox points to the local reference for the graphic file itself (see Example 3.1, “The TopBox CSS”)

Example 3.1. The TopBox CSS

```
#topbox {
    background-color: transparent;
    background-image:url("chrome://browser/skin/background-top-gradient.png");
    background-repeat: repeat-x;
                }
```

To change the theme to reflect the Rustic view, a new graphic was created by the same name, but with different horizontal dimensions:

Figure 3.14. New Rustic Background-Top-Gradient

This was then added to the Generic.jar file. The browser.css file in this case was unchanged. This will cause the graphic to repeat in the horizontal dimension over the length of the toolbar

2. The throbber. The throbber is a graphic that is used to indicate that the browser is performing some action (usually searching for or navigating to a new web page). As with the topbox, the source graphic for the default throbber state is one we wish to keep (it is Netscape's browser, after all), but because the throbber's initial image incorporates part of the repeating pattern that is also used by the topbox, the throbber graphic should be modified to make this background transparent in an application such as Photoshop. The new throbber graphic (Figure 3.15, “Displaying the throbber element.”) keeps the circular region of the Generic theme intact, but eliminates the background (the animated gif states for the throbber are already transparent).

Figure 3.15. Displaying the throbber element.

The CSS for this code can be found under the browser.css entry throbber-box (Example 3.2, “Throbber-box CSS declaration.”)

Example 3.2. Throbber-box CSS declaration.

```
#throbber-box {
    background-image: url("chrome://browser/skin/throbber-bg.png");
    min-width: 58px;
    min-height: 60px;
                }
```

3. The titlebar-separator. The TopBox region includes two distinct areas, the title bar and the navigator buttons. The title bar in the Generic implementation overrides the default behavior where the menu is separate, instead, incorporating the menu into the titlebar itself, a mode that is becoming increasingly common in user interfaces.

   The Generic titlebar-separator provides some visual differentiation between the lined background and the black of the menu:

Figure 3.16. The Titlebar Separator

in which the white area is in fact transparent, letting the repeating pattern show through (Example 3.3, “Reworking the Titlebar Separator”.

Example 3.3. Reworking the Titlebar Separator

```
#titlebar #titlebar-separator {
    width: 153px;
    background-image:
    url("chrome://browser/skin/titlebar-separator.png");
    background-repeat: no-repeat;
                }
```

With the Rustic theme, on the other hand, the differentiation provided by this element is minimal. Instead, the wood pattern background is used to provide a more consistent theme (???):

4. The titlebar-menu-stack. A XUL stack is a container in which multiple items are stacked one on top of the next. In the Generic theme, the menu-stack holds all of the menu elements in a horizontal box, with CSS defining the characteristics of the stack contents. The initial entry for Generic assumes simply a black color background with border lines defined along the bottom of the component (see Example 3.4, “Setting the Titlebar Menu Font”

Example 3.4. Setting the Titlebar Menu Font

```
#titlebar #titlebar-title {
    font-family: "Trebuchet MS", Arial, sans-serif;
    font-weight: bold;
    font-size: 10pt;
    color:#026b86;
    margin: 2px 4px 0px 5px;
    }
#titlebar .titlebar-menu-stack{
    background: black;
    border-bottom: 2px solid;
    -moz-border-bottom-colors: #dff5fb #303030;
                }
```

Unlike previous entries, this one will need to be revised slightly in order to eliminate the flat-black background and the borders, and in order to change the fonts. One benefit that dealing with full applications provides is the ability to bundle fonts in with the rest of a theme, making it possible to perform font effects (and incorporate very non-standard fonts) in ways that are much more problematic when dealing with traditional web applications. In order to do so, you should copy the font-resource (such as Comic Sans MS, used here) into the theme jar file in the same folder as before, then at the top of the browser.css document you should include

@font-face {font-family:"Comic Sans MS"; src:url(chrome://browser/skin/comic.ttf); }

You can then reference this within the menu theme by rewriting the first titlebar entry, as shown in Example 3.5, “Invoking Fonts from Styles”:

Example 3.5. Invoking Fonts from Styles

```
#titlebar #titlebar-title {
    font-family: "Comic Sans MS", Arial, sans-serif;
    font-weight: normal;
    font-size: 10pt;
    color: #026b86;
    margin: 2px 4px 0px 5px;
                }
```

The stack itself similarly will need to be changed to be made transparent, and eliminate the very subtle borders, as shown in Example 3.6, “Working with the menu stack.”:

Example 3.6. Working with the menu stack.

```
#titlebar .titlebar-menu-stack {
    background:transparent;
                }
```

5. The window controls. The controls at the far right of the Generic display that provide support for maximizing, minimizing and closing the window can similarly be modifed. The #window-controls container, a box which holds the three buttons, can be modified with the #window-controls refer-

ence in CSS, changing it to

#window-controls { -moz-box-align: center; background: transparent; padding: 0px 0px 0px 4px; }

in order to make the starfield apear behind these elements. Note that each button itself is a GIF image with its own transparency layer, and these individual buttons will need to be modified by hand to assure that the backgrounds *are* transparent.

It should be noted in passing that these buttons (like most buttons within a theme) have multiple states depending upon whether the mouse is rolled over or pressed on an item, or if the item is disabled. The minimize button provides a good case in point here. The CSS for the minimize button looks something Example 3.7, “Assigning CSS hover states.”:

Example 3.7. Assigning CSS hover states.

```
#minimize-button {
    height: 22px;
    padding-right: 2px !important;
    padding-top: 4px !important;
    padding-bottom: 4px !important;
    list-style-image:url("chrome://global/skin/icons/
        window-control-minimize.png");
    }
#minimize-button:hover {
     list-style-image:url("chrome://global/skin/icons/
        window-control-minimize-h.png");
                }
```

For XUL <button> elements, the list-style-image property controls the image associated with the button itself, and as with fonts, this property uses the Mozilla chrome: convention for retrieving resources out of themes. The hover pseudo-property (as in #minimize-button:hover) automatically gets invoked whenever the mouse moves over the icon itself, and this can be used to provide for complex button and related behavior without getting wrapped up in Javascript code.

A full theme involves several dozen of these modications, but most of them ultimately are reasonably simple - replacement of colors, graphics or fonts, changes in spacing or padding, and so on. More sophisticated changes can be accomplished via XBL, but this is outside the scope of this article.

Note

Once you're done designing your theme, take a screenshot of the browser using the newly designed theme and save it. This will come in handy when creating an installer, as shown in the next section.

Installing Themes

While modifying existing themes is useful from a learning standpoint, the ultimate goal in changing those themes is to create stand-alone themes that can actually be distributed. This is not, in general, a difficult proposition, though there are a few niggling details that need to be paid attention to closely.

In order to create a unique theme, it's necessary to generate an install.rdf file that provides critical information in identifying the theme and contents.rdf file which holds a manifest of the contents. As it turns out, these files do not reside in the themes that ship with Netscape, so need to be added by hand.

Note

Now is a good time to rename the Generic.jar file you were working with earlier to Rustic.jar, and to copy the old Generic.jar file back into the chrome directory and restore its name to Generic.jar if you saved it under a different name.

The contents.rdf file contains links to the various components within the theme itself, and is shown in Example 3.8, “The Rustic contents.rdf file”. This particular contents.rdf would be used to identify the Rustic them built earlier.

Example 3.8. The Rustic contents.rdf file

```
<RDF:RDF xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:chrome="http://www.mozilla.org/rdf/chrome#">
  <!-- List all the skins being supplied by this theme -->
  <RDF:Seq about="urn:mozilla:skin:root">
    <RDF:li resource="urn:mozilla:skin:Rustic/1.0" />
  </RDF:Seq>
  <RDF:Description about="urn:mozilla:skin:Rustic/1.0"
        chrome:displayName="Rustic"
        chrome:author="Netscape"
        chrome:description="A theme of old growth and log cabins"
        chrome:name="Rustic/1.0"
        chrome:image="preview.png">
    <chrome:packages>
      <RDF:Seq about="urn:mozilla:skin:Rustic/1.0:packages">
        <RDF:li resource="urn:mozilla:skin:Rustic/1.0:browser"/>
        <RDF:li resource="urn:mozilla:skin:Rustic/1.0:communicator"/>
        <RDF:li resource="urn:mozilla:skin:Rustic/1.0:global"/>
        <RDF:li resource="urn:mozilla:skin:Rustic/1.0:mozapps"/>
      </RDF:Seq>
    </chrome:packages>
  </RDF:Description>
```

The contents.rdf file identifies a particular URN (Universal Resource Name) called `urn:Mozilla:skin:Rustic` that's associated with the name Rustic. The subsequent resources (global,browser,mozapps, and help) correspond to the chrome entries in somewhat inverted fashion ... that is to say, `chrome://browser/skin/myResource.png` with the Rustic skin selected tells the browser to look in the Rustic jar file in the browser/skin folder to find the resource. This file should be added to the root of the Rustic.jar file (see ???

Example 3.9. The Rustic Theme install.rdf file

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:em="http://www.mozilla.org/2004/em-rdf#">
  <Description about="urn:mozilla:install-manifest">
    <em:id>{99cd1e10-dd02-11d9-8cd5-0800200c9a66}</em:id>
    <em:version>1.0</em:version>

<!-- Target Application this extension can install into,
         with minimum and maximum supported versions. -->
   <em:targetApplication>
    <Description>
      <em:id>{3db10fab-e461-4c80-8b97-957ad5f8ea47}</em:id>
      <em:minVersion>8.0</em:minVersion>
      <em:maxVersion>10.0</em:maxVersion>
    </Description>
  </em:targetApplication>

    <!-- Front End MetaData -->
    <em:name>Rustic Theme</em:name>
    <em:description>A theme evoking old growth forests and log cabins.</em:descrip
    <em:creator>Netscape</em:creator>
    <em:contributor>Netscape contributers</em:contributor>

    <!-- Front End Integration Hooks (used by Theme Manager)-->
    <em:internalName>Rustic/1.0</em:internalName>
  </Description>
</RDF>
```

Save this file as `install.rdf` in the same root directory for the Rustic.jar file as the contents.rdf file. You're nearly done at this stage ... all you will need to do is to create a preview graphic. Take the screenshot of your themed browser (you did take a screenshot from last section, right?) and trim it down to 230x230 pixels in size. Save the file out as a PNG file called `preview.png`, then save this back to the Rustic.jar file. This is the preview image that is displayed for the Rustic them in the Themes dialog when the latter pops up. This image doesn't *have* to be a screenshot, of course, but it is probably not that useful for it to be anything else.

Congratulations, you have created a new theme! Now you just have to install it on your system to insure that it in fact does work correctly. Fortunately, this stage is easy:

1. Using the File->Open File menu item, navigate to the Rustic.jar file and open it.The installer should recognize the JAR file as a theme installer, and will launch the Theme dialog.
2. Select the Rustic theme.
3. Press the Use Theme button.
4. Close all instances of the Netscape browser and restart the browser. Your theme should now be working.

Customizing the Netscape browser offers the option not just of expressing one's inner artistic skills, but can make it possible to configure the browser for secondary branding and the addition of custom capabilities tied specific to a certain theme (a case where themes and extensions overlap).

Appendix A. Licenses

This appendix will contain the various licenses for use within the Netscape theme package.