

Capítulo 7: Personalização da Aparência do Plone

Trad.: José Marcelo Miranda e Michael Almeida Lucas de Souza

Nos dois capítulos anteriores foram mostrados alguns componentes principais da interface do usuário Plone, incluindo objetos Script (Python) e page templates. Está na hora de mostrar como construir a aparência de um site Plone. Este capítulo inclui os objetos dos capítulos anteriores e serão introduzidos mais alguns.

Para começar, mostrarei as definições chaves e os elementos do Plone que compõem um site. Irei definir termos que você já deve ter ouvido falar, tal como **skins** e **layers**. Será mostrado como personalizar a interface do usuário Plone, utilizando o CSS - Cascading Style Sheets. Passarei pelas variáveis chaves e irei mostrar como modificá-las, também mostrarei novamente os ajustes dos logos e elementos de skins, assim juntando os tópicos eu cobrirei momentaneamente os três capítulos anteriores.

Então, mostrarei como fazer um novo skin e explicarei as técnicas para desenvolver no (file system) sistema de arquivos.

Finalmente, terminarei este capítulo com um exemplo de um site. Especificamente, mostrarei o Maestro site, que foi utilizado pela NASA para distribuir dados sobre Mars rovers. Este é o Web site de maior trafego construído em Plone, e o skin fornece uma excelente ferramenta de estudo para personalizar um site. Este é um exemplo real de como você pode personalizar e modificar um Plone site e fazer o mesmo com o seu site.

Introdução ao Plone Skins

Quando um documento aparece no Plone, o conteúdo deste documento mostrado é familiar à interface verde e azul do Plone. O **skin** determina como que o documento é mostrado para o usuário, incluindo as imagens e os estilos que cercam o conteúdo. Os elementos dos grupos de skin envolvem esta parte do conteúdo, e as apresenta de uma maneira clara.

Para gerar a representação que o usuário vê, um skin tem vários elementos, incluindo os itens estáticos, tal como as imagens, e as partes dinâmicas, tal como scripts. No capítulo anterior, o formulário feedback é um exemplo de como adicionar alguns elementos para um skin e criar novos elementos de um skin. Este exemplo contém a lógica do formato de um objeto Python (Script) e das novas páginas no formato de page templates. Todos estes elementos do skin deverão interagir com o formulário.

Você pode utilizar partes de um skin já existente para montar um novo skin. Você pode fazer um skin com poucas ou grandes mudanças, como você pode ver nos sites da comunidade tal como **http://www.zopezen.org** e **http://www.zopera.org**. Cada Plone site deve ter no mínimo um skin, o qual será utilizado como padrão, mas pode ter vários skins. Um usuário pode escolher opcionalmente entre skins, se o desenvolvedor do site assim o permitir, embora seja muito raro isto acontecer.

O skin padrão do Plone é o que você vê em um site Plone, tal como em

`*http://www.plone.org*`, com a interface azul ou verde. Mas o Plone não tem que ser visto sempre assim, ou ser reconhecido como um site Plone. Pegue, por exemplo, a lista de sites disponíveis em `*http://www.plone.org/about/sites*`; cada um desses sites fornece experiências habituais e diferentes para o usuário. Na maioria dos casos, estes sites podem facilmente mudar de skins e fornecer uma aparência diferente para o usuário. Outros sites utilizam o poder e a flexibilidade da interface do Plone para criar e editar conteúdo enquanto fornecem uma aparência totalmente diferente para o usuário externo.

Já vi várias perguntas na lista de e-mail tais como, o Plone tem que se parecer com o Plone site? Ele pode parecer de um jeito para um usuário e de outra forma para outro usuário? Pode se parecer com meu site corporativo? A resposta para todas estas perguntas é "`*sim*`": o único limite será a sua imaginação (e o tempo que você pode gastar personalizando seu site).

Utilizar Camadas Dentro de um Skin

Um skin é dividido em um conjunto de lógica, de templates e scripts chamados `*layers*`. Alterando este conjunto um por um, permite o usuário facilmente adicionar componentes para um skin ou remove-los. Layers (camadas) são representadas em um skin por uma lista hierárquica de pastas. Cada camada combina com o nome da pasta, e cada pasta contém os elementos de skin.

Por exemplo, um skin pode ter as seguintes camadas:

`custom, gruf, plone_ecmascript, plone_wysiwyg ...`

A ordem das camadas na lista é o fator chave para o Plone encontrar os elementos. Quando um elemento, tal como `*logo.jpg*`, é pedido pelo skin, o skin procura pelas camadas para encontrar o elemento. O skin começa procurando na primeira camada determinada para o skin (neste exemplo, `*custom*`). Se o skin não encontra o elemento na primeira camada, ele vai para a segunda camada (no exemplo, `*gruf*`). E continua procurando através da lista de camadas até encontrar o elemento solicitado. Se não for encontrado nenhum elemento, então o erro 404 é chamado e retornado para o navegador.

Um conceito parecido é o uso da variável de ambiente PATH na maioria dos sistemas. Quando é digitado um comando ou a busca por um programa executável, o sistema operacional procura através dos diretórios no sistema de arquivos como especificado na variável de ambiente PATH.

Ao permitir que camadas superiores tenham preferência sobre camadas inferiores, os desenvolvedores e administradores tem a possibilidade de personalizar e manipular seus sites através dessas camadas. Se você não gosta de alguns elementos em particular de um skin Plone, então poderá movê-lo entre as camadas e adaptar o resultado. Você pode ordenar seus skins e layers no Plone com a ferramenta `*portal_skins*`.

Gerenciar os Skins com a Ferramenta `portal_skins`

Você utiliza a ferramenta `*portal_skins*` no Plone para definir o skin e o comportamento das camadas. Esta ferramenta também fornece serviço e um API - application programming interface para criação e utilização dos skins.

Para acessar a ferramenta **portal_skins**, vá para ZMI - Zope Management Interface e clique em **portal_skins**. Você verá duas telas chaves na ZMI; a primeira, a aba Contents (conteúdo), que mostra todas as pastas e diretórios do sistema de arquivo (FSDVs) localizado nesta ferramenta (veja Figura 7-1).



Figura 7-1. O conteúdo da ferramenta **portal_skins** em uma instalação padrão do Plone

Todas as pastas e visões dos diretórios do sistema de arquivo na aba Contents não são camadas por definição, mas agora você pode transformá-las em camadas. Além disto, a segunda tela, a aba Properties, mostra todos os skins e camadas que você definiu no seu Plone site (veja Figura 7-2).

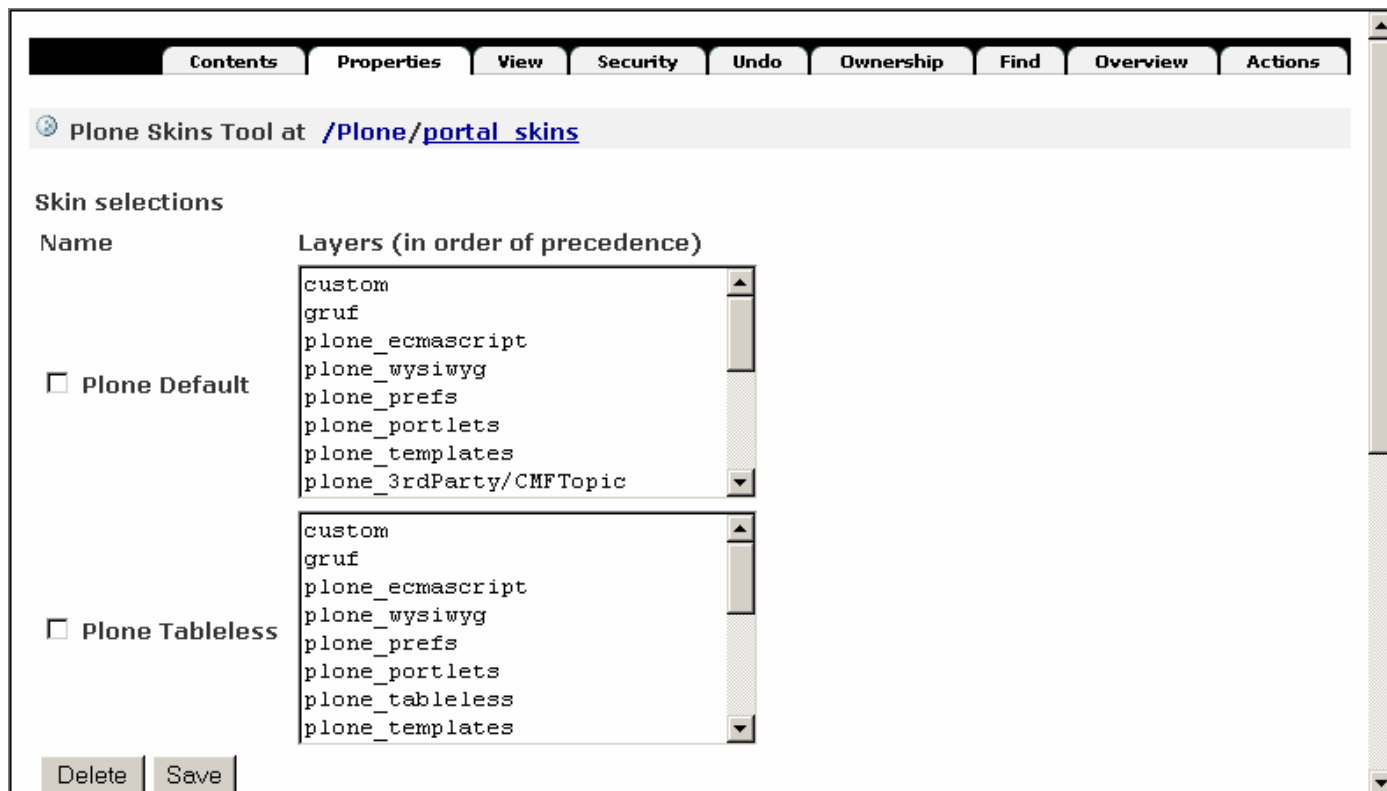


Figura 7-2. Os skins e camadas na instalação padrão do Plone

Existem um grande número de camadas que fornecem ao desenvolvedor um alto grau de flexibilidade e reutilização. Cada skin mostra na esquerda, uma área de texto contendo a direita todas as camadas dentro desse skin. Como já mencionado, para localizar os elementos, o Plone procura as camadas pelo topo para o final. Cada camada representa o nome da pasta ou FSDV da aba Contents. Na Figura 7-2, você pode ver o diretório *plone_ecmascript*, e na Figura 7-1 você pode ver o objeto combinando o FSDV.

Um FSDV é um novo objeto muito útil do Plone; ele permite acesso direto para os elementos do skin que são definidos no sistema de arquivo ao invés do armazenamento no banco de dados do ZOPE como regularmente acontece. Através do FSDVs fica muito mais fácil o desenvolvimento e ajustes, lendo objetos diretamente do sistema de arquivos pode-se escrever e editar o código necessário para produzir o site. Quando você instala o Plone, o skin é escrito sobre o sistema de arquivo. Quando você ajusta um objeto, você faz uma copia local dentro do banco de dados do Plone. A utilização do FSDV possibilita manter a separação clara entre o código baixado da Web e o código ajustado na instância da raiz.

Plone 2 vem com dois skins, Plone Default e o Plone Tableless. Plone Default utiliza tabelas (<table>) para fazer o corpo principal da página (main body) flanqueados por outras células uma de cada lado, contendo slots de conteúdo. Para compatibilidade dos diversos tipos de navegador, esta é uma configuração padrão. Entretanto, se você trocar para o Plone Tableless, então você vai adquirir um skin com a mesma aparência, sem a utilização da tag *<TABLE>* para produzir a página, o qual lhe dá mais flexibilidade, como desenvolvedor. Na época de seu desenvolvimento, o skin Plone Tableless pode ter problemas com alguns navegadores, tal como o Internet Explorer.

Para mudar o skin, vá ao botão do formulário onde está o valor Default Skin e selecione o skin default da lista de opções. Se você selecionar a opção Skin Flexibility, então os usuários serão capazes de escolher seu próprio skin da seção **my preferences**.

Retornando para aba Contents da ferramenta **portal_skins**, você pode ver que algumas pastas, como a **custom**, são definições que existem no Zope. Outras por exemplo, **plone_images** são FSDVs que apontam para áreas do sistema de arquivo. Estas possuem o ícone da pasta com uma fechadura verde dentro dela, indicando que você não pode adicionar ou editar elementos em um FSDV através da Web; você pode fazer isto através do sistema de arquivo.

Para ver onde os arquivos se encontram em seu disco rígido, clique na aba Properties do FSDV. Por exemplo, pela aba Contents da ferramenta **portal_skins**, clique Properties, e irá listar os caminhos do sistema de arquivos de **CMFPlone/skins/plone_images**. Este caminho é a localização do diretório no sistema de arquivo relativo para a instância home que você especificou no processo de instalação.

Ajustar Skins

Você viu como os skins e camadas interagem. Agora verá como ajustar o Plone site. Retornando ao exemplo do Capítulo 4, onde você aprendeu como personalizar o logo. Utilizando seu novo conhecimento de como os skins funcionam, você será capaz de acompanhar adiante e personalizar o skin. Assim mostrarei o poder do CSS no Plone e como você pode personalizá-lo. Finalmente, mostrarei o main template visto em capítulos anteriores e falarei de cada elemento dele.

Personalizar o Logo (Revisando)

No Capítulo 4 você aprendeu como ajustar o logo no canto do topo esquerdo do Plone site, mas foi na oportunidade foi omitido o que realmente acontece. Esta seção vai revisar este exemplo.

A imagem **logo.jpg** é a que aparece no canto do topo esquerdo de toda página. Você agora irá descobrir o que acontece quando o navegador tenta criar a página. Uma vez que o Plone recebe o pedido para esta imagem, o Plone procura através das camadas tentando encontrar o **logo.jpg**. Na definição do site, este é o elemento no **plone_images**, chamado **logo.jpg**. Por ser um FSDV, como mencionado anteriormente, você será incapaz de alterá-lo através da Web. Para controlar seu site de futuras mudanças, você não é capaz de mudar isto no sistema de arquivos também. Ao invés disso, observe bem o que este botão **Customize** é capaz de fazer, você pode ver que a esquerda do botão, existe uma lista drop-down de contendo pastas da ferramenta **portal_skins**.

****NOTA**** As pastas listadas são as que existem dentro do banco de dados do Zope. Os FSDVs não estão incluídos na lista drop-down; por padrão, eles mostram somente as pastas.

Clique no botão Customize para criar uma cópia local do item na pasta selecionada no box drop-down. Por definição esta pasta é a **custom**, então agora você tem a copia da pasta **custom**. Quando o Plone procura um elemento, **logo.jpg**, irá acessar a versão da pasta **custom**, porque esta é a pasta da

camada superior do skin. Então, quando o `*logo.jpg*` é chamado, irá encontrar a imagem na camada `*custom*`. Isto explica o porque da criação do novo `*logo.jpg*`.

O modo mais rápido para iniciar o seu Plone site, seria colocando todos os elementos personalizados na pasta `*custom*`. A pasta `custom` é pasta padrão do Plone, então colocando seus itens ali você garante também que eles sejam encontrados de forma mais rápida.

Introdução ao Plone CSS - Cascading Style Sheets

A representação visual do Plone site no navegador é toda baseada na utilização do CSS. Talvez o modo mais fácil de ver exatamente como o CSS atua em um Plone site é comparar as Figuras 7-3 e 7-4. A primeira mostra o Plone com CSS, e a segunda sem nenhum CSS.

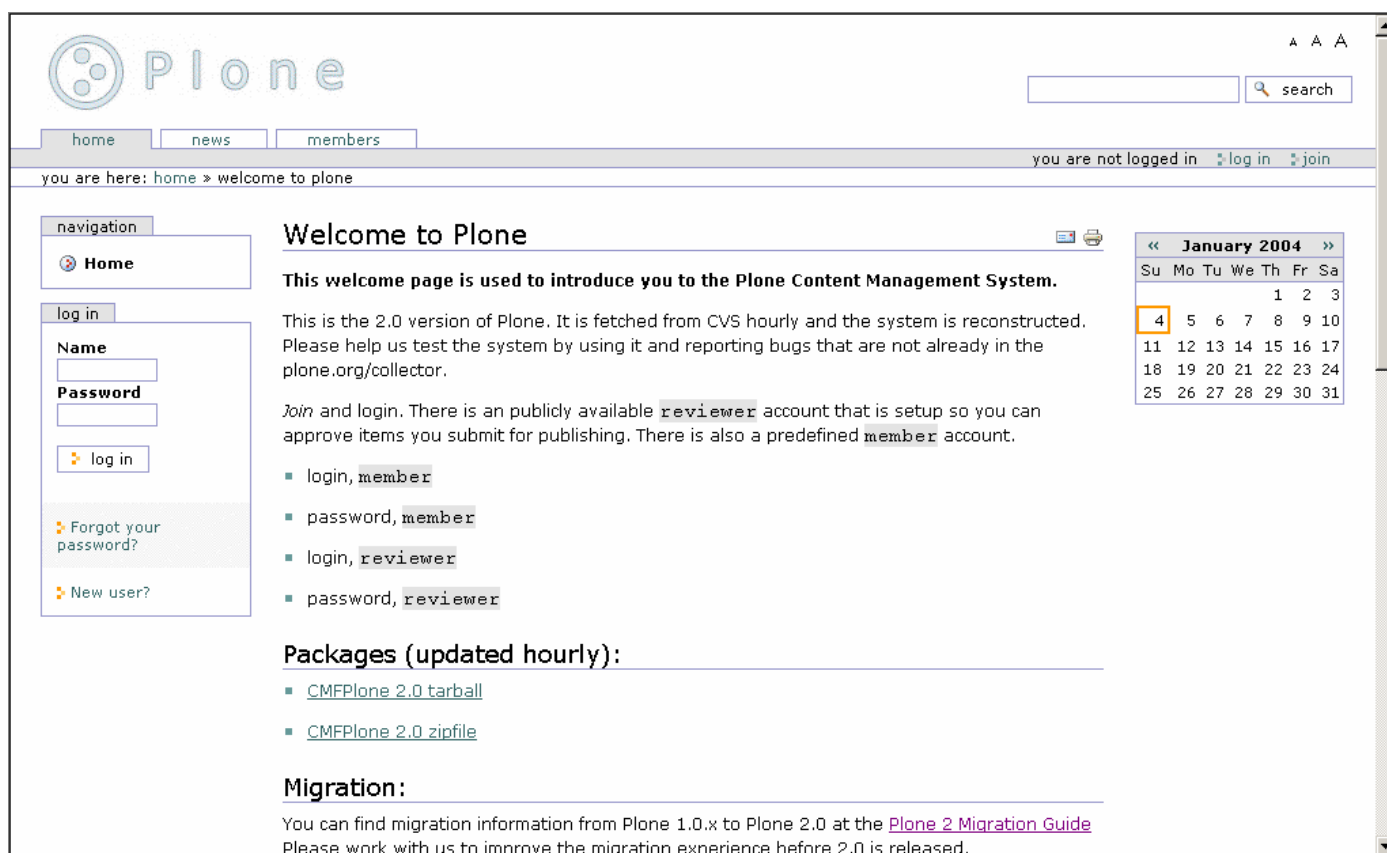


Figura 7-3. Plone com CSS

[Skip to content.](#)

Portal

Search

Sections

- ◆ [Home](#)
- ◆ [News](#)
- ◆ [Members](#)

Personal tools

- ◆ You are not logged in
- ◆ [Log in](#)
- ◆ [Join](#)

You are here: [Home](#) » Welcome to Plone

Welcome to Plone



This welcome page is used to introduce you to the Plone Content Management System.

This is the 2.0 version of Plone. It is fetched from CVS hourly and the system is reconstructed. Please help us test the system by using it and reporting bugs that are

Figura 7-4. Plone sem CSS

****TIP**** Se você quer reproduzir o site sem o CSS, desligue o CSS em seu navegador. O Internet Explorer não permite que você faça isto facilmente, mas o Firefox ([*http://www.mozilla.org/products/firefox/*](http://www.mozilla.org/products/firefox/)), o Mozilla é um navegador baseado em open-source(código aberto), permite isto facilmente. No Firefox, selecione Tools - Web Developer - Disable - Disable Styles. Firefox é o navegador da maioria dos desenvolvedores Plone.

A diferença é impressionante, o CSS fornece não apenas a representação visual das páginas, mas também o layout. Mudando o CSS, você pode mudar a representação visual e o layout no Plone site (dentro das restrições do CSS).

Esta impressionante performance e utilização do CSS na apresentação do Plone se deve a talentosos usuários/desenvolvedores de interface, e trazem os seguintes benefícios:

- O CSS fornece uma camada separada da apresentação e os templates é que irão gerar a apresentação.
- Você pode fazer um grande número de mudanças sem ter que alterar os templates. Tudo que é preciso é ser um conhecedor de CSS.
- CSS faz com que o site fique mais rápido enviando arquivos menores. Cada arquivo HTML - Hypertext Markup Language é menor desde que o layout do site não estejam contido no código HTML mas somente em arquivos CSS que pode ser cacheado.
- CSS possibilita a customização da aparência (“look and feel”) sem alterar todo o trabalho de acessibilidade já implementado nos sites Plone.

Code Layers

As 3 últimas camadas de código são as responsáveis pela renderização (montagem) de uma página de um site Plone. Por exemplo, as abas que aparecem de lado a lado no topo do Plone site, isto é como eles são montados:

pergunte a si mesmo como posso customizar estas tabs? Você precisa considerar que customizações você gostaria de implementar. Isto significará ter que mudar o CSS, o HTML, o Dado ou underlying tabs. Via de regra é assim que acontece:

De fato, o Plone é personalizável, em vários níveis. Para ter certeza, quando no futuro houver mudanças nos templates do Plone, que estas alterações não comprometam o design da aplicação, tente não ajustar/alterar os templates. Ao invés, recomendo a você tentar modificar primeiro o CSS ou as *actions*. Deste modo você poderá ter menos dor de cabeça, quando houver mudanças dos templates em futuras versões do Plone.

Ajustar Fonte, Cores, e Espaço

A folhas de estilo que faz a maioria desse trabalho é a *plone.css*. Utilizando DTML - Dynamic HTML ela possui um grande número de variáveis. Não foi explicado DTML neste livro, porque isto é provavelmente sua única utilização dentro do Plone. O sistema Zope Page Templates já lhe fornece qualquer coisa que você precisar. Excelentes referências on-line de DTML para Zope, entretanto; refere-se para *http://zope.org/Documentation/Books/ZopeBook/2_6Edition/DTML.stx*.

A sintaxe DTML para esta folha de estilo é muito simples; cada variável se relaciona a um atributo correspondente na propriedade da folha, para acessar clique em *portal_skins*, *plone_styles*, e então *base_properties*. Na Figura 7-5, você pode ver como esse arquivo é visto na ZMI.

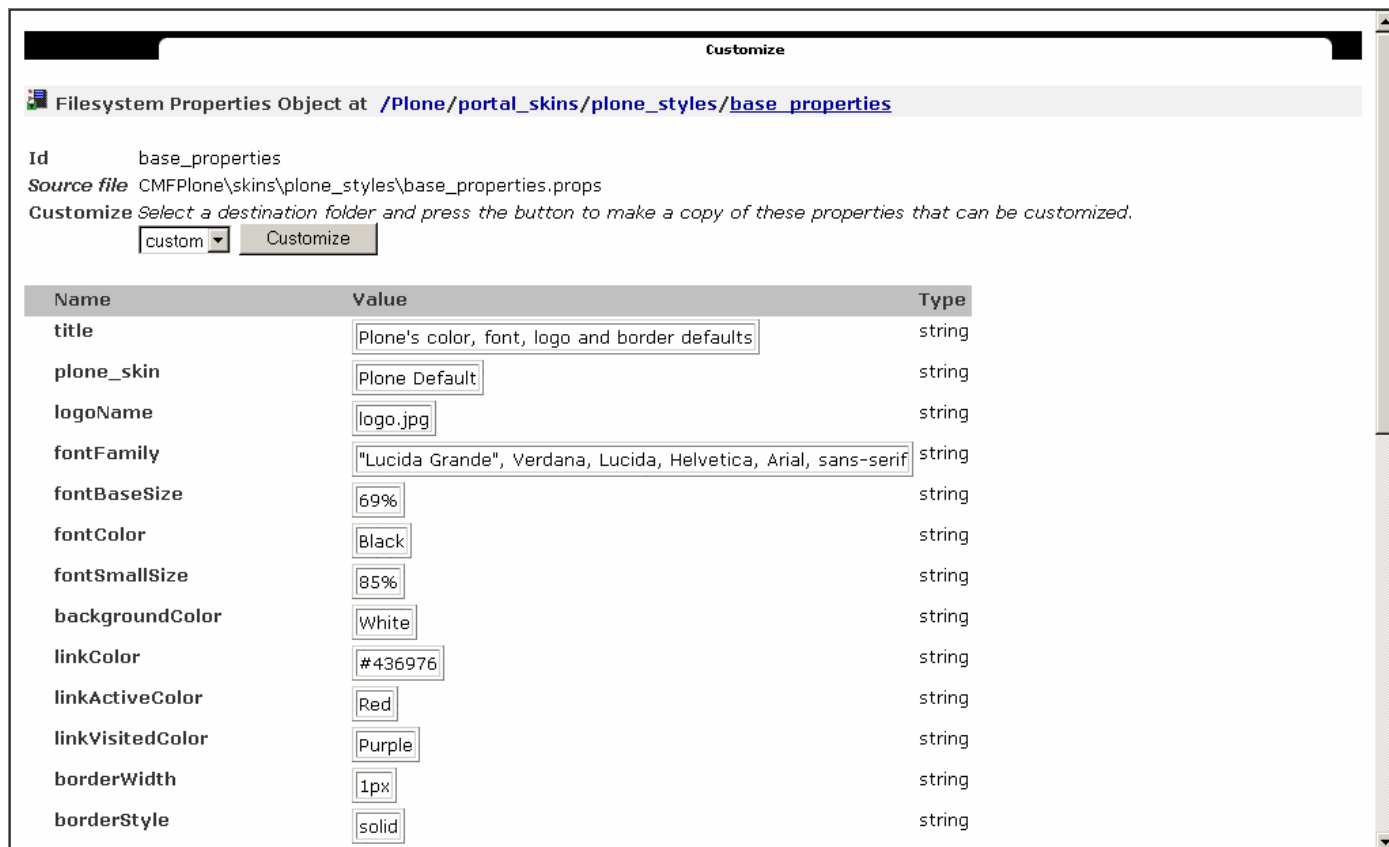


Figura 7-5. A propriedade base para a folha de estilo

Por exemplo, `*dtml-fontColor*` aloca a variável `*fontColor*` (preta) e a coloca na folha de estilo. Agora você pode ver onde está variável é referenciada no arquivo `*plone.css*`. Para acessar o arquivo CSS, clique `*portal_skins*`, `*portal_skins*`, e então `*plone.css*`. Neste arquivo você pode ver que `*mainFontColor*` é referenciado em vários lugares; por exemplo, isto é referenciado na main body da página, deste modo:

```
body {
  font: dtml-fontBaseSize; <dtml-var fontFamily>;
  background-color: dtml-backgroundColor;;

  color: dtml-fontColor;;

  margin: 0;
  padding: 0;
}
```

Você pode continuar lendo através da folha de estilo, mas mudar a variável é sempre o modo mais rápido para ver o que exatamente isto afeta.

Retornando a ZMI, clique em `*portal_skins*`, `*plone_styles*`, `*base_properties*` e então clique no botão `*Customize*`. Como já visto (com o exemplo do logo.jpg), este procedimento irá criar um objeto na ZMI que você pode ser personalizável. Neste momento o objeto personalizado é na verdade uma pasta. Para acessar as propriedades que você personalizou, clique `*portal_skins*`, `*custom*` e então clique `*base_properties*`. Depois, selecione a aba `*Properties*` (veja Figura 7-6).

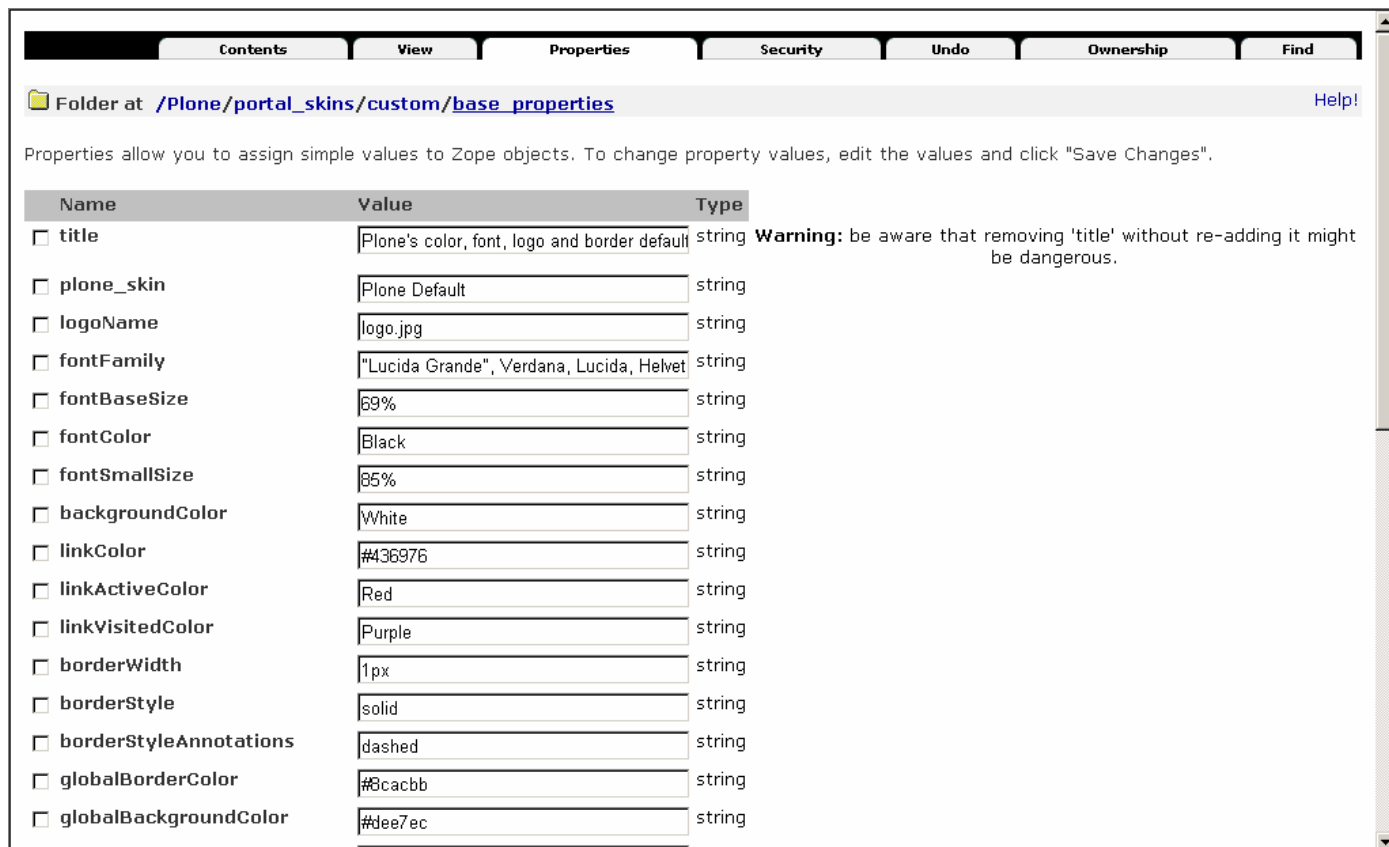


Figura 7-6. Propriedades da pasta

Esta lista de propriedades permite a você fazer mudanças do `*mainColor*` para algo diferente, por exemplo, `*red*` ou `*#cc9900*`. Mude o valor desta propriedade, e clique Save Changes. Retornando ao Plone site, você deve ver agora a nova cor.

No capítulo 4 você viu um exemplo onde, para mudar a aba no topo da página, os usuários podem mudar as ações. Embora você possa digitar a ação com o primeiro caractere em maiúsculo (tal como `*Members*`), ele será mostrado em letras minúsculas na página Web. Isto porque o CSS transforma em minúsculo por causa do `*textTransform*` na propriedade da folha. Para parar esta transformação, mude a propriedade de `*textTransform*` para `*none*`.

Na folha de estilo, as propriedades são definidas para todas as cores, espaçamentos, e fontes que são usadas no Plone site. Tabela 7-1 descreve todos os parâmetros.

Tabela 7-1. Propriedades CSS

Variable Name	Descrição
logoName	O nome do arquivo do portal logo
fontFamily	Fonte family usada para todo texto que não é cabeçalho
fontBaseSize	The base font size from which everything is calculated
fontColor	Cor da fonte principal

background-color Cor do background

link-color Cor utilizada em links normais

link-active-color Cor utilizada em links ativos

link-visited-color Cor utilizada em links visitados

border-width Largura da maioria das bordas no Plone

border-style Estilo das linhas da borda

border-style-annotations Estilo da linha da borda nos comentários

global-border-color Cor da borda usada na aba principal, portlets etc...

global-background-color Cor do background para abas selecionadas, portlet headings etc...

global-font-color Cor da fonte nas abas e nos portlet headings

heading-font-family Fontes para título *h1*, *h2*, *h3*, *h4*, *h5* e *h6*

heading-font-base-size Tamanho usado quando calcula diferentes tamanhos de títulos

content-view-border-color Cor da borda da tabela na aba de conteúdos

content-view-background-color Cor do background na aba Contents

content-view-font-color Cor da fonte usada na tabela na aba de Conteúdos

text-transform Whether to lowercase text in portlets, tabs, and so on

even-row-background-color A cor do background das linhas nas listas

odd-row-background-color A cor do background das linhas nas listas

notify-border-color A cor da borda de elementos de notificação ex: mensagem de status e o calendário

notify-background-color Cor do background dos elementos de notificação

help-background-color Cor do background do pop-up de calendário

discreet-color The color of the credits, document byline, form help

portal-min-width A mínima largura do portal

column-one-width A largura da coluna a esquerda

column-two-width A largura da coluna a direita
=====

Personalizar o CSS

Se você possui pequenas personalizações, coloque-as em `*ploneCustom.css*`. Esta é uma segunda folha de estilo que é carregada após `*plone.css*`. Utilizando funcionalidades de estilo em cascata, você pode aplicar quaisquer mudanças para o `*ploneCustom.css*` e estas afetarão todas as demais folhas de estilo.

Por exemplo, para mudar o byline que aparece na base de toda página, simplesmente mude o `*ploneCustom.css*`. Novamente, acesse este arquivo através da ZMI, e então clique em Customize. Isto irá criar uma cópia da folha de estilo na pasta `*custom*`. Para alterar o byline, mova para lado esquerdo da página e coloque bold (negrito), como mostra na Figura 7-7.

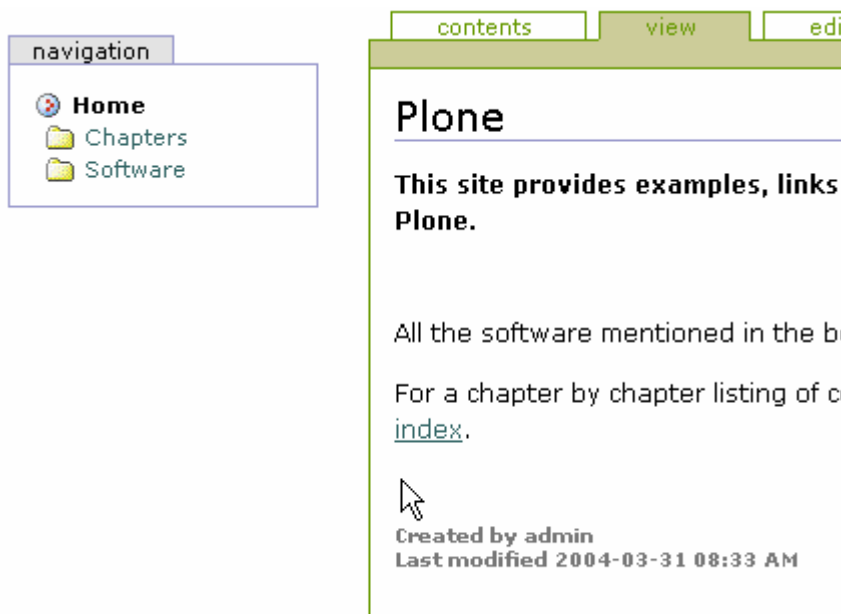


Figura 7-7. Novo byline em bold(negrito) na esquerda

Você faz isto adicionando o seguinte:

```
div.documentByLine {
    text-align: left;
    font-weight: bold;
}
```

Aqui você ajustou dois atributos para o elemento `*byline*`: `*text-align*` e `*font-weight*`. Observe que você não mudou qualquer outro atributo do elemento `*byline*`; os atributos que faltarem serão herdados pela folha de estilo original. Com poucas e simples linhas de CSS, você muda o site e está certo de que futuras versão do Plone não afetarão o seu site. Mudar o `*ploneCustom.css*` é o melhor modo de fazer pequenas mudanças.

Utilizando diferentes folhas de estilos, você pode usar o Plone para fornecer diferentes visões para diferentes clientes. Frequentemente Web sites tem o botão pra imprimir a página que mostra a página simples preparada para ser impressa, sem muita formatação. O Plone suavizou este problema fornecendo um estilo de folha separado; quando um navegador imprimir a página, este estilo de folha formatará a página. Todo estilo de folha alternativos estão incluídos no

topo da pagina; você pode encontrá-los clicando em *portal_skins*, depois em *plone_templates*, e então clicando em *header.pt*.

****NOTA**** Uma folha de estilo incomum é a *projection*. Ele é suportado apenas pelo Opera, e é usado quando o navegador se encontra no modo de tela cheia.

Personalizando o Main Template

Como você viu no capítulo anterior, para ter a aparência Plone em uma pagina, você precisa utilizar o *master* macro do *main_template*. Toda página Plone page utiliza esta macro que preenche os slots apropriados. Observando o main template em detalhes, você pode ver como uma página Plone é construída e ver exatamente como você pode personalizar os elementos da página.

Se você olhar na página main Plone, você verá o número de elementos na página. Figura 7-8 mostra a página Plone com todos os elementos da interface do usuário marcados. Tabela 7-2 descreve cada um dos elementos e suas finalidades. Para cada elemento na Figura 7-8, você encontrara o número correspondente na tabela.

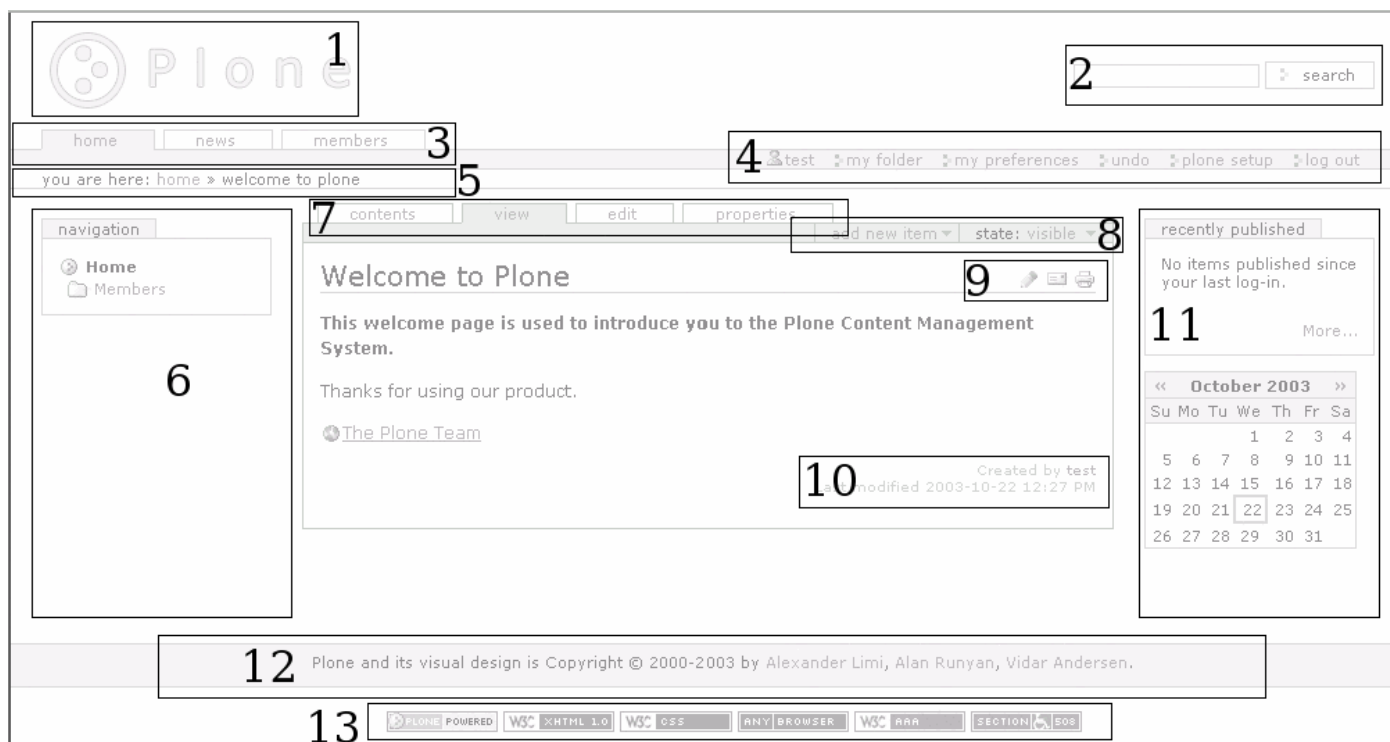


Figura 7-8. Todos os elementos principais na interface do usuário no Plone

Tabela 7-2. Elementos da Interface do Usuário.

=====
No. Nome Descrição
=====

1 Site logo Mostra o logo do topo.

2 Search form Mostra o formulário de pesquisa.

3 Portal tabs Mostra as abas acima do topo do site.

 4 Personal bar Mostra a informação pessoal para o usuário tal como `*login*` e `*pasta particular*`.

5 Breadcrumbs Mostra a localização do conteúdo atual.

6 Left slot Determina onde o portlet adicionado deve ser mostrado no caso `*left_slot*`.

7 Content tabs Mostra as ações com a categoria `*content_tabs*` para esta parte de conteúdo.

8 Content drop-down lists Mostra menus drop-down para o conteúdo, workflow e novos tipos de conteúdos.

9 Document actions Mostra as ações para esta parte particular de conteúdo: imprimindo ou e-mail.

10 Byline Mostra a descrição do conteúdo e seu autor.

11 Right slot Determina onde o portlet adicionado deve ser mostrado no caso `*right_slot*`.

12 Footer Mostra a informação na base da página.

13 Colophon Mostra mais informações abaixo da base da página.
 =====

Não foi explicada nenhuma seção de template: de conteúdo. Todo o texto de `*Welcome to Plone*` abaixo para `*The Plone Team*` é conteúdo adicionado e editado por usuários. Este é o `*slot*` principal na página template, o qual é preenchido por um tipo de conteúdo ou page template. O Capítulo 6 cobre o assunto “Usando slots”; neste capítulo, mostro como, usando o `*main*` slot , você pode assegurar que o conteúdo apareça dentro da página Plone.

Assim, dado estes componentes da sua página Plone, como você pode personalizar uma parte em particular? A resposta é encontrar a parte que combina com o `*main_template*`, veja qual parte que é chamada e então personalize ela. Por esta razão, explicarei o main template em detalhes.

Olhando pela primeira vez, o main template parece ser complicado, mas a maioria são macros, e seu principal objetivo é simplesmente pegar conteúdo de outras áreas. Você pode encontrar o main template clicando em `*portal_skins*`, em `*plone_templates*`, e então `*main_template*`.

A filosofia atrás do main template é que o usuário não deve alterar a configuração atual do template, a menos que tenha a maioria das mudanças planejadas. O main template pega todo o conteúdo de lugares dentro do Plone, assim você pode alterar a montagem da página alterando individualmente cada elemento. Isto significa que você apenas altera as seções que deseja do que alterar todo o template.

O main template utiliza XML - Extensible Markup Language com o namespaces

metal quase sempre presente. Por exemplo:

```
<metal:headslot define-slot="head_slot" />
  <!-- Espaço onde você pode inserir elementos no cabeçalho do template -->
```

Aqui, o nome da tag não é um elemento padrão de XHTML - Extensible HTML; no lugar, ele utiliza o prefixo *metal:* para definir o namespace como *metal:headslot*. Isto possui as seguintes vantagens:

- O elemento *headslot* é semântico, que descreve o elemento. Fica fácil dispor isso em um slot para adicionar qualquer coisa que você queira no cabeçalho da página.
- Attributes in that element use the namespace in the element if not otherwise declared; so, instead of *metal:fill-slot*, you can just use *fill-slot*.
- A tag atual não é uma tag XHTML válida, assim não será mostrada. Entretanto, se a renderização da tag gerar qualquer XHTML válido, então este XHTML será mostrado.

Quando uma macro é utilizada, o conteúdo do template chamado é removido, assim é possível colocar comentários (texto) no template que é chamado. Por exemplo:

```
<div metal:use-macro="here/global_searchbox/macros/quick_search">
  The quicksearch box, normally placed at the top right
</div>
```

Desta maneira, é fácil determinar a que esta macro se refere: ao search box (caixa de pesquisa) no topo do canto direito do site (elemento 2 na Figura 7-8). Para ver a macro, encontre o script chamado *global_searchbox* e a macro *quick_search* contida dentro dele. O main template continua através dos *main* macros, pegando informações de diferentes templates e scripts, e construindo a página.

Após esta seção, o template determinará o conteúdo principal da página, na qual o objeto está sendo criado. No capítulo 6 foi explicada a diferença entre slot e macro; lembre-se que um template define slots que são preenchidos por conteúdos. O Slot de maior importância para o conteúdo da página é o *main* slot.

Um padrão comum no Plone e que pode confundir é a definição de um slot dentro de outro slot. Por exemplo, a definição para o *css_slot*:

```
<metal:cssslot fill-slot="css_slot">
  <!-- Um slot onde você pode inserir CSS de um template -->
  <metal:cssslot define-slot="css_slot" />
</metal:cssslot>
```

Este padrão pode parecer um pouco estranho: ele define o slot e então recria o preenchimento do slot. Se você olhar o main template cuidadosamente, aqueles slots estão na verdade dentro do cabeçalho *use-macro*, assim o header macro pode preencher este slot. Mas você também pode no fim do template preencher o slot, por esta razão o slot é redefinido. Isto significa que um slot pode ser

preenchido em dois lugares, e esta pode ser uma técnica muito útil para modificar templates.

Examinando de baixo para o resto do main template, você encontrará colunas da esquerda e da direita, o rodapé (footers) e o “colophon”. Note que a coluna da esquerda talvez apareça antes que o conteúdo principal da página, mas a folha de estilo muda isso. Isto assegura que se você visitar o site com um navegador de texto apenas, o conteúdo principal aparece primeiro, mas não após todas as opções de navegação.

Tabela 7-3 descreve as macros e slots no main template.

Tabela 7-3. Main Template Macros e Slots

****Name Description Slot or Macro?***

Cache headers Ajusta o http - Hypertext Transfer Protocol cache headers para o conteúdo. Macro: `*cacheheaders*` em `*global_cache_settings*`

Head slot Permite adicionar `*head*` element de página. Slot: `*head_slot*`

CSS slot Permite adicionar custom CSS à página. Slot: `*css_slot*`

JavaScript head slot Permite adicionar custom JavaScript para a página. Slot: `*javascript_head_slot*`

Site actions As ações do site permite você ter uma serie de ações sobre a pesquisa. Por default isto permite você mudar o tamanho da fonte. Macro: `*site_actions*` em `*global_siteactions*`

Quick search A caixa de busca rápida no topo do canto direito. Macro: `*quick_search*` em `*global_searchbox*`

A aba portal (normalmente azul) está normalmente no topo esquerdo. As abas mostradas são determinadas pelas ações. Isto determina como as abas são feitas em HTML. Macro: `*portal_tabs*` em `*global_sections*`

Personal bar A barra particular no topo direito: login, logout, e assim por diante. Macro: `*personal_bar*` em `*global_personalbar*`

Path bar O caminho breadcrumbs (trilha de pão) que inicia com "Você está aqui." Macro: `*path_bar*` em `*global_pathbar*`

Content views A aba (normalmente verde) acima do top do conteúdo. Isto irá mostrar apenas quando o conteúdo pode ser editado pelo usuário corrente. A aba atual mostrada é determinada pelas ações. Isto determina como as abas são criadas em HTML. Macro: `*content_views*` em `*global_contentviews*`

Content actions Pequena lista drop-down de ações no canto do topo a direita na barra de contexto. Macro: `*content_actions*` in `*global_contentviews*`

Portal status message Mensagem mostrada toda hora que algo é modificado. Macro: `*portal_message*` em `*global_statusmessage*`

Header O cabeçalho de um fragmento do conteúdo. Slot: `*header*`

Main A parte principal do fragmento de conteúdo. Slot: `*main*`

Sub A parte inferior de um fragmento do conteúdo onde os comentários em um objeto irão aparecer. Slot: `*sub*`

Left portlets Os slots ou portlets mostrados na esquerda da página. Possui poucas definições: `*column-one-slot*` é toda coluna à esquerda, e `*portlets-one-slot*` é então o slot. Se nenhum destes slots é definido, ele chama o macro. Macro: `*left_column*` em `*portlets_fetcher*`

Right portlets Os slots ou portlets mostrados na direita da página. Veja os portlets esquerdos. Macro: `*right_column*` em `*portlets_fetcher*`

Footer Copyright e outras mensagens. Macro: `*portal_footer*` em `*footer*`

Colophon Diferentes mensagens para a base. Macro: `*colophon*` in `*colophon*`

Com essas informações, agora o importante é personalizar a macro ou o slot para mudar a aparência da página. Novamente, não é recomendado a você personalizar o main template agora, ao invés você pode personalizar por partes que são chamados pelo main template. A próxima seção mostra alguns exemplos de personalizações que você pode fazer no Plone.

Examinando o Exemplo de Personalizações

As seguintes seções mostram alguns exemplos que demonstram simples personalizações que você pode fazer em seu Plone site. Algumas soluções fornecem um ou dois modos diferentes de responder a mesma pergunta.

Remover Bloco

Você pode facilmente remover blocos das interface do usuário tais como a barra de caminho (breadcrumbs) ou o box de busca (search box). Você tem duas maneiras de fazer isto; o mais obvio é personalizar a macro que mostra o elemento. Por exemplo, para remover a breadcrumbs (barra de caminho) você pode clicar `*portal_skins*`, depois `*plone_templates*`, `*global_pathbar*`, e então desabilitar o elemento no nível da page template; por exemplo, você pode mudar o seguinte:

```
<div metal:define-macro="path_bar"
  id="pathBar"
  tal:define="breadcrumbs python:here.breadcrumbs(here);
    portal_url portal_url|here/portal_url">
```

Para isto, adicione a seguinte linha de código:

```
<div metal:define-macro="path_bar"
  id="portal-breadcrumbs"
  tal:condition="nothing"
  tal:define="breadcrumbs python:here.breadcrumbs(here);
    portal_url portal_url|here/portal_url">
```

Isto significa personalizar a page template, no qual não é problema nenhum e é agora algo com que você pode estar familiarizado. A abordagem um pouco diferente é que você pode esconder elementos no nível do CSS. Isto ainda significa que o item é criado e o HTML é gerado, mas isto é desabilitado para o cliente. Porque a geração do HTML ainda ocorre, isto não é o ideal mas é uma boa solução.

A maioria dos elementos em Plone tem um único elemento ID DOM - Document Object Model; por exemplo, no caso do breadcrumbs (barra de caminho), é `*portal-breadcrumbs*`, como você vê no código anterior. Para não mostrar o `*portal-breadcrumbs*`, simplesmente adicione o seguinte ao `*ploneCustom.css*`:

```
#portal-breadcrumbs {
    display: none;
}
```

Mudar as Abas

Já foi mostrado como você pode mudar o texto da aba portal se você mudar as ações. Elas são mostrados utilizando a folha de estilo, sem utilizar tabelas (embora usuários podem inicialmente pensar assim). Olhando a Tabela 7-3, você pode ver o código para a aba portal que é `*portalTabs*`. Para fazer a borda da aba pontilhada não selecionada, você pode mudar simplesmente a folha de estilo em `*ploneCustom*` com o seguinte:

```
#portal-globalnav li a {
    border: 1px dotted;
}
```

As abas são uma serie de tags HTML de lista (`**`) e elementos ancora (`*<a>*`), assim mudando o CSS para estes elementos, você pode mudar a aparência das abas. Na seção "Estudo de caso: Examinando o SKIN do site da NASA Skin" será mostrado como mudar estas abas de textos para imagens.

Utilizando CSS você pode mover qualquer localização de um elemento alterando a posição do atributo. Depois, mova suas abas para o topo da tela, acima do logo e da caixa de pesquisa. Para fazer isto, você utiliza o valor da posição, no qual permite definir a posição utilizando os atributos `*left*`, `*right*`, `*top*`, e `*bottom*`. Adicione o seguinte na sua folha de estilo `*ploneCustom*` para trocar as abas portal para o topo do seu Plone site:

```
#portal-globalnav {
    position: absolute;
    top: 0em;
}
```

Está é uma técnica poderosa para mover elementos, possuindo múltiplas opções para posicionar os elementos, incluindo posicionamento relativo, mas isto toma um pequeno trabalho com CSS para fazer o posicionamento certo.

Mover os Slots Esquerdo e Direito

Falamos, no Capítulo 4, sobre os slots esquerdo e direito, e também foi mostrado como adicionar um novo para a lista de slots. Você deve ter notado que os termos `*left*` e `*right*` slots podem levar a um pequeno engano. O correto é mostrar os slots em suas posições, mas é fácil move-los.

****NOTA**** Isto funciona somente se você estiver utilizando o Plone Tableless skin. Não é um ajuste padrão, assim você tem que mudar o skin na ferramenta `*portal_skins*`, como já mostrado em "Gerenciando Skins com a Ferramenta `portal_skins`".

Por exemplo, se você quer mover os portlets a esquerda para o lado direito da página, então você pode fazer isto mudando `*ploneCustom.css*` para o seguinte:

```
#portal-column-one {
    float: right;
}
#portal-column-content {
    float: left;
}
```

Isto move a coluna mais a esquerda para a direita e empurra a seção principal para a esquerda.

Esconder a “Ajuda” dos Formulários

Se você quer esconder a “ajuda” em todos os formulários, você não pode na realidade mudar todos os templates. Mas você pode empregar uma tática similar para esconder o caminho da barra, ajustando `*display: none*` para os elementos do formulário. O seguinte tem o efeito desejado de não substituir a entrada de elemento em uma nova linha:

```
div.formHelp {
    display: none;
}
```

Figura 7-9 mostra a página de resposta sem a breadcrumbs (barra de caminho), com a “ajuda” escondida, com abas pontilhadas, e com o slot esquerdo movido para a direita da página, tudo modificado com apenas algumas linhas de CSS.

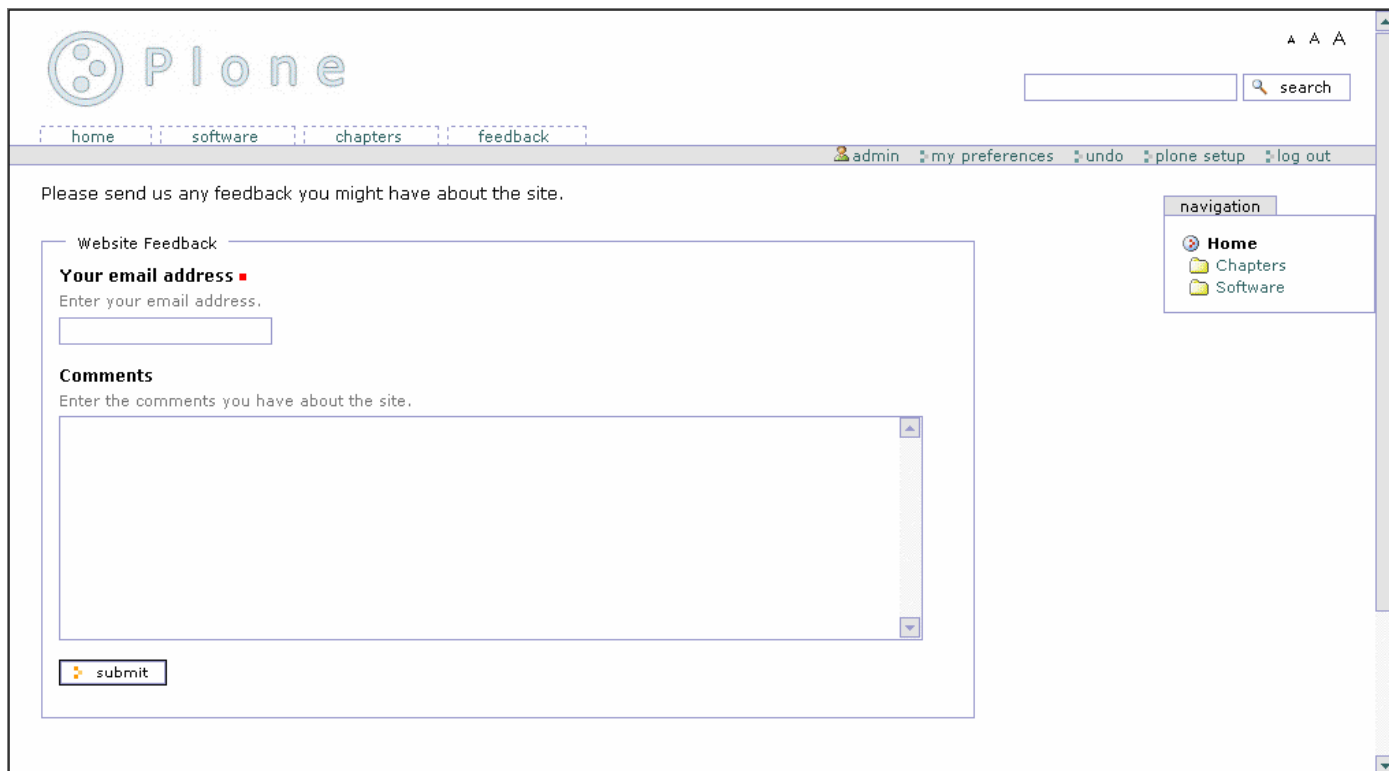


Figura 7-9. Efeito combinado de vários exemplos

Como Encontrar o Elemento X?

Como mostrado, os templates, scripts, e imagens contidos no diretório *skins* da instalação Plone cria um Plone skin. Muitos arquivos existem neste diretório, assim para passar por todos os arquivos é demorado e não é produtivo quando eles são modificados. Para tornar este trabalho um pouco mais fácil é preciso entender algumas técnicas básicas de procura de elementos dentro do Plone.

Tenha em mente, se posicione, no nível que você quer personalizar o elemento. Como já explicado, você tem três níveis para criar um objeto. Se você quer mudar a representação visual, ou sua localização, então as mudanças que deverão ser feitas é no CSS.

Se o CSS não é o suficiente, então a próxima etapa é procurar dentro dos templates. Por exemplo, suponha que você quer mudar o texto que aparece quando o usuário faz a autenticação, ou você queira mudar a página inteira. No exemplo, você irá alterar a página mostrada na Figura 7-10 para fazer um script que faz algo pouco comum.

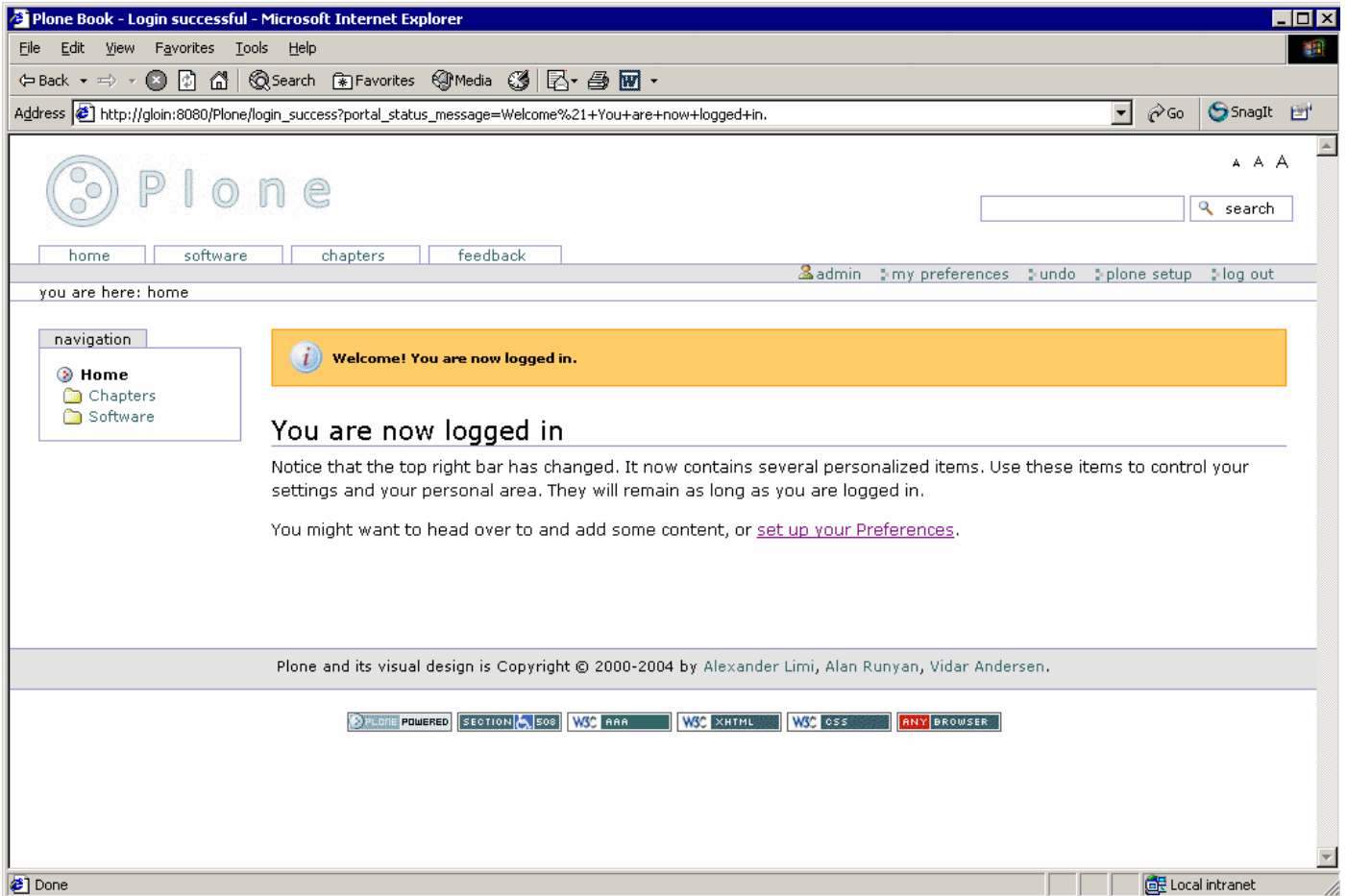


Figura 7-10. A página "Agora você está autenticado"

Existem algumas pistas para encontrar este template para que você possa alterá-lo; Irei explicar cada um deles agora.

Pesquisa Utilizando URL

The Uniform Resource Locator (URL) para a página traduz para uma série de objetos do Plone que são traversed. Na Figura 7-11, I've traversed para o página de `*login_success*`. Neste caso, a última parte da URL é `*login_success*`, como você pode ver na barra de endereço na Figura 7-11. Quando um objeto é carregado em um FSDV, a extensão é retirada, assim você esta procurando por um template ou um script que inicia com `*login_success*`.

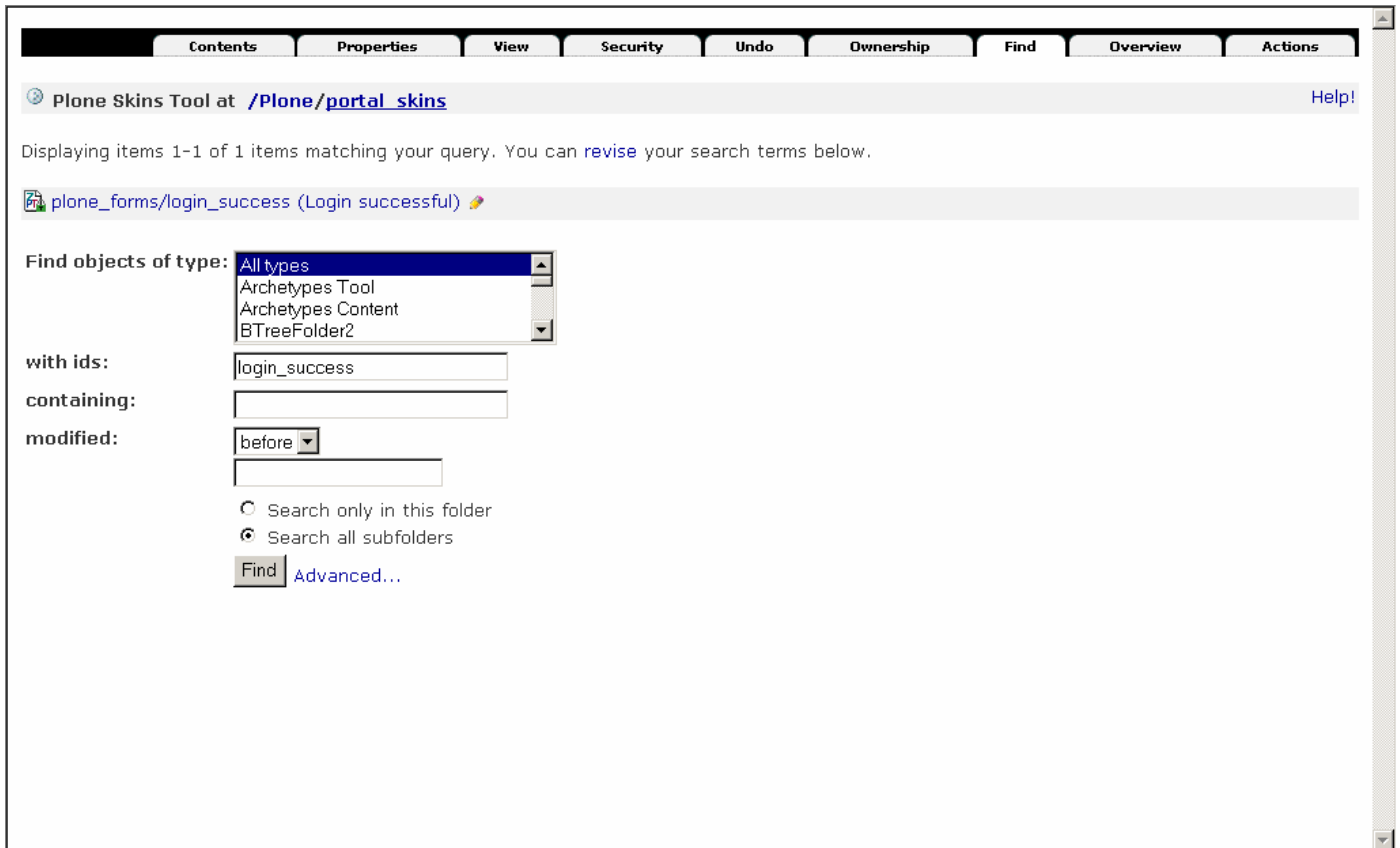


Figura 7-11. Pesquisando por uma ID

No Zope você pode executar esta pesquisa na ferramenta **portal_skins** e clicar na aba *Find*, entre com ***login_success*** no campo **with ids**. Deixe todas as outras opções iguais, e clique no botão *Find*. Você irá encontrar o template **login_success**.

Você pode também administrar esta procura no file system (sistema de arquivos), dependendo de seu sistema operacional e das ferramentas disponíveis. O modo mais rápido para encontrar este arquivo no Linux é ir ao seu diretório **CMFPlone** e fazer o seguinte:

```
$ cd skins
$ find -name 'login_success*' -print
./plone_forms/login_success.pt
```

No Windows, abra a pasta **CMFPlone** no Windows Explorer e clique na aba *Search*. Então entre com o nome do arquivo como ***login_success***, e clique em *Search*. Isto deve fornecer a lista dos arquivos prováveis.

Esta pesquisa deve fornecer o resultado, **CMFPlone/plone_forms/login_success.pt**. Se você executar a mesma pesquisa na ZMI, clique em **portal_skins**, depois **plone_forms**, e então clique em **login_success**.

Pesquisar por uma Parte de Texto

Uma outra maneira de fazer a pesquisa é procurar por uma parte do texto do código para encontrar o elemento que renderiza a página. Por exemplo, olhando a

página na Figura 7-12, você pode ver que ela contém o texto `*Notice that the top*`. A maneira mais simples de encontrar parte do texto é procurando por ele.



Figura 7-12. Pesquisando por texto

No Zope você pode executar esta pesquisa na ferramenta `*portal_skins*` e clicar na aba `Find`, entre com `**Notice that the top**` no campo `*containing*`. Deixe todas as outras opções iguais, e clique no botão `Find`. Você irá encontrar o template `*login_success*`.

Você pode também executar esta pesquisa no file system (sistema de arquivos), dependendo de seu sistema operacional e das ferramentas disponíveis. O modo mais rápido para encontrar este arquivo no Linux é ir ao seu diretório `*CMFPlone*` e fazer o seguinte:

```
$ grep -ri "Notice that the top" *
plone_forms/login_success.pt: Notice that the top
```

No Windows, abra a pasta `*CMFPlone*` no Windows Explorer e clique na aba `*Search*`. Então entre com o conteúdo do arquivo como `**Notice that the top**`, e clique `Search`. Isto deve fornecer para você a lista de arquivos prováveis. Utilizando uma dessas técnicas você encontrará o template, `*login_success*`, que renderiza a mensagem de retorno para o usuário.

Você tem que tomar os seguintes cuidados com estas técnicas de busca:

- cuidado com letras em maiúsculas e minúsculas - o resultado da busca pode ser sensível a este contexto.
- Se você tentar fazer isto em outra linguagem que não o Inglês, the content

may have been localized, causing the search to fail.

- em certos casos o texto pode não ter sido encontrado , assim executar a busca pela URL seria o mais recomendável.

Construindo Novos Skins e Layers

Não muito atrás falei sobre personalizar os skins existentes. O processo para fazer um novo skin por inteiro e um novo layer (camada) não é muito diferente. Irei explicar um ponto chave, como colocar seus templates e scripts no file system.

Fazer templates e scripts no file system e criar novos skins e layers são definitivamente o melhor modo para manutenção do site. Não por ser mais fácil criar elementos de skin nas ferramentas do file system, mas também ele permite você distribuir seu código. Desenvolver no file system é a escolha da maioria dos desenvolvedores Plone, com as menores modificações no diretório `*custom*` se necessário.

Construindo um Novo Skin

Como vimos um skin na prática não é nada mais que uma coleção de layers (camadas). Para meu novo skin, vou colocar todos meus códigos em um único lugar, então fui à ferramenta `*portal_skins*`, adicionei uma nova pasta, e dei a ela um ID de `*custom_chrome*`.

Então, para adicionar um novo skin, você tem que clicar em `*portal_skins*`, selecionar a aba Properties, e adicionar um novo skin abaixo do texto `*Add a new skin*`. Você tem que entrar com a série de layers (camadas) que você deseja para ajustar esta skin. Neste exemplo, adicionei uma nova skin chamada `*Custom Chrome*` e uma série de layers (camadas), como mostrado na Figura 7-13.

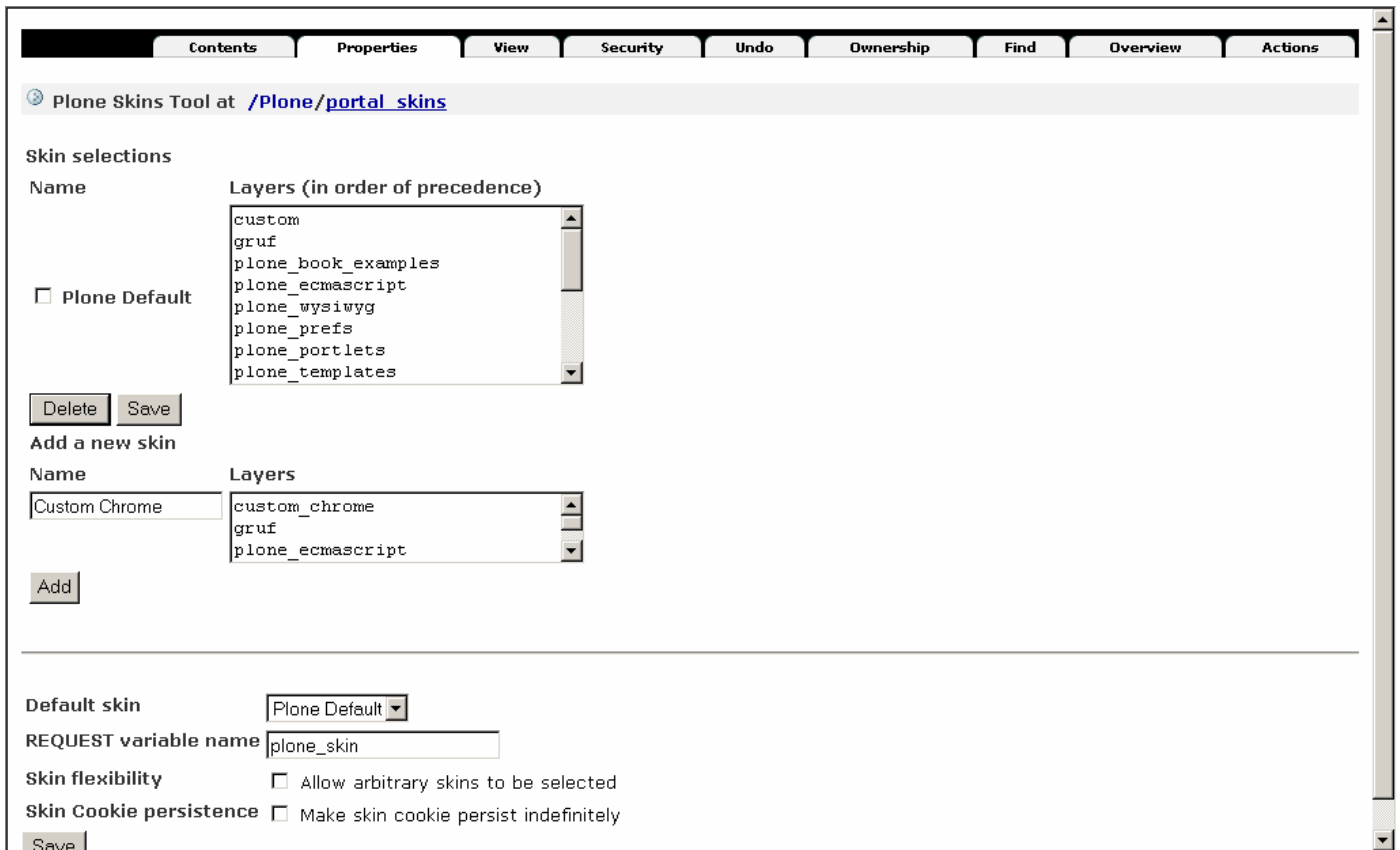


Figura 7-13. Adicionando uma Custom Chrome skin

Então adicionei as layers (camadas) para o skin. Neste caso, o skin não possui a layer chamada *custom*; ao invés tem uma pasta chamada *custom_chrome*. Você agora tem duas skins que utiliza duas camadas e duas pastas. Todo objeto adicionado na pasta *custom_chrome* irá afetar esta skin, não o Plone Default skin.

Utilizando Múltiplos Skins

Como mencionado, um site Plone padrão possui duas skins, Plone Default e Plone Tableless. Na seção anterior, foi adicionada uma nova skin, chamada Custom Chrome. Como discutido no Capítulo 4, você pode ajustar o skin padrão utilizando a Plone interface. Clique *plone setup*, e então clique no botão Portal Skin. This mirrors the choices available in the ZMI after clicking *portal_skins*, selecting the Properties tab, and scrolling to the bottom of the page.

Você tem mais uma opção, pelo: *REQUEST *variable name* *. Esta é a variável request que irá conter as informações do skin do usuário. Este é *plone_skin* por default, que é o nome do cookie. Mas isto também pode ser passado por outras variáveis como a string query, sendo disponível somente pela ZMI.

Você pode também ajustar os skins programaticamente. Isto permite aos desenvolvedores mostrar diferentes skins para diferentes usuários para certo assunto ou lógica do site. Por exemplo, se um usuário esta escrevendo conteúdo para um site, ele deve ver o skin padrão do Plone. Se for usuário anônimo, então pode ver um skin totalmente diferente. Se você realmente deseja, você pode apoiar o skin na pasta que o usuário está acessando; entretanto, este

acesso pode ocasionar confusão, assim não é recomendável.

Para mudar o skin, adicione um objeto Script (Python) chamado `*setSkin*` na raiz do Plone site e adicione o seguinte código:

```
##title=Skin changing script
##parameters=
req = context.REQUEST
if req['SERVER_URL'].find('internal.somesite.org') > -1:
    context.changeSkin("Plone Default")
context.changeSkin("Custom Chrome")
```

A lógica atual para determinar o skin irá depender das regras de assuntos dos sites. Neste caso qualquer pessoa acessando `*http://internal.somesite.org*` irá fornecer o skin Plone Default, e qualquer pessoa que acessar `*http://external.somesite.org*` será fornecido o skin Custom Chrome. Infelizmente, você não pode determinar o skin no nível de segurança dos usuários (por exemplo, authenticated users (usuários autenticados) vêem um skin, e os administradores vêem outro). Isto pode parecer obvio mas não é possível a não ser que você gaste um bom tempo “hackeando” o seu site Plone.

****NOTA**** Esconder informações através da utilização de skins é uma pratica comum mas não completamente segura. Para ter certeza que é seguro irá depender das configurações da rede. Na maioria dos casos, você pode cuidar disto facilmente no firewall ou utilizando um servidor proxy tal como Apache, que pode ser configurado para bloquear todos os pedidos externos para `*http://internal.somesite.org*`. A integração com Apache é discutida no Capítulo 10.

Para ativar este código, atribua uma regra de acesso para este objeto. Isto significa que a cada vez que este o site Plone é acessado, o objeto Script (Python) será executado e o skin será ajustado de acordo com o script. Para atribuir a regra para este script, selecione Set Access Rule no menu drop-down e então entre com o nome do seu objeto Python (Script).

Você tem que ter cuidado com as regras de acesso porque elas são executadas em todo pedido de pasta (ou Plone site); você tem que se assegurar que as regras estão corretas e que não possuam nenhum erro. Se você acidentalmente escreveu um objeto Script (Python) e não consegue ter acesso de volta na ZMI para arrumar isto, então você desative as regras de acesso reiniciando o Plone com as seguintes variáveis de ambiente:

```
::
SUPPRESS_ACCESSRULE = 1
```

Apêndice B explica como ajustar as variáveis de ambiente se você não é familiar com este processo.

Fazer um novo Skin no File System

Em todos os capítulos foi utilizada a ZMI, mas a maioria dos desenvolvedores utiliza o file system. Fazer um skin no file system é realmente fácil.

Vá para a instância do diretório home da sua instalação Plone. Dentro do diretório `*Products*`, crie um novo diretório; o nome do diretório será o nome do produto, normalmente é uma palavra curta, sem espaços ou sublinhados e contendo caracteres em caixa baixa e alta. `*PloneBookExample*`, `*CMFPlone*`, e `*PloneSilverCity*` são exemplos. Dentro desta pasta, crie um arquivo chamado `*__init__.py*` e um diretório chamado `*skins*`. No arquivo `*__init__.py*`, você precisa adicionar as seguintes linhas:

```
from Products.CMFCore import DirectoryView
DirectoryView.registerDirectory('skins', globals())
```

Próximo passo, reinicie o Plone, e então clique em `*portal_skins*` para adicionar um `*FSDV*`. Isto irá abrir uma lista de diretórios registrados. Procure o seu diretório que você registrou; este será o nome do diretório com `*/skins*` no final. Entre com um ID, e clique `*Add*`. Agora você tem um diretório vazio onde você pode adicionar layers (camadas) para seus skins.

Corrigindo Skins

Outra razão para estar usando a ZMI repetidamente, é que ao invés do file system, ela te fornece o feedback sobre os erros e fornece a você maior comodidade para copiar e mover objetos entre as pastas. Outra característica positiva de utilizar a ZMI é que as mudanças são instantâneas. Se você mudar um objeto e então atualizar a página, você verá a mudança imediatamente (se não tiver cache).

Este não é o caso do file system. Se você mudar alguma coisa através do file system, por razões de desempenho, isto não será atualizado no Plone. O Plone não tem como saber que foi feita esta mudança, não tem um mecanismo de notificação automática. O sistema do Plone funciona em dois modos/estados: modo produção e modo debug. Quando o Plone está em modo debug, ele checa todos os diretórios, e encontra os arquivos que foram modificados, e então atualiza o Plone. Isto significa que você pode fazer uma modificação, e ela aparecerá imediatamente. Quando esta sendo executado no modo de produção, suas mudanças não irão acontecer até você atualizar o skin (veja Capítulo 11) ou reiniciar o Zope.

Por razões óbvias, se você esta desenvolvendo skins em Plone, então execute-o em modo debug. O Capítulo 2 mostra como você pode mudar a configuração para o Plone. Rapidamente falando, abra o arquivo `*zope.conf*` dentro do diretório `*etc*` de sua instalação e assegure que o diretivo `*debug-mode*` esteja ajustado para `*on*`.

Utilizando o File System Objects

O FSDVs permite apenas o mapeamento de objetos Zope que possui configuração específicas para serem utilizados desta maneira. Isto é determinado pela extensão do filename (nome do arquivo). O conteúdo destes arquivos é o conteúdo de um atributo do objeto, normalmente o conteúdo principal, tal como o conteúdo binário de uma imagem, ou o texto contido dentro do template.

Para criar um objeto em seu FSDV vazio, apenas vá ao diretório `*skins*` e

adicione arquivos que sejam compatíveis com os objetos que você queira criar. Uma vez que o arquivo é carregado no Zope como um objeto Zope, a extensão é retirada. Por exemplo, `*some_template.pt*` se torna uma file system page template com o ID `*some_template*`. A Tabela 7-4 descreve as extensões.

Tabela 7-4. Extensões

****Extensions Object Type Equivalent Zope Object****

`*.pt*`, `*.zpt*`, `*.html*`, `*.htm*` Filesystem Page Template Page Template

`*.cpt*` Controller Filesystem Page Template Controller Page Template

`*.py*` Filesystem Script (Python) Script (Python)

`*.cpy*` Controller Python Script Controller Python Script

`*.vpy*` Controller Validator Controller Validator

`*.doc*`, `*.pdf*`, `*.swf*`, `*.jar*`, `*.cab*`, `*.ico*`, `*.js*`, `*.css*` Filesystem File File

`*.gif*`, `*.jpg*`, `*.jpeg*`, `*.png*` Filesystem Image Image

`*.props*` Filesystem Properties Object Folder with Properties

`*.zsql*` Filesystem Z SQL Method ZSQL Method

`*.dtml*` Filesystem DTML Method DTML Method

Assim, para obter uma imagem em seu diretório view, adicione um arquivo com a extensão `*.gif*` ou `*.jpeg*`. Se você quiser um objeto Script (Python), então adicione um arquivo com a terminação `*.py*`.

Ajustando o File System Object Metadata

Conteúdo extra para um objeto é o title, security, e a cache é armazenada em um arquivo diferente, com `*.metadata*` adicionado no final. Se o arquivo original é `*logo.jpg*`, por exemplo, então o metadata será incluído `*logo.jpg.metadata*`.

O arquivo metadata está no Windows no formato `*.ini*` com os pares `*chave = valor*`. Esse formato foi estendido para conter informações sobre as formas para o objeto Form Controller, que será visto na próxima seção. Todas as escolhas, mesmo a presença deste arquivo, são opcionais. O que segue é um exemplo de arquivo:

```
[default]
title = Test object
cache = RAMCache
proxy = Manager
```

```
[security]
Access contents information = 1:Manager,Anonymous
```

O que segue são os valores que você ajustou neste arquivo:

****title****: Este é o title que foi aplicado ao objeto na ZMI e no Plone; será mostrado no Plone templates.

****cache****: Este é o ID do objeto cache do qual você deseja que seja cacheado. Por default o Plone vem com dois objetos cache: o RAM Cache Manager e o HTTP Cache Manager.

****proxy****: Esta é a regra proxy que você deseja aplicar a este objeto. Veja Capítulo 9 para mais informação.

****security****: Esta é a área de segurança, que permite múltiplas linhas de ajustes de segurança. A chave possui o nome da permissão. O lado direito contém ajuste de aquisição, seguido das regras delimitadas por vírgulas. Por exemplo, `*View = 0:Manager*` significa que apenas usuários com regras de membro e administrador podem ver o objeto, e os ajustes de segurança não são adquiridos para esta permissão.

Utilizando Validators no File System

Para especificar os validators no file system, adicione o verificador no arquivo `*.metadata*`. A seção validator do arquivo `*.metadata*` se parece com isto:

```
[validators]
validators = validate_script1, validate_script2
```

Isto vai executar dois scripts de validação: `*validate_script1*` e `*validate_script2*`, nesta ordem. O script de validação irá examinar o dado e adicionar erros ao formulário controlador de estado, se tiver algum problema.

A opção `*contextType*` e `*button*` precisa de uma sintaxe um pouco diferente. As validações são executadas no contexto que estão sendo executados – por exemplo, um documento ou imagem. Você pode ter diferentes validadores executados para os documentos e para as imagens. Por exemplo, para ter um script validator diferente executado quando for um documento, adicione as seguintes linhas:

```
validators.Document = validate_script2
```

Você pode diversificar os validadores, dependendo do botão clicado no formulário, anexando o nome do botão no formulário. O nome do botão deve começar com `*form.button*`. Por exemplo:

```
<input type="submit" name="form.button.button1" value="First" />
```

O arquivo metadata deve então conter o seguinte:

```
validators..button1 = validate_script1
```

O `*..*` é um espaço para o tipo de conteúdo, assim se você quiser que ocorra para o `*button1*` em um documento, o arquivo metadata irá se parecer com:

```
validators.Document.button1 = validate_script5
```

Utilizando Actions no File System

Da mesma forma como os validadores, você pode especificar actions no arquivo `*.metadata*`. A sintaxe para a seção `*actions*` é:

```
[actions]
action.success = traverse_to:string:script1
```

No exemplo anterior, quando o formulário é enviado os scripts de validação retornam o estado de sucesso, the `traverse to` action é chamado com o argumentos `*string:script1*`. A action default para o estado de falha é carregada novamente no formulário atual. O formulário terá acesso a todas as mensagens de erro pelo objeto `*state*`.

Você pode especificar uma action em particular em contexto específico; por exemplo, para especificar uma action de sucesso quando em um documento, você pode fazer o seguinte:

```
action.success.Document = traverse_to:string:document_script
```

Novamente, você pode especificar a action para o seguinte botão:

```
<input type="submit" name="form.button.button1" value="Button" />
```

Adicionando o que se segue no arquivo `*.metadata*`:

```
action.success.button1 = traverse_to:string:script1
```

Este exemplo não tem nenhum contexto explícito, assim ele é validado para qualquer tipo de contexto.

```
***start sidebar***
```

Encontrar Exemplos de File System Skins e Exemplos do Livro

Todos os exemplos neste livro foram coletados em um skin para você instalar. Você pode encontrar no Plone book Web site em `*http://plone-book.agmweb.ca/Software/PloneBookExamples*` e no Apress book Web site em `*http://www.apress.com*`. Está disponível como um arquivo `*.zip*` de um skin; após você baixar e descompactá-lo, você vai encontrar uma estrutura de arquivo parecida com o já mencionado.

Você tem um arquivo `*__init__.py*` e um diretório `*skins*`. Neste diretório você irá encontrar uma série de page templates, Controller Validator objects, e todos os arquivos metadata. Se você quiser instalá-lo copie a pasta `*PloneBookExamples*` para o diretório `*Products*` de sua instância home. Reinicie o Plone, e então clique em `*plone setup*`. Selecione `Add/Remove Products`, `*e*` `*você vai ver uma entrada para *PloneBookExamples*`; verifique isto, e então clique `install`. Você agora instalou os templates e pode ir para `*feedbackForm*` e adquirir a page template, essa parte foi vista no capítulo anterior.

Qual o procedimento de instalação que automatiza o processo de adicionar um FSDV e então adicione uma layer (camada) para cada skin. Se você clicar *portal_skins* e então selecionar a aba Properties, você irá ver a nova camada *plone_book_examples* adicionada.

end sidebar

Caso de Estudo: NASA Skin

Em Janeiro de 2004 duas sondas da NASA pousaram em Marte: Spirit e Opportunity. Estes robôs controlados remotamente andaram na superfície de Marte, enviando fotos e análises da superfície. As sondas tiveram grande sucesso, enviando impressionantes fotos da superfície de Marte que impressionaram o Mundo.

Uma pequena parte de todo este projeto foi o site *http://mars.telascience.org*. Este site tem publicado o programa chamado *Maestro*, que entre suas funcionalidades:

baixar versões, de tamanhos diferentes, do programa que os cientistas da NASA utilizaram para operar o Spirit e Opportunity. Atualizações também são disponíveis para o maestro que contem dados reais de Marte e você pode adicionar para sua copia do Maestro.

Um grande número de membros de comunidades e voluntários ajudou os membros do grupo no desenvolvimento do site do Maestro. Figura 7-14 mostra o Plone site.

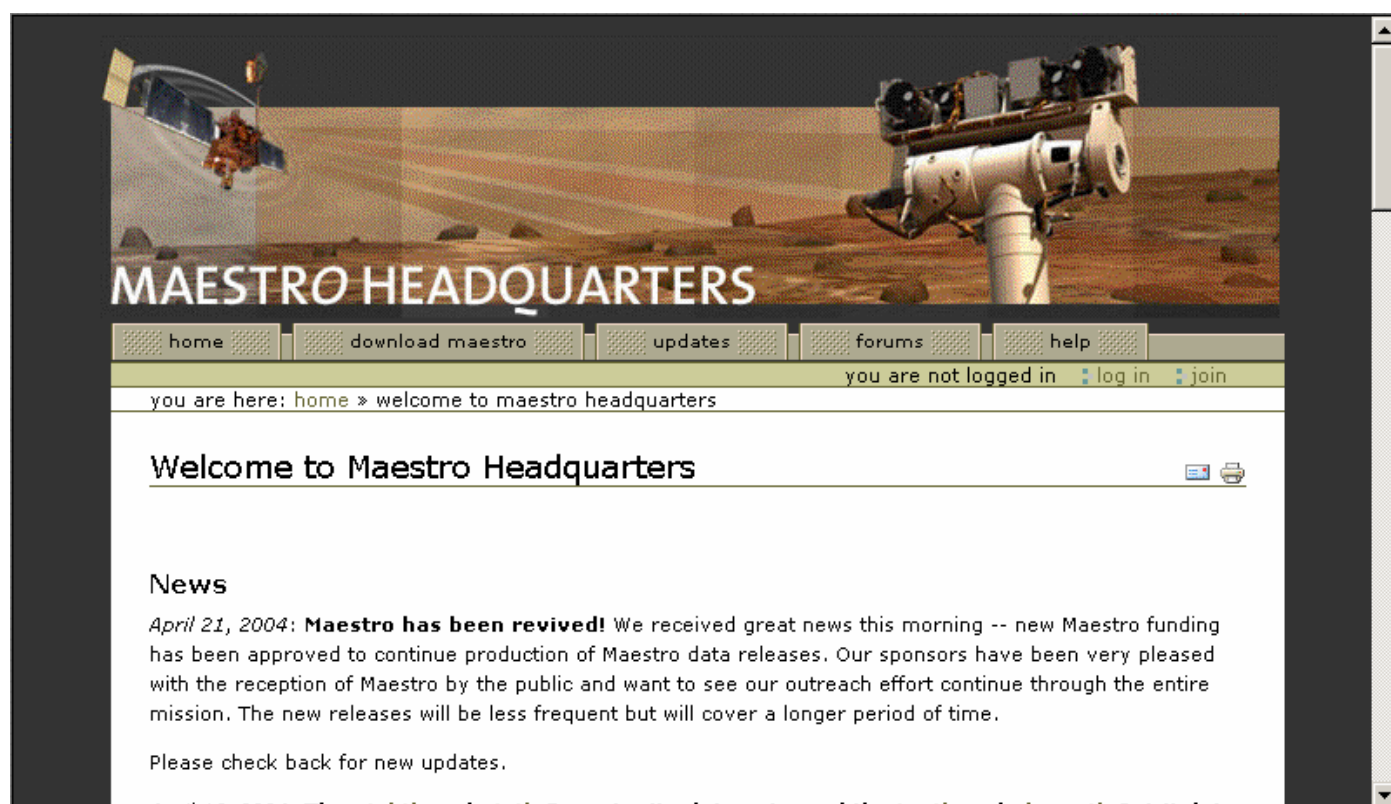


Figure 7-14. The Maestro site

Você provavelmente reconhece alguns sinais característicos de site Plone: as abas transversais no topo, a barra pessoal no canto do topo direito, e o barra de localização (breadcrumbs). Além disso, o site se parece totalmente diferente

de um site Plone padrão. Nas seções seguintes, será passado exatamente como isto foi feito. Bem, é muito simples porque a maioria feitas foi através da modificação do CSS.

Primeiramente vamos falar de algumas mudanças feitas, no template e nas propriedades, foras do arquivo CSS.

Remover Portlets e alguns dos elementos principais

Os portlets foram removidos porque neste site não são relevantes. Para removê-los, a partir da ZMI, vá para raiz do Plone site e clique em Properties e apague todos os valores dos campos `*left_slots*` e `*right_slots*`.

Neste site, poucos elementos foram removidos: as actions do site, o campo de busca (search box), o rodapé, e o colophon.

Para completar, alguns templates foram personalizados/alterados. Por exemplo, para remover a caixa de pesquisa, na ZMI clique em `*portal_skins*`, `*plone_templates*`, `global_searchbox*` e então no botão Customize, agora modifique o template para o seguinte:

```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en"
  i18n:domain="plone">
<body>

  <div id="portal-searchbox"
    metal:define-macro="quick_search"
    tal:condition="nothing">
    Nothing to see here.
  </div>
</body>
</html>
```

Esta técnica já foi mostrada para remover elementos; apenas ajuste o elemento macro `*tal:condition*` para assegurar que esta condição seja `*falsa*`.

Personalizando Cores

Você personalizou as cores para o site no objeto `*base_properties*`, sendo modificadas para as seguintes:

```
linkColor: #776a44
globalBorderColor: #776a44
globalBackgroundColor: #e0d3ad
globalFontColor: #776a44
```

A cor modificada e de fácil percepção é a `*globalBackgroundColor*`, que afeta a cor da barra pessoal que foi mudada de azul para marrom. Estas cores menos importantes irão alterar a base da folha de estilo assim combinando com as imagens e, sobretudo ficar com uma aparência legal.

Criando folha de estilo

A maior parte da aparência do se encontra da folha de estilo, que está reproduzida no apêndice B. Onde serão destacadas algumas das principais partes da folha de estilo. Este folha de estilo é baseada em `*ploneCustom.css*`, a qual foi personalizada na pasta `*custom*`. Então, alguns dos elementos da página foram sobrescritos no novo arquivo `*ploneCustom.css*`.

Primeiro, o background do corpo da página foi ajustado para a cor `*#343434*`.

```
body {
  background: #343434;
}
```

Segundo, o conteúdo que você pode editar está contido na classe chamada `*documentContent*`. Porque a cor do background do elemento `*documentContent*` é ajustado para `*white*` no arquivo principal `*plone.css*`, o background do texto é branco que produz uma área em branco no meio da tela.

Próximo, a imagem do satélite e do robô no topo do Web site é uma imagem enorme. Você a coloca no topo utilizando CSS. O código para isto é:

```
#portal-top {
  background: url("http://mars.telascience.org/header.jpg") transparent no-repeat;
  padding: 162px 0 0 0;
  position: relative;
}
```

Este código CSS ajusta os parâmetros para o elemento que tem o ID `*portal-top*`. Se você observar o código HTML do Plone site, você irá ver o elemento `*portal-top*` no topo da página, exatamente abaixo do elemento `*body*`. Ajustando o background da imagem para a URL em questão, você pode ter uma imagem com 162 pixels de altura, configurando o elemento `*#portal-top*` para `*162px*`. Se você não fizer isto, então todos os itens abaixo serão empurrados para cima, afastando-se da imagem.

A imagem do cabeçalho tem 677 pixels de largura, e você notará que o texto cabe na página perfeitamente abaixo da imagem. Você pode fazer isto ajustando o valor do elemento para `*680px*`. O elemento `*visual-portal-wrapper*` HTML abaixo do body, tem sua largura ajustada para todo o corpo da página. O código para isto é o seguinte:

```
#visual-portal-wrapper {
  width: 680px;
  margin: 1em auto 0 auto;
}
```

Isto ajusta a largura para todas as páginas que possuam um largura fixa. Não importando o tamanho da janela do browser do usuário, a parte principal da página nunca irá crescer além dos 680 pixels e assegurando que combine com a imagem.

Provavelmente outras mudanças foram feitas nas abas no topo da página, que são agora imagens ao invés de apenas caixas padrões do Plone. São três imagens que criam as abas no topo da página: o espaçamento entre as abas, a parte esquerda da aba, e a parte direita da aba(tab). Colocando-as juntas, você obtém o efeito de uma aba. A Figura 7-15 mostra estas três imagens.



Figure 7-15. As 3 imagens que formam uma aba

Para editar o CSS, lembre-se que cada aba é uma lista de itens que contém um link dentro de um elemento com o ID `*portal-globalnav*`. Para ajustar a figura de fundo que preenche o espaço entre as abas, o skin primeiro determina a figura de fundo pra todo o elemento. Observe que ajustando a altura da imagem para 21 pixels, que é o mesmo tamanho da imagem, você assegura o espaço apropriado para a imagem. O código, para isto, é o seguinte:

```
#portal-globalnav {
  background: url("http://mars.telascience.org/listspacer.gif") transparent;
  padding: 0;
  height: 21px;
  border: 0;
  margin: 0 0 1px 6px;
  clear: both;
}
```

Para ajustar a imagem a esquerda da aba, você utiliza o `*start image*`, ajustando o valor no elemento `*li*`, ao invés do elemento `*anchor*`, assim:

```
#portal-globalnav li {
  display: block;
  float: left;
  height: 21px;
  background: url("/liststart.gif") transparent no-repeat;
  padding: 0 0 0 33px;
  margin: 0 0.5em 0 0;
}
```

Finalmente, você ajusta a parte direita da aba adicionando uma imagem para o elemento `anchor`, alterando este elemento dentro da aba. O código a seguir mostra onde você ajusta esta imagem de fundo:

```
#portal-globalnav li a {
  display: block;
  float: left;
  height: 21px;
  background: url("/listitem.gif") transparent right top;
  padding: 0 33px 0 0;
  border: 0;
  line-height: 2em;
  color: black;
  font-size: 90%;
}
```

```
margin: 0;
}
```

Assim você substituiu a aparência padrão das abas do Plone, por botões de uma aparência mais agradável.

Criar Splash Page

Esta página possui um outro elemento chave. O **splash page** fica a frente da página do site, o qual mostra um gráfico e chamando a atenção dos usuários para navegar no site. Você pode adicionar isto indo até a ZMI e removendo o objeto **index_html** onde normalmente estaria contido. Uma vez removido, crie um novo arquivo chamado **index_html**. Neste arquivo, crie o código para mostrar a home page, incluindo o custom CSS. O elemento principal do objeto é uma imagem, colocada pelo seguinte CSS:

```
div {
  background: url(/splash.jpg) transparent no-repeat;
  width: 260px;
  height: 335px;
  position: absolute;
  ...
}
```

O código CSS restante trata o do texto e do links com a imagem. Esta página não tem elementos Plone e é um código HTML estático.

Conclusão

A primeira vista pode parecer um site razoavelmente complexo, um código de folha de estilo (CSS) relativamente simples fez a maior parte do trabalho. Utilizando o CSS você altera a aparência do Plone sem ter que saber muito sobre Plone além de HTML. Também, assegurando a colocação das imagens através do CSS, você mantém as funções chave da acessibilidade.

Agradeço a NASA e a todas as pessoas envolvidas da Comunidade Plone por ajudar com o site e com este caso de estudo. Estes incluem, mas não se restringem a John Graham, Alma Ong, Joe Geldart, Michael Zeltner, e Tom Croucher.