Nice Flash messages in Rails 2 and Rails 3

 I've been doing some Rails work recently and one of the things that I've found to be a nice aesthetic add-on is putting some nicer flash messages in.

Over at dzone.com there is an example which I've worked off of – http://snippets.dzone.com/posts/show/3145

The example is as follows. Add this section to your **application_helper.rb**

```
def show_flash
    [:notice, :warning, :message].collect do |key|
        content_tag(:div, flash[key], :class => "flash flash_#{key}") unless flash[key].blank?
    end.join
end
```

In your CSS file you now add the settings to style your messages. Note that my examples assume some images are present so add images as you wish for within the message:

```
.flash_notice {

BORDER-RIGHT: #090 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%; BORDER-TOP: #090 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE: url(/images/icon_success_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT: #090 4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #090 4px solid; BACKGROUND-REPEAT: no-repeat

}

.flash_warning {

BORDER-RIGHT: #c60 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%; BORDER-TOP: #c60 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE: url(/images/icon_warning_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT: #c60 4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #c60 4px solid; BACKGROUND-REPEAT: no-repeat

}

.flash_error {

BORDER-RIGHT: #f00 4px solid; PADDING-RIGHT: 10px; BACKGROUND-POSITION: 5px 50%; BORDER-TOP: #f00 4px solid; PADDING-LEFT: 10px; BACKGROUND-IMAGE: url(/images/icon_error_lrg.gif); PADDING-BOTTOM: 10px; MARGIN: 2px; BORDER-LEFT: #f00 4px solid; TEXT-INDENT: 40px; PADDING-TOP: 10px; BORDER-BOTTOM: #f00 4px solid; BACKGROUND-REPEAT: no-repeat

}
```

In your view, which in my case is the views/layouts/application.html.erb (same for Rails 2 or 3) I have added this where I want my flash notices:

```
<%= show_flash %>
```

If you are running Rails 3 there are two additional steps you need to take. First you need to change the code on your view to display as follows:

```
<%= raw show_flash %>
```

Note that under Rails 3 the flash will render the html code as text rather than as the raw html. By adding the raw statement you fix this issue and it will render correctly.

The second change you want to make is for your scaffolds. By default the show.html.erb file already contains a message section. If you leave this in place you will end up with 2 flash messages on each render of the show action. To fix this, go into the show.html.erb under your app/view/scaffoldname/ folder and remove this line:

```
<p id="notice"><%= notice %></p>
```

Here are the 3 image files that I've used as referenced in the CSS

That's my hopefully useful Rails tip of the day!

PowerShell – Listing aged out computer accounts in Active Directory

Do you remember having to use OldCmp.exe from JoeWare (great tools by the way) to capture a a list of computer accounts which have not been attached to your Active Directory for a period of time. Luckily with PowerShell and Quest ActiveRoles you can do this with a simple two line script.

Here's the code: http://gist.github.com/492494

Find Old Computers In Active Directory

PowerShell – Get serial numbers for computers in Active Directory

There are a lot of posts about pulling data from a file to do actions against computers/users. While

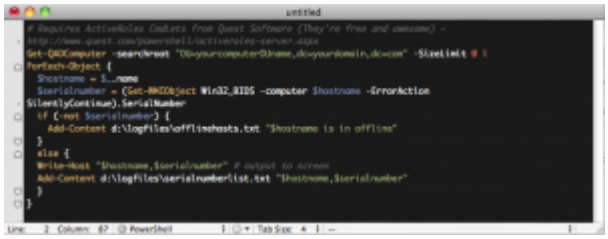
this is valuable, I prefer to do dynamic capturing of computer and user objects directly from my Active Directory.

This is a simple script which will capture a list of computers from Active Directory and subsequently pull the serial number using the **Get-WMIObject** CmdLet using the **Win32_BIOS** class.

The neat addition over many other scripts I have used is the addition of the **-ErrorAction SilentlyContinue** option which prevents offline computers from sending error information in the output. Add some error handling and you now have a list of computers which are offline or inaccessible.

As with many of the scripts I will show you, you need to install ActiveRoles CmdLets from Quest Software (follow the link in the Gist)

Here's the code: http://gist.github.com/492475



List of serial numbers from Active Directory computers

PowerShell – Active Directory – Email users with password expiry < 14 days

PowerShell – Active Directory – Email users with password expiry under 14 days

This script is handy for users who are not typically on your network (remote access, email only etc…). Normally users are warned at 14 days before expiry unless they are not logging into Active Directory through the workstation in which case they will not receive warnings until the password expires.

Simply set this to run daily and it will locate and email anyone who has a password expiry less than 14 days. To change the criteria, simply modify the If statement.

You need to install the ActiveRoles CmdLets from Quest Software (follow the link in the Gist)

Here's the code: http://gist.github.com/469656

EmailADPasswordExpiryUnder14days.ps1

```powershell
# Requires ActiveRoles CmdLets from Quest Software (They're free and awesome) -
http://www.quest.com/powershell/activeroles-server.aspx
Get-QADUser -SizeLimit 0 | Select-Object samAccountName,mail,PasswordStatus |
Where-Object {$_.PasswordStatus -ne "Password never expires" -and $_.PasswordStatus -ne "Expired" -and
$_.PasswordStatus -ne "User must change password at next logon." -and $_.mail -ne $null} |
ForEach-Object {
  $today = Get-Date
  $logdate = Get-Date -format yyyyMMdd
  $samaccountname = $_.samAccountName
  $mail = $_.mail
  $passwordstatus = $_.PasswordStatus
  $passwordexpiry = $passwordstatus.Replace("Expires at: ","")
  $passwordexpirydate = Get-Date $passwordexpiry
  $daystoexpiry = ($passwordexpirydate - $today).Days
  if ($daystoexpiry -lt 14) {
    $emailFrom = "PasswordManagement@yourdomain.com"
    $emailTo = "$mail"
    $subject = "Your password will expire in $daystoexpiry days"
    $body = "Please change your password to prevent loss of access to your corporate systems`n`n"
    $body += "If you are unable to change your password, please contact the help desk"
    $attachment = "C:\batch\docs\PasswordChange.doc"  # Attach a HowTo document for changing passwords
    $smtpserver = "smtp.yourdomain.com"
    Send-MailMessage -To $emailTo -From $emailFrom -Subject $subject -Body $body -Attachments $attachment
-SmtpServer $smtpserver
    Write-Host "Email was sent to $mail on $today"
    Add-Content c:\batch\logs\maillog$logdate.txt  "Email was sent to $mail on $today"
  }
}
Send-MailMessage -To "administrator@yourdomain.com" -From "PasswordManagementProcess@yourdomain.com"
-Subject "Password change log for $today" -Body "This is the log from $today" -Attachments
"d:\batch\logs\maillog$logdate.txt" -SmtpServer $smtpserver
```

Line: 27 Column: 1 PowerShell Tab Size: 4

Email Password Expiry Under 14 days