mp3harvester

Below is a small piece of Ruby code that I wrote to harvest mp3 files from an RSS feed. The code was designed to be run as a cron on a local machine once a day. It checks an RSS2 feed for new content, looks for any mp3s listed in the enclosure tags, and if they exist, it downloads each of them to my computer, creates a SIP that DSPACE can accept, uploads the SIP to my DSpace machine, and imports it using the batch importer. I thought that someone else might want to use or adapt something like this.

To use the code, create a file with a name like "mp3harvester.rb". Put the code in it. Change the appropriate paths, usernames, and passwords. Ensure that you have Ruby and the referenced Gems installed. Then, just run the file on a cron job once a day.

```
# Creator: Jason Fowler
# This  script is designed to harvest mp3 files and their metadata from the RSS feed and add it to DSpace

# get required libraries
require "rss/1.0"
require "rss/2.0"
require "rss"
require "mp3info"
require "shared-mime-info"
require 'net/sftp'
require 'net/ssh'

# supply local directory for storage here
def go_home
        Dir.chdir("/PathToStagingDirectoryOnMyLocalMachine")
end

# supply your Dspace server's URL, your username, and your password below.
def upload(package)
        # Upload the package to DSpace
        Net::SFTP.start('my.dspace-server.edu', 'dspace-username', :password => 'dspace-password') do |sftp|
                sftp.upload!("#{package}", "/PathToBatchDirOnMyServer/batch_uploads/#{package}")
        end

# Import the package into DSpace
        Net::SSH.start('my.dspace-server.edu', 'dspace-username', :password => 'dspace-password') do |ssh|
                ssh.exec 'rm /PathToBatchDirOnMyServer/batch_uploads/mapfile.txt'
                ssh.exec 'cd /PathToDspace/bin/'
                ssh.exec "/PathToDspace/bin/dsrun org.dspace.app.itemimport.ItemImport -a -e
myemail@myinstitution.edu -c 12345/1 -s /PathToBatchDirOnMyServer/batch_uploads/#{package} -m
/PathToBatchDirOnMyServer/batch_uploads/mapfile.txt"
        end
end


# Get RSS Feed and parse it
source = "http://MyBlog.com/resources/feed/rss2"     # replace this with your blog's RSS url
content = "" # raw content of rss feed will be loaded here
open(source) do |s| content = s.read end
rss = RSS::Parser.parse(content, false)

# Create Temporary Directory for storing files
go_home
@time = Time.now.strftime("%Y%m%d")
@directory = "#{@time}"
Dir.mkdir("#{@directory}")
Dir.chdir("#{@directory}")



# Download the mp3s
@dirname = 0
rss.items.each do |s|
        if s.enclosure.nil? == false
        if Time.now - 86400 < s.date
                #The following puts statements are just for debugging purposes.                @author = s.
dc_creator
                puts @author
                @title = s.title
```

```
                puts @title
                @date = s.date
                puts @date
                @url = s.enclosure.url
                puts @url
                @categories = s.categories
                def contents
                  @array=[]
                  for c in @categories do
                    @array.push(c.content)
                  end
                end
                contents
                puts @array

                #Determine if the file is an mp3, and if so continue
                file = open(s.enclosure.url)
                if MIME['audio/mpeg'].match_file?(file) == true

                @filename = File.basename("#{s.enclosure.url}")
                Dir.mkdir("#{@dirname}")
                Dir.chdir("#{@dirname}")

                open("#{@filename}","w").write(open("#{s.enclosure.url}").read)

                        #Write the Dublin Core
                        File.open("./dublin_core.xml", 'w') do |x|
                                x.puts '<?xml version="1.0" encoding="UTF-8"?>'
                                x.puts "<dublin_core schema='dc'>"
                                x.puts("\t" + '<dcvalue element="title" qualifier="none">' + "#{@title}" + '<
/dcvalue>')
                                x.puts("\t" + '<dcvalue element="contributor" qualifier="author">' + "#
{@author}" + '</dcvalue>')
                                for c in @array do
                                        x.puts("\t" + '<dcvalue element="subject" qualifier="none" language="en"
>' + "#{c}" + '</dcvalue>')
                                end
                                x.puts("<\/dublin_core>")
                        end

                        #Write the contents list
                        File.open("./contents", 'w') do |x|
                                x.puts "#{@filename}\tbundle:ORIGINAL"
                                x.puts"license.txt\tbundle:LICENSE"
                        end

                        #Write the license
                        File.open("./license.txt", 'w') do |x|
                                x.puts "This is a dummy license."
                        end
                puts "\n"
                Dir.chdir("../")
                @dirname = @dirname + 1
                 end


                end
                end

end

go_home

upload(@directory)
```