################################################################################

# Copyright (C) 2015 Surfacingx #

# #

# This Program is free software; you can redistribute it and/or modify #

# it under the terms of the GNU General Public License as published by #

# the Free Software Foundation; either version 2, or (at your option) #

# any later version. #

# #

# This Program is distributed in the hope that it will be useful, #

# but WITHOUT ANY WARRANTY; without even the implied warranty of #

# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the #

# GNU General Public License for more details. #

# #

# You should have received a copy of the GNU General Public License #

# along with XBMC; see the file COPYING. If not, write to #

# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA. #

# http://www.gnu.org/copyleft/gpl.html #

################################################################################

import xbmc, xbmcgui, urllib, sys, time, uservar

import wizard as wiz

ADDONTITLE = uservar.ADDONTITLE

COLOR1 = uservar.COLOR1

COLOR2 = uservar.COLOR2

urllib.URLopener.version = 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36 SE 2.X MetaSr 1.0'

def download(url, dest, dp = None):

if not dp:

dp = xbmcgui.DialogProgress()

dp.create(ADDONTITLE ,"Downloading Content",' ', ' ')

dp.update(0)

start\_time=time.time()

urllib.urlretrieve(url, dest, lambda nb, bs, fs: \_pbhook(nb, bs, fs, dp, start\_time))

def \_pbhook(numblocks, blocksize, filesize, dp, start\_time):

try:

percent = min(numblocks \* blocksize \* 100 / filesize, 100)

currently\_downloaded = float(numblocks) \* blocksize / (1024 \* 1024)

kbps\_speed = numblocks \* blocksize / (time.time() - start\_time)

if kbps\_speed > 0 and not percent == 100:

eta = (filesize - numblocks \* blocksize) / kbps\_speed

else:

eta = 0

kbps\_speed = kbps\_speed / 1024

type\_speed = 'KB'

if kbps\_speed >= 1024:

kbps\_speed = kbps\_speed / 1024

type\_speed = 'MB'

total = float(filesize) / (1024 \* 1024)

mbs = '[COLOR %s][B]Size:[/B] [COLOR %s]%.02f[/COLOR] MB of [COLOR %s]%.02f[/COLOR] MB[/COLOR]' % (COLOR2, COLOR1, currently\_downloaded, COLOR1, total)

e = '[COLOR %s][B]Speed:[/B] [COLOR %s]%.02f [/COLOR]%s/s ' % (COLOR2, COLOR1, kbps\_speed, type\_speed)

e += '[B]ETA:[/B] [COLOR '+COLOR1+']%02d:%02d[/COLOR][/COLOR]' % divmod(eta, 60)

dp.update(percent, '', mbs, e)

except Exception, e:

wiz.log("ERROR Downloading: %s" % str(e), xbmc.LOGERROR)

pass

if dp.iscanceled():

dp.close()

wiz.LogNotify("[COLOR %s]%s[/COLOR]" % (COLOR1, ADDONTITLE), "[COLOR %s]Download Cancelled[/COLOR]" % COLOR2)

sys.exit()