FTPBatch

COLLABORATORS

	TITLE : FTPBatch		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 28, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FTPBatch

1.1 FTPBatch

FTPBatch 4.1 (18.10.98) (C) by Martin Steigerwald
- adds batch download facility to OpusFTP

1.2 Introduction

FTPBatchlister is a package of two scripts. FTPBatchLister opens a lister where you can add FTP URLs by drag and drop from open FTP listers. FTPBatchDownload is a script to download all or selected entry in the FTPBatchLister lister, when you tell FTPBatchLister to start it.

You can even start FTPBatchLister in auto download mode. So you can use a scheduler to start it at night. It then suppresses all requesters and thus runs non-interactively. It will produce a log file then.

1.3 Features

- Full features FTPBatch lister. Works like a hotlist. You can add entries, edit them. Save and load them.
- Able to handle several FTPBatch lists.
- Configurable via requesters.
- Uses an own nice tool bar which is configurable via Opus.
- Fully supports drag and drop, lister button sorting and lister context menus.
- Has an own filetype to support double clicking on FTPBatch lists to open a FTPBatch lister.
- Comes with example buttons on how to use it.
- Sports an non-interactive automatic download mode.
- Has the ability to use up to 8 FTP connections for downloading. It can even use more, but I restricted this. ;-)
- Downloading happens asynchronously and thus doesn't block the FTPBatch lister.
- Downloaded entries can be automatically deleted from the FTPBatch lister.
- When automatically deleting entries and download all entries the download process will check for files you add to the FTPBatch lister after the download process has been started.
  Thus you can started downloading after you added some files and then go on adding files.
  Note: This does not work when an error occurs to avoid endless download loops.

Limits

- Doesn't trap the normal »Copy« or »Move« operation. There is nothing I can do about it.
  So you can only use drag and drop.

- Can't automatically save the FTPBatch list when you close the FTPBatch lister. Limitation of Opus itself. It just closes the window and tells me that after it has been closed.
  I might be able to workaround this when I keep a copy of the list internally to the ARexx script or save each change to it.

- See chapter Future.

1.4 Legalinfo

It is freeware. It is freely distributable non-commercially. Distribute it unchanged and completely.

The contents of the »Buttons«- and the »Filetypes«-directories are public domain. Do with them what you want. You may release different tool bars for FTPBatch on your own. But I appreciate it when you send them to me so that I can include them into the FTPBatch distribution.

You may change the scripts for your personal use. Don't distribute changed versions without my permission. Better send me the changes so that I can include them in future versions when they are good.

THERE ARE NO WARRANTIES, NO WARRANTIES! If it beams you to another planet or eats your hard disc I am not responsible. This is free software.

GP Software may distribute it on their wonderful Opus Plus CD-ROM.

1.5 Requirements

- DOpus Magellan and its OpusFTP

- ARexx

1.6 Installation

Unarchive the archive over the »DOpus5:«-directory. Or copy the extracted archive contents manually to the »DOpus5:«-directory in the same directory structure I use in my archive. Look into the »Filetypes« directory. There is a directory »Deutsch« and »English« for german and english versions of the filetypes. Copy the filetype you desire to the main »Filetypes« directory.

Start it as ARexx script from a button. Look at "Buttons/FTPBatchButtons" for an example how to do it.

Note: The toolbar of FTPBatch uses the images that are installed into »DOpus5:Images«. There are no extra images for it. Also see chapter {"Future" LINK Future}.

Arguments

```
DOpus5:ARexx/FTPBatchlister.dopus5 DOPUSPORT/A LISTERHANDLE/A FILE/A
                                   DOWNLOAD/S

DOPUSPORT       - ARexxportname of Opus, normally "DOPUS.1".
                  Use {Qp} from Opus.

LISTERHANDLE    - 0      = New lister will be opened.

                  The following option is currently not supported:

                  handle = The lister of the given handle will be
                           overtaken.

FILE            - This is the prename of FTPBatch list file,
                  the configuration file and the log file:

                      <FILE>.batch  = FTPBatch list file
                      <FILE>.config = configuration file
                      <FILE>.log    = log file
                  FTPBatch list file and configuration file will be
                  loaded automatically on startup.w

DOWNLOAD        - When this option is given FTPBatchLister will
                  automatically start downloading the files after
                  it opened its lister. FTPBatchDownload will
                  collect all files and the immediately closes the
                  FTPBatch lister.

                  No matter what FTPBatchLister is configured to
                  FTPBatchDownload won't display requesters. It will
                  generate a log file instead.

                  This is good for scheduled downloads without user
                  interaction.
```

Example

```
DOpus5:ARexx/FTPBatchlister.dopus5 {Qp} 0 DOpus5:System/FTPBatch
```

1.7 Usage

Open a FTPBatch lister and open a OpusFTP lister. Drag some files from the OpusFTP to the FTPBatch lister. When you are finished: Save the FTPBatch list if desired and start downloading.

The lister has two columns. One displays the entry name. This is also the filename that is used when you download the file to your local directory. Thus you can change local names indepently from the remote ftp server filenames. The second columns displays the complete URL in the syntax:

ftpsite//directory/file

You can use inline editing to edit name and url of an entry. You can also double click on an entry to edit its name.

You add files by dragging them from ftp listers and droping them to a FTPBatch lister. When you add two files with the same name, one will get a ».1« appened to the local filename. You can't use the copy or move buttons to add files. See chapter Future and Features for more info.

Toolbar

when you want to have a look at it.

The buttons from the left to the right in the format »Function on left mouseclick / function on right mouseclick«.

Save list / load list
- Saves or loads the FTPBatch list. Note: Always save the list, before you close the FTPBatch list window. Currently FTPBatch

can’t save the list when you close the window. See chapter {"Features" LINK Features} or {"Future" LINK Future} for more details.

Select all entries / Deselect all entries

Select entries by pattern / Toggle selection state of all entries

Copy all entries / Copy selected entries

- Downloads all or selected entries to the local directory you configured.
- Doesn’t delete successfully downloaded entries from the list.

Move all entries / Move selected entries

- Downloads all or selected entries to the local directory you configured.
- Deletes successfully downloaded entries from the list.

Edit name / Edit URL

- Allows you to change the name or URL of the selected entries.
- The name will be used as local file name when downloading files.

Edit and save configuration

- Asks you to set the maximum # of parallel connections, to define if should log actions and/or display requesters, and to set the desired local directory the files should be downloaded to. Saves the configuration.
- The configuration is loaded automatically when you start FTPBatchLister.

Delete

- Deletes selected entries from the list.

File context menus

Delete

- Deletes the entry from the list.

Edit Name
Edit URL

Path gadget

Shows the filename prefix for the FTPBatch list file, the config file and the log file. When you change it, it will use the new filename prefix and try to load the according list and config file.

This way you can easily create a second batch list. You can also open more than one FTPBatch lister and thus have a batch list for AmiNet® downloads and one for GeekGadgets downloads for example.

1.8 Commands

FTPBatchLister implements a handler which adds its own commands locally to the FTPBatch lister. Normally you do not need to care about them. But when you want to change the FTPBatch tool bar to your needs it might be good to know what commands FTPBatch supports. Note: Leave FTPBatch running when trying to change the tool bar. Opus unfortunately won’t accept when you try to set a internal command of FTPBatch to a button function when FTPBatch does not run. The internal commands of FTPBatch won’t be displayed in the command list of the function editor.

```
FTPBatchConfig
    LOAD     - Loads the configuration file
    SAVE     - Saves the configuration file
    EDIT     - Let the user set new options.
    EDITSAVE - Let the user set new options and save them.

FTPBatchDownload
    Starts FTPBatch download.

    First argument:
    ALL      - Downloads all files
    SELECTED - Downloads selected files

    Second argument:
    When you add »DELETEENTRIES« as second argument successfully
    downloaded items will be deleted from the FTPBatch lister.


FTPBatchEditEntry
    NAME     - Let the user edit the name for selected entries.
    URL      - Let the user edit the URL for selected entries

FTPBatchList
    LOAD    - Load the FTPBatch list.
    SAVE    - Save the FTPBatch list.
```

1.9 Background

How does it work?

FTPBatchLister does all the FTPBatch list handling and the configuration. It implements a handler for all this work. For FTP batch downloading it starts FTPBatchDownload with all the necessary parameters.

FTPBatchDownload will then look into the FTPBatchLister lister and collect the files it should download. After this the lister is not needed anymore. Thus when you are in automatic download mode, the lister of FTPBatch will be closed right now.

After this FTPBatchDownload decides how many connections it should use. When you have selected 4 connections at a time and there is only one file to download only 1 connect will be established. When it uses 1 connection it switches into single download mode, otherwise it calls the marvellous DownloadThemAllAtOnce parallel heavily multitasking download routine. ;)

It will only connect to a new ftp server and change the current directory of a ftp lister when it is necessary. It is only necessary when the new entry to download has a different ftp server or directory than the old one. So its the best, when you change the sort order in the FTPBatchLister lister to sort the comment, the FTP URL field alphabetically.

In parallel download mode it opens a ftp lister for each connection. Then it goes into a loop:

- Check for next non busy lister
- Check its status
  - Ready? Connect to new ftp site if necessary.
  - Connected? Set a new current directory if necessary.
  - Correct directory set? Start download.
  - Download ended? Do file checking.
- Repeat loop until all FTP listers are ready and all file are downloaded.

So, enough about this. Interested ARexx programmers may look into the scripts themselves.

1.10 Future

FTPBatch can be enhanced. I for myself didn’t use it a single time in real live situation. I only used it with my local ftp server for testing. Maybe I have a use for it rarely but not often.

This means: I don’t care about enhancing it to much unless you request it. The future of this script depends on you.

I most probably won’t implement any of these - except Hyperguide documentation - unless I get some feedback.

Possible new features

- Localization
- Installer script
- Appicon support
- New tool bar with one nice NewIcons or MagicWB icons. This is something for a icon specialist.
- HyperGuide & PowerGuide documentation to support images in Usage section.
- Checking for file sizes to detect bad downloads in parallel mode. Can slow down FTPBatchDownload as I have to use scandir or a direct ftp command to obtain the size of the remote file.
- Signal handling to also handle wierd error situation. However my scripts behave nicely here.
- Cleaner parallel download mode. See below.
- Handling of directories.
- Handling of non-anonymous downloads.
- Add automatic saving of the list when the user closes the FTPBatch lister.

Cleaner parallel download mode

At the moment the download in several tasks is handled by one script:

FTPBatchDownload.rexx. I call this current approach. I can change FTPBatchDownload to start a FTPBatchDownloadFile script for each download instead.

Advantages:

- I may get rid of the DELAY()-ed busy loop in FTPBatchDownload. Thus I may get cleaner multitasking behaviour.
  With the current approach this will only be possible with a LISTER WAIT command that waits for mutiple listers. I don’t know whether this will ever be implemented in Opus.
- I can run FTP commands synchronously. This means checking for correct downloads is done by checking the return code of the commands. This is the safest way, and way better then the current approach to check for the existance of the local file and the »Transfer Aborted« comment.
  With the current approach this is only possible, when OpusFTP commands copy the RC variable to a lister variable. I don’t know whether this will ever be implemented into Opus.
- Requesters won’t block FTPBatchDownload anymore. Only the according FTPBatchDownloadFile process will wait for the answer of the user. This means all other download processes can go on with their work.
- The code for downloading a file can be shared between the single and the parallel download mode.

Disadvantages:

- One more task for each download process
- Many accesses to the DOpus5:ARexx directory to start the FTPBatchDownloadFile script. Unless I copy this script to RAM: before.
- Needs quite major cut&paste rewrite of the whole script.
- Some code for requester and log file handling will be duplicated then as it has to be in FTPBatchDownloadFile as well.

1.11 Author

Donations appreciated! Please no chocolate or beer as I only eat vegetable raw food and only drink water. ;) Ask me for my snail mail address when you want to sent me a donation.

Martin Steigerwald
Email: steigerw@stud.uni-frankfurt.de
WWW: http://www.rz.uni-frankfurt.de/~steigerw
       http://home.pages.de/~helios
IRC: helios@#amigager sometimes

1.12 Credits

A big thank you to Dave Clarke, who helped me a lot. I looked often into his scripts, especially "ArcDir.dopus5", to get a glue. And I often read his excellent DOpus Magellan II ARexx Guide. Great Work!

I added a nice credit to Dave Clarke here. He gave me his FTPBatch
ARexx script so that I can include the good ideas in it in my script. I wanted to
thank him even more for that. And what did he do? Look at this
damn credit section ;-). (The link works only when his ARexx Tutorial is
in DOpus5:help. If it doesnt work open the ARexx tutorial manually when
you want to see the truth.)

Thanks go to Neil Bothwick for feedback and his nice FTPBatchAdd ARexx
script.

Thank you to, Greg Perry, for motivating me to do this script actually. I
didnt know it is actually not that hard to code something useful for Opus.

Thank you, Jon Potter and Greg Perry and Andrew, for writing DirectoryOpus
and OpusFTP that made this script possible.

Thanks to all FTPBatchlister betatesters, all Opus betatesters, all
friends.