Charon

COLLABORATORS			
	TITLE : Charon		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 19, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Charon

1.1 Charon.guide

Charon
******

This is the documentation for Charon v1.0, a download manager for the Amiga. Copyright (C) 2000 Andrija Antonijevic. All rights reserved. Program and documentation by Andrija Antonijevic.

1.2 Charon.guide/Legal information

Legal information
****

Charon

Copyright (C) 2000 Andrija Antonijevic. All rights reserved.

The producer of this software will grant the Licensee a limited, non-exclusive right to use the program Charon and its associated files on a single machine. The producer will provide the Licensee with a special identification file which is called the key file. This key file will enable the Licensee to fully use the Charon package according to the terms of this license.

The key file is stricly for personal use by the Licensee only. Any disclosure of they key file will invalidate the license and the key file itself. The Licensee will be held liable of any damage arising out of the disclosure of the key file.

THIS SOFTWARE PACKAGE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE USE OF THIS SOFTWARE PACKAGE IS ASSUMED BY THE USER. IN NO EVENT WILL THE AUTHOR BE LIABLE FOR ANY DAMAGES, DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL, RESULTING FROM USE, MISUSE OR INABILITY TO USE THIS SOFTWARE PACKAGE EVEN IF IT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

No part of this archive may be redistributed except with the entire contents of the original archive.

Aminet is explicitly allowed to distribute this archive on its CDs.

1.3 Charon.guide/Introduction

Introduction
****

One of the Internet programs that Amiga was missing was a good download manager. Or just any download manager, for that matter. As author of HTTPResume, I got quite a few suggestions to make something "like GetRight or Go!Zilla on the PC". So, as the need was obvious, I slowly started working on it from the scratch.

Charon is a download manager, plain and simple. You can use your browser to download files, but they (mostly) don’t support resuming and don’t have enough options to finetune the process. In order to have full control over downloads, you need a download manager: something designed for that specific purpose. This Charon (pronounced something like Kerohn), unlike mythical Charon who transfers souls of the dead over the river Styx to the underworld, transfers files over the Internet to your HD.

What does it do?

- Multiple download of HTTP/HTTPS/FTP files all with/without proxy, with resume (if server supports it). HTTPS support using AmiSSL or MiamiSSL.

- Automatic retry/resume of downloads up to specified number of tries.

- Huge number of options to customise it as you wish including: setting timeout after which download is retried, setting minimum allowed number of CPS, maximum number of CPS...

- Scheduled downloads.

- It can automatically choose proxy based on your hostname or IP address.

- ARexx interface.

- MUI GUI.

1.4 Charon.guide/Registration

Registration
****

Charon is shareware. You are allowed to use evaluation version for 30 days and than you must either register or stop using it. Registered users will receive their personal key file which will enable them to use all features of Charon. To see how to register, please visit Charon home page (see Support).

Evaluation version has following limitations:

- Only one copy of Charon can be started.

- There’s maximum of two simultaneous downloads at the same time.

- "Max CPS" option can’t be switched off and the fastest download speed is limited to 500 CPS.

- About window is opened at the start of the program and can’t be closed until ten seconds have passed.

1.5 Charon.guide/Requirements

Requirements
****

- An AmiTCP compatibile TCP/IP stack such as Miami or Genesis.

- AmiSSL or MiamiSSL packages for HTTPS transfers.

- MUI for graphic interface and some MUI custom classes.

1.6 Charon.guide/Installation

Installation
****

To install Charon, use supplied Installer script. Basically, it will copy the whole archive to a directory of your choice and let you select what kind of pictures do you want to use for Charon toolbar. You should already have NList MUI custom class (which can be found on Aminet in dev/mui directory) in your MUI:Libs/MUI directory for Charon to work.

1.7 Charon.guide/Menus

Menus
*****

Here's a short description of all menu items:

Project/URL/New...
Opens a window in which you can enter new URL.

Project/URL/Start
Starts downloading selected URLs. If there's already maximum number of transfers going on, the one you started will be queued. You can't start locked URLs (the ones that are underlined).

Project/URL/Stop
Stops downloading selected URLs.

Project/URL/Edit...
Opens windows that enables you to edit selected URLs. You can't edit locked URLs.

Project/URL/Remove
Removes selected URLs from the list. If necessary, some downloads will be stoped.

Project/URL/Schedule
Schedules URL selected URLs for download at specified time. You can't schedule locked URLs.

Project/Import list...
Imports list of URLs to download from a file. This file is the one you get when you choose Export list... from the Project menu.

Project/Export list...
Exports list of files in main window to a file in format that Charon understands.

Project/About...
Opens a window with information on Charon.

Project/About MUI...
Opens a window with information on MUI.

```
Project/Iconify...
     Hides the Charon window either in an icon on the Workbench screen
     or in the Workbench "Tools" menu (selectable in MUI preferences).

Project/Quit
     Exits Charon. All downloads will be stoped and the list of URLs
     will be exported to PROGDIR:Charon.list.

Settings/Settings...
     Opens the settings window.

Settings/Save settings as...
     Saves the settings under a filename that you choose.

Settings/Save settings as default
     Saves the settings as PROGDIR:Charon.cfg.

Settings/Load settings...
     Loads settings from a file that you choose.

Settings/Load default settings
     Load the settings from PROGDIR:Charon.cfg.

Settings/MUI...
     Opens the MUI settings for Charon.

Settings/Execute ARexx script...
     Starts a specified ARexx script.
```

1.8 Charon.guide/Settings

```
Settings
********

   Settings window has five pages. Names in the brackets next to option
names are significant only to ARexx programmers.

Global
     These are global settings that affect every URL or behaviour of
     the program itself.

        * Base directory (STARTDIR) When you enter URL, Charon will try
          to create output file name using this value and filename from
          URL. For example, if you set base directory to Downloads: and
          enter URL http://www.somehost.net/file.lha, Charon will
          automatically fill in Downloads:file.lha as output file name.
          Note that you have to press enter in URL field for Charon to
          do this.

        * Error sound (ERRORSOUND) This is the sound Charon will play
          when error is encountered during download of the file.

        * Finished sound (FINISHEDSOUND) This is the sound Charon will
```

play when a transfer is finished.

- Finished all sound (FINISHEDALLSOUND) This is the sound Charon will play when all started/queued transfers are finished.

- Convert from (CONVFROM) and Convert to (CONVTO) These options can be used to convert some characters in the filename. For example, setting Convert from to "a" and Convert to to "b" would convert all letters "a" to "b". This is case sensitive! If lengths of these values are not equal, both values will be cleared (it is not possible to check this from ARexx, so be careful when you do this)! Common uses for this is to change spaces to underscores or something similar...

- Max transfers (MAXTRANSFERS) This sets maximum number of transfers that Charon will do at the same time. If more transfers are started, they will be queued.

- Max filename length (MAXNAMELEN) This lets you limit length of the filename. Note that if you set this to x characters, Charon will copy first x - 3 characters and use URL to calculate last three characters. This should make sure that you can use maximum filename length that your filesystem allows and that http://some.host.net/verylongfilename1 and http://some.host.net/verylongfilename2 don't end up with the same filename.

- No safety requesters (NOSAFEREQUESTERS) This disables those "Are you sure?" requesters which are annoying to some people.

- Auto start (AUTOSTART) If set, automatically starts downloading URL as soon as you enter it (or queues it). Also, when you start Charon, it will automatically start/queue all downloads (except scheduled ones).

- Auto exit (AUTOEXIT) If all URLs are downloaded (there are no more entries in the list) and this option is set, Charon will exit.

- Don't load list (NOLOADLIST) Normally, Charon loads list of URLs that it didn't download the last time. If you don't want this to happen, you can set this option.

- Don't save list (NOSAVELIST) If you set this, Charon won't save URLs that were left in the list on exit.

- Delete files (DELETEFILES) If you think that Remove URL should remove not only URL(s) from the list, but also actual file(s) from the disk, set this option.

Proxies

These settings allow Charon to automatically choose proxy based on your hostname or IP address. When Charon wants to find out which proxy to use, it starts from the beginning of the list and checks if your hostname/IP address matches pattern in the left column of the list. If it does, Charon will use the corresponding proxy from the right column of the list. Proxy list (PROXIES) string gadget should contain name of the file where these settings will be saved.

Local

Here you can set default settings for every URL. But, unlike global settings, every URL can have these settings changed.

- Download attempts (DOWNLOADATTEMPTS) How many times should Charon try to download this URL before it gives up?

- Download delay (RETRYDELAY) Specifies how many seconds should Charon wait before it retries download of this URL.

- Timeout (resumable) (TIMEOUTRES) If resumable download stops for this many seconds, Charon will retry it. Zero means no timeout.

- Timeout (not resumable) (TIMEOUTNRES) If non-resumable download stops for this many seconds, Charon will retry it. Zero means no timeout.

- Max timeouts (MAXTIMEOUTS) One timeout is not regarded as error. If Max timeouts occur, it will be regarded as one error.

- Max CPS (MAXCPS) If you are f. ex. chatting on IRC and you don't want all bandwidth eaten up by file downloads, you can limit the speed of transfer.

- Min CPS (resumable) (MINCPSRES) If resumable download gets too slow, Charon will stop it and retry.

- Min CPS (not resumable) (MINCPSNRES) If non-resumable download gets too slow, Charon will stop it and retry.

- Ignore CPS (IGNORECPS) Some servers need a few seconds to really start transfering files properly. In order to prevent Min CPS option to stop download too early, before it actually got started, this options specifies how many seconds to ignore Min CPS option.

- Directory structure (TREE) This allows you to preserve directory structure from the URL. Setting this option to 2 and trying to download http://www.aminet.net/pub/aminet/comm/irc/amirc.lha will make Charon will skip two directories and append /comm/irc/amirc.lha to Base directory. Directories that don't exist will be created! If this option is set to -1, it will be ignored.

- No proxy (NOPROXY) Don't use proxy even if it's specified.

- Overwrite (OVERWRITE) Always start downloading file from scratch, never try to resume.

- Don't keep date (NOKEEPDATE) Charon normally tries to find out date of the file on server and keep it the same on the local file. If you don't want that to happen, switch this on.

- Force comment (FORCECOMMENT) When Charon downloads a file, it writes the URL to its comment, but only if it didn't already have it. If you want Charon to always write the comment to the output file, you should set this option.

- Owerwrite if no resume (OVERWRITEIFNORESUME) Charon will always try to resume interrupted transfers. However, some servers don't support resume and Charon will then report fatal error. If you want it to restart downloading file from scratch instead, set this option. Overwrite will automatically be set in this case.

- Read buffer (READBUFFERLEN) This is the buffer that Charon uses for reading data from network. If you have a modem connection (upto 56800), there's no point in increasing this value from default 10 kB. On the other hand, if you have faster connection (ISDN or similar) you will probably want to increase this value. ARexx programmers, you will have to get/set this value in bytes. See also DOS buffer option.

- Write buffer (WRITEBUFFERLEN) This is the buffer that Charon uses for sending data to network. The default 1 kB is quite sufficient and you don't need to increase it, no matter how fast your connection is. ARexx programmers, you will have to get/set this value in bytes.

- DOS buffer (DOSBUFFERLEN) This is the DOS buffer that Charon will set for the file it's downloading. If you think that Charon is writing to disk too often, increase this value. If you increase length of the buffer, the upside is that Charon will write to disk less often and in larger chunks. But, if you set it too high (say 100 kB) and your computer crashes for some reason, contents of that buffer won't be written to disk! This option is not available if you don't have OS3.1 or higher. ARexx programmers, you will have to get/set this value in bytes.

HTTP

Everything said for Local settings applies here to. These are some settings specific to HTTP transfers (this also includes those that go through proxy).

- Proxy (PROXY) Proxy server. Most of the time it's something like http://proxy.mydomain.net:8080/ or similar.

- Referer (REFERER) This sets Referer: HTTP header field. Some sites allow download only if link to the file was on their HTML page. When you click on a link on a page, browsers send URL of that page in Referer: field (unless you turn it off). Charon doesn't send Referer: field by default, but if you find a link that requires this field, use this option to set it. This field will always be sent until you clear it. Typically, Referer: field for f. ex. http://www.host.com/dir/dir1/dir2/file.ext should be http://www.host.com/ or http://www.host.com/dir1/dir2/. See also Auto Referer: field option.

- User-Agent (USERAGENT) This sets User-Agent: HTTP header field. You should very rarely need to change this. Some servers send the same HTML pages to browsers differently depending on User-Agent: field ("This works in Netscape, this in Explorer..."). If you don't know anything about this, you don't need to use it.

- No cache (NOCACHE) There are some stupid proxy servers that don't check if the copy of the file they have cached is up to date. In that case, you can force them to fetch it from the original server again.

- Date check (DATECHECK) If you want Charon to check whether partial copy of the file is older than the one on the server (f. ex. if the copy on the server was changed in the mean time) you can switch this on. However, because of daylight savings time, inaccurate clocks and many other things, this might not work as expected.

- Only if updated (ONLYIFUPDATED) If you are making a mirror of a site, you will probably want to download a file only if it's updated, ie. if your copy is older than the one on the server. In that case you should switch this option on. This option is valid only in combination with Overwrite option! Also, you shouldn't use Don't keep date option.

- Auto Referer: field (AUTOREFERER) If this is set, Charon will automatically change and send Referer: HTTP field with every request. This might be neccesary since many servers now allow download only if request for the file has appropriate Referer: field, otherwise they usually redirect you to their main page. See also Referer: option.

FTP

Everything said for Local settings applies here to. These are some settings specific to FTP transfers (those that don't go through proxy).

- No "PORT" Charon normally sets up FTP transfers using PASV command which has some advantages over PORT command. In case PASV doesn't work, Charon will fall back to PORT command. However, if you are on a firewall, this definitely won't work and will block for some time. If this happens to you with every FTP server, switch this on. Note that there are some firewalls whose purpose is not to let you use direct FTP downloads.

- FTP e-mail Anonymous FTP servers require you to send your e-mail address as password. However, you shouldn't need to use your real e-mail address, you can put anything you want here. Some servers perform more or less complex checks, so sometimes you might need to use at least domain part of your e-mail address (after "@" sign).

1.9 Charon.guide/ARexx interface

ARexx interface
****

Charon's ARexx port is called CHARON.x where x depends on number of instances of Charon that are started. Entries in the list are references by IDs which can be gotten in numerous ways. In addition to standard MUI ARexx commands, Charon defines the folowing commands:

NEWURL - Enters new URL into list
```
     Template: NEWURL URL/A,FILE,LOCK/S
```

Inserts new URL into list. RESULT will have ID of the new URL or 0 if it couldn't have been created. FILE is the output file. If it is not given, Charon will try to construct it using Base directory and filename from URL. LOCK optionally locks URL which prevents it from being edited, started (f. ex. by AUTOSTART option).

START - Starts download of specified URL(s)
```
     Template: START ID/N,SELECTED/S,ALL/S
```

Starts specified URL(s). If you want to start specified URL you should specify its ID. SELECTED switch starts only selected URLs, whilst ALL switch starts all URLs. Locked URLs can't be started.

STOP - Stops download of URL
```
     Template: STOP ID/N,SELECTED/S,ALL/S
```

Stops specified URL(s). Options are the same as in START command.

EDIT - Opens edit window for specified URL(s)
```
     Template: EDIT ID/N,SELECTED/S,ALL/S
```

Edits specified URL(s). Options are the same as in START command.

REMOVE - Removes specified URL(s)
```
     Template: REMOVE ID/N,SELECTED/S,ALL/S,FORCE/S
```

Stops specified URL(s). FORCE won't ask user for confirmation even if "No safety requesters" is not checked. Other options are the same as in START command.

SCHEDULE - Schedules specified URL(s)
```
     Template: SCHEDULE ID/N,SECONDS/N/A,SELECTED/S,ALL/S
```

Schedules specified URL(s) to a date which is SECONDS seconds after January 1st 1978. See also GETNUMSECONDS ARexx command.

IMPORTLIST - Imports list of URLs
```
     Template: IMPORTLIST NAME/A
```

Imports list of URLs in format that Charon understands. NAME is the name of the file which contains the list.

EXPORTLIST - Exports list of URLs

Template: EXPORTLIST NAME/A

Exports list of URLs to a file named NAME.

LOCKURL - Forbids changes to/starting/scheduling etc. of URL(s)
Template: LOCKURL ID/N/A

Locks URL with specified ID. This prevents it from being changed, scheduled or started etc.

UNLOCKURL - Removes the lock from URL(s)
Template: UNLOCKURL ID/N/A

This command is reverse to LOCKURL command.

GETFIRSTURL - Returns ID of the first (selected) element in list
Template: GETFIRSTURL SELECTED/S

Returns ID of the first URL (or first selected URL) or 0 if list is empty.

GETNEXTURL - Returns ID of next (selected) element in list
Template: GETNEXTURL ID/N/A,SELECTED/S

Returns ID of the URL which is after the one with ID ID (or of the next selected URL) or 0 if it’s the last one in the list. If you specify "SELECTED", the ID you pass must be of a selected URL.

GETPREVURL - Returns ID of previous (selected) element in list
Template: GETPREVURL ID/N/A,SELECTED/S

Returns ID of the URL which is before the one with ID ID (or of the previous selected URL) or 0 if it’s the first one in the list. If you specify "SELECTED", the ID you pass must be of a selected URL.

GETLASTURL - Returns ID of the last element in list
Template: GETLASTURL SELECTED/S

Returns ID of the last URL (or last selected URL) or 0 if list is empty.

GETURLSTATE - Returns state of the URL
Template: GETURLSTATE ID/N/A

Returns state of URL with specified ID. Possible responses are "LOCKED", "READY", "QUEUED", "DOWNLOADING", "ERROR" and "SCHEDULED".

QUERY - Returns state of some options for URL
Template: QUERY ID/N/A,OPTION/A/F

Queries state of a URL option with specified ID. ID of 0 specifies default options (ie. default ones for new URL). Also, use 0 to get global options (MAXTRANSFERS etc). For possible values for OPTION, see settings. Other possible values are:

* VERSION Returns version of Charon.

* STARTEDTRANSFERS Returns number of currently started transfers.

* NUMTRANSFERS Returns number of entries in the list.

* URL Returns URL of the specified transfer.

* FILE Returns name of the output file of the specified transfer.

* SCHEDULE Returns how many seconds from January 1st 1978 is transfer scheduled after.

SET - Sets some of the options for URL
Template: SET ID/N/A,OPTION/A/F

Sets state of a URL option with specified ID. ID of 0 specifies default options (ie. default ones for new URL). Also, use 0 to set global options (MAXTRANSFERS etc). You can’t change locked or the one that is being downloaded. For possible values for OPTION, see settings. Other possible values are:

* URL Sets URL of the specified transfer.

* FILE Sets name of the output file of the specified transfer.

GETNUMSECONDS - Returns number of seconds that passed up to specified date
Template: GETNUMSECONDS DATE

Returns number of seconds that passed between January 1st 1978 and given date. Empty DATE is the same as current date. Otherwise, it should be in format day:month:year:hour:minute:second. F. ex. if you want to be the first one who asks for a transfer in the next millenium, you would ask for GETNUMSECONDS 1:1:2001:00:00:00 and forward the result to SCHEDULE command.

FINDURLBYID - Checks whether URL with given ID is in the list
Template: FINDURLBYID ID/N/A

Finds URL with specified ID and returns its position in the list (starting from 1) or 0 if it was not found. Useful if you want to see if URL with the ID you have is still in the list.

1.10 Charon.guide/Tips and tricks

Tips and tricks
****

* Edit URL window is an app window. You can drop icons on it and Charon will use name of that file. If the file has URL in its comment, that comment will be copied to the URL field.

```
   * Edit URL window has a menu which will let you load a .cfg file.
     You can create a .cfg file if you choose Save settings as... entry
     from Settings menu.

   * When you enter URL in edit URL window, press enter for Charon to
     automatically create filename from the URL.

   * You can rearrange order of the downloads using drag & drop in the
     listview.

   * Use provided ARexx script to send links from browsers to Charon.

        * In AWeb, go to Settings/GUI settings/Pop-up and an entry to
          each type. The name of the command can be f. ex. Download and
          the command should be SYSTEM CharonPath:Rexx/AddURL.rexx
          "%u". This will download files to your default download
          directory.

        * In IBrowse 2.2+, go to Preferences/Settings.../GUI/FAB menus,
          then drag Custom entry to each Link, Imagelink, Page and Image
          sections (you can then rename it if you want). Select ARexx
          in the cycle gadget and select AddURL.rexx script. In the
          argument field enter "%i", "%l" or "%u" to download image
          URL, link URL or page URL respectively.

        * If anyone has any information how to do this in Voyager,
          please let me know.
```

1.11 Charon.guide/History

```
History
*******

Version 1.0
        * First public release.
```

1.12 Charon.guide/Support

```
Support
*******

Mailing list
     For discussion about Charon you can join the Charon support mailing
     list. To join, send an e-mail to charon-list-subscribe@onelist.com.

WWW support site
     Visit Charon WWW site at http://www.bigfoot.com/~TheAntony/Charon/
     to get updates and support.
```

1.13 Charon.guide/Acknowledgements

Acknowledgements
****

I would like to thank

* Cedomir Pavlovic for graphics.

* Stefan Stuntz for MUI.

* Gilles Masson for NList MUI custom class.

* Ljubomir Jankovic for betatesting.

* Holger Kruse for MiamiSSL SDK.

* All registered users of Charon.

1.14 Charon.guide/Index

Index
*****