

**split\_n\_build**

COLLABORATORS

	TITLE : split_n_build		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>split_n_build</b>	<b>1</b>
1.1	Split and Build version 1.1 . . . . .	1
1.2	disclaimer . . . . .	1
1.3	copyright . . . . .	1
1.4	WHAT HAVE WE GOT ? . . . . .	2
1.5	WHAT DO THESE PROGRAMS NEED? . . . . .	3
1.6	INSTALLING . . . . .	3
1.7	WHY COULD YOU NEED THEM? . . . . .	3
1.8	USING SPLIT . . . . .	3
1.9	USING BUILD . . . . .	4
1.10	HISTORY . . . . .	4
1.11	ABOUT LOCALE . . . . .	5
1.12	SPL0/SPL1 File Format . . . . .	5
1.13	KNOWN BUGS . . . . .	6
1.14	HOW TO CONTACT THE AUTHOR . . . . .	6

# Chapter 1

## split\_n\_build

### 1.1 Split and Build version 1.1

```
Split version 1.1
Build version 1.1
© 1995 by Stefano Reksten
of 3AM - The Three Amigos !!!
```

CONTENTS OF THIS FILE:

```
DISCLAIMER
COPYRIGHT and CARDWARE
WHAT HAVE WE GOT?
WHAT DO THESE PROGRAMS NEED?
INSTALLING
WHY COULD YOU NEED THEM?
USING SPLIT
USING BUILD
HISTORY
ABOUT LOCALE
SPL0/SPL1 FILE FORMAT
KNOWN BUGS AND INCOMPATIBILITIES
HOW TO CONTACT THE AUTHOR
```

### 1.2 disclaimer

```
-----
The author is NOT responsible for the suitability or accuracy of this
documentation and/or the program(s) it describes. Any damage directly or
indirectly caused by the use or misuse of this documentation and/or the
program(s) it describes is the sole responsibility of the user her/him self
-----
```

### 1.3 copyright

---

---

Split v1.1 and Build v1.1 are Copyright © 1995 by Stefano Reksten. All rights reserved. These two programs may be distributed non-commercially and providing that executable, source code, documentation and copyright notices remain unchanged and are included with the distribution.

The archive must contain the following files:

```
Split_v1.1/
  Catalogs/
    Italiano/
      Build.catalog
      Split.catalog
  Sources/
    Catalogs/
      Build.cd
      Build_Italiano.ct
      Split.cd
      Split_Italiano.ct
    askfile.c
    build.c
    commodity.c
    diskhandler.c
    doslist.c
    gadgets.c
    handler.s
    locale.c
    main.c
    smakefile
    split.c
    split.h
    window.c
  Build
  Build.info
  Docs.guide
  Docs.guide.info
  Split
  Split.info
Split_v1.1.info
```

Please note that the Sources directory comes in an archive.

This program is CARDWARE. If you use it you \*MUST\* send me a postcard from your city/country. He who uses this program not having sent me a postcard can be prosecuted by diarrhoea, data loss, expired driving licence. I aint asking a lotta money from your pockets, just a stamp! :-)

---

## 1.4 WHAT HAVE WE GOT ?

---

Simple - a system for splitting and rebuilding big files. But... why would you need this if there are other ways for files splitting and transferring? Well a reason would be its GUI (I am somewhat proud of that) and because it's FREE.

---

---

## 1.5 WHAT DO THESE PROGRAMS NEED?

---

Any Amiga with kickstart 2.04 should be enough (if it is not LET ME KNOW!). If you want to use their catalogs you must have them installed, and you must have locale.library 38+! :-)

---

## 1.6 INSTALLING

---

Well, it is not so difficult... Just drag the Split\_v1.1 drawer wherever you would like to have it.

---

## 1.7 WHY COULD YOU NEED THEM?

---

Try to bring a 12 meg file from one Amiga to another! You can use ParNet IF you have the cable :-) a backup program if you have one but these generally require to write rawly on disks' tracks... or a FREE splitting system like this - just fill every disk you have to the last byte :-) Split has a nice comfortable interface that shows you volumes (with free space), allows you to split files with choosen chunk dimension and offset, updates the volumes list as disks are inserted, and more.

---

## 1.8 USING SPLIT

---

Quite easy. Select a file by pressing the "File" gadget. You will see its name and size shown in the lower part of the window. (Status won't change from Idle until you write the first chunk of the file.) All you have to do is split that file. To do that you must choose a destination, chunk sizes and offsets. Don't worry - all this can be done automatically. There is a listview in the upper part of the window; you can read in it which mounted volumes are in the system and how much free space they have. If you double click on a volume, its space \*minus some bytes for the file header and some bytes for sector linking (required by DOS)\* will become the chunk size. Chunk size and chunk offset are checked to be sure they are consistent i.e. chunk offset isn't greater than file's size, or chunk's offset + dimension aren't bigger than the whole file. Note that even if a disk's space changes chunk size will \*remain the same\*. Note also that if you specify a n bytes chunk size you will get a \_n+48 bytes\_ chunk file. Those 48 bytes keep the information needed to rebuild the file. Choose a path and write the chunk.

---

Repeat this until the file has been split and the status gets back to Idle.

Now let's get more into it. Sometimes it can be useful to split a file FROM a certain offset. Well, that's what the size and offset gadgets are for! If you fill them with your values you'll get a file splitted as you need. The program will check that chunks don't overlap, and keep track of written chunks - so if you forget something just press the "Chunk" gadget, and you will get the correct size and offset for the next chunk to be written.

Split is a commodity, that means it can be quitted with Exchange, with the "Quit" menu entry or by sending a CTRL\_C. It also has a popkey. (By default ctrl-alt-f.) Split's window's initial coordinates can be specified via the WLEFT and WTOP tooltypes.

-----

## 1.9 USING BUILD

Well this one is very simple. Build needs to get any chunk file of the file you want to rebuild (not necessarily the first one, anyone will do). Then, a destination path is needed. After that, Build will ask you to select the remaining files one by one; once again it is not necessary to specify them in order. Build will copy them the correct way. End.

Build's template is CHUNK/K,PATH/K,OTHER\_CHUNKS/M. From CLI you can specify the files from that or just leave something unspecified - an ASL requester will pop up and ask you to choose something for the missing voices. The ASL requester will appear immediately if Build is launched from Workbench.

CHUNK is any of the file's chunks, PATH is the path in which the file must be rebuilt, and OTHER\_CHUNKS is a list of the remaining chunks - if any of them is missing an ASL requester will pop up.

Build's window's initial coordinates can be specified via the WLEFT / WTOP tooltypes.

-----

## 1.10 HISTORY

-----

Mar 9, 1995: v1.0 - First public release.

Mar 12, 1995: Font sensitiveness added (refers to default system font).

Mar 22, 1995: GUI added to Build, bug removed from Split.

Jul 4, 1995: WLEFT and WTOP tooltypes added. A little bug in Split removed (forgot to ArgArrayDone()).

Jul 27, 1995: Removed a bug in system font managing.

Sep 3, 1995: All the icons support the NewIcon package by Nicola Salmoria.

Nov 12, 1995: v1.2 - Font sensitiveness improved, SPL1 chunks added.

-----

## 1.11 ABOUT LOCALE

-----

As I think it's real fun and nice to have a program that speaks my language I enclosed some catalogs. If you see that your beloved language is not here with the others, you can do the following actions:

- Get your language's .ct file in the Sources/Catalogs directory.
  - Fill in the entries.
  - Send it back to me and I will compile and enclose it in the archive.
- 

## 1.12 SPL0/SPL1 File Format

-----

I am enclosing here the file format of SPL0 and SPL1 chunks, just in case someone would like to make a Mac/MSDOS/Windows version of this program, to help other people carry their big precious files thru' various filesystems.

SPL0 format:

Every chunk has a header of this type:

```
struct Header
{
    ULONG h_Identifier;      /* "SPL0" */
    char h_FileName[32];     /* original filename (no path) */
    ULONG h_FileSize;        /* original file size */
    ULONG h_ChunkStart;      /* offset from beginning of file */
    ULONG h_ChunkEnd;        /* offset from EOF */
};
```

$h\_ChunkEnd - h\_ChunkStart$  gives length of this chunk.

All ULONGs have MSB first. Sorry for the INTEL users but this is the Internet standard :)

(For those of you who don't know it, MSB=Most Significant Byte, i.e. \$DEADBEEF is stored in memory as the sequence '\$DE, \$AD, \$BE, \$EF' in 4 consecutive bytes. INTEL based machines would store the same thing with the sequence '\$EF, \$BE, \$AD, \$DE'. It's quite useless to start discussing which is the best. My program uses the first.)

SPL1 format:

Header is extended to this:

```
struct ExtHeader
{
    struct Header exth_Header; /* "SPL1" */
    ULONG exth_Date;           /* Splitting date of the file in */
};                             /* seconds. Just to be more sure */
                             /* of WHAT we're processing. */
```

-----



## 1.13 KNOWN BUGS

-----  
None known. I really hope not to get any bug report concerning files... It wouldn't be nice to know I destroyed other ppl's precious archives ;-)  
Just joking, I tested them on mine!

Does \*NOT\* support MAC filesystem. At least, it may work well, but I do not think so. Double clicking on the volumes' list could report you a chunk size too big, or too small. This bug will be corrected as I get docs on MAC file system, and a MAC drive (or a high density disk and Consultron's program).  
-----

## 1.14 HOW TO CONTACT THE AUTHOR

-----  
I tested this program with a MMU on my accelerated A1200, and it seemed to work correctly. If you find any bug or have some brilliant ideas or want to tell me how much time I am losing doing stupid proggies like this, or anything else, write to:

rekststef@unisi.it

Stefano Reksten c/o Naimi  
v.le Cavour, 40  
53100 Siena  
ITALY  
-----

<EOF>