

RxDOS version 7.24 notes

This document (2008-2018 by C. Masloch) is in the Public Domain.

This document has been sloppily compiled on 2018-10-09, release 2018-10-08. WARNING: Most sections of this document are outdated, dating back to the release 2011-01-14.

Contents

Section 1: RxDOS license	4
1.1 Contributions	4
1.1.1 Porting	4
1.1.2 lDOS, ldosboot, lmacros, instsect	4
1.1.3 iniblz BriefLZ depacker	4
Section 2: Introduction	6
Section 3: Bug reports	7
Section 4: Developers	8
Section 5: Contact	9
Section 6: Advise on usage of RxDOS version 7.20N	10
6.1 Developing RxDOS	10
Section 7: Introduction	11
7.1 Where are all the programs?	11
7.2 How do I install RxDOS?	11
7.3 What are the files RxDOS is made up of?	11
Section 8: Changes of RxDOS version 7.20N since 7.1.5	13
8.1 Initial changes, individual programs' fates	13
8.2 Changes to DOS	14
8.2.1 New interfaces	14
8.2.2 New kernel layout and building with linker obsolete	15
8.2.3 Later kernel changes	15
8.2.4 Misc	16
Section 9: Building RxDOS 7.2x from the source code	17
9.1 Required tools	17
9.2 Warnings to expect	17

9.3 Scripts	18
-------------	----

Section 1: RxDOS license

This product is distributed AS IS and contains no warranty whatsoever, including warranty of merchantability or fitness for a particular purpose.

Copyright (C) 1990-1999 Mike Podanoffsky

This is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License, see the file LICENSE.TXT.

1.1 Contributions

1.1.1 Porting

C. Masloch ported the provided source code to NASM, and packed these files up for the RxDOS 7.20N release. Her associated work is released as-is, without any warranty whatsoever, and is in the public domain. This does not affect other copyrights as noted above.

1.1.2 IDOS, Idosboot, Imacros, instsect

2008-2018 by C. Masloch

Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.

DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

1.1.3 iniblz BriefLZ depacker

BriefLZ - small fast Lempel-Ziv

8086 Assembly IDOS iniload payload depacker

Based on: BriefLZ C safe depacker

Copyright (c) 2002-2016 Joergen Ibsen

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented

as being the original software.

3. This notice may not be removed or altered from any source distribution.

Section 2: Introduction

This is the free RxDOS operating system. The main programs are a widely MS-DOS compatible DOS kernel (RxDOS.COM) and a custom command line interpreter/COMMAND.COM replacement (RxCMD). The RxDOS distribution includes some supplemental programs as well. These are a virtual file system driver for use in conventional memory (RxVDISK), boot loaders that load RxDOS.COM (ldosboot boot12, boot16, boot32), and a small program that installs ldosboot on a DOS drive (instsect).

Section 3: Bug reports

You may send bug reports directly to my e-mail address or register a account to write it to the BTTR Software support forum. (I'm a member of BTTR Software, and even if I can't help you there are some other DOS experts.) Contact data is in section 5.

Section 4: Developers

RxDOS was initially written and developed for many years by the professional programmer Mike Podanoffsky. He also wrote a book titled "Dissecting DOS" about RxDOS that was first published in 1994. However he stopped his development on RxDOS in 1999 or so, probably because he saw both MS-DOS and its best known competitor DR-DOS stopping their activities. Fortunately he released RxDOS version 7.1.5 on 1999-08-07 under the GNU General Public License.

I, C. Masloch, learned Assembly language (NASM) in 2007. I already used parts of FreeDOS for a long time when I thought about programming a DOS kernel. I didn't wanted to help with DOS-C (the FreeDOS kernel) because I can't write C so well and I really like Assembly more. In 2008 I heard from RxDOS and got it from the FreeDOS mirror servers. Later (2008-05 or so) I started porting the free RxDOS version 7.1.5 from the MASM source to NASM. This port eventually became RxDOS 7.20N.

Section 5: Contact

BTTR Software

<http://www.bttr-software.de/>

C. Masloch

cm -AT- bttr-software.de

Section 6: Advise on usage of RxDOS version 7.20N

Do not use RxDOS version 7.20N. It tries to re-produce every bug of RxDOS 7.1.5 (as found on the FreeDOS mirrors). It is not usable currently. The release of version 7.20N is only intended for developers.

6.1 Developing RxDOS

1. Do not take anything for granted.
2. Read and understand all the source code that you use from this release before claiming it works.
3. You can read the book 'Dissecting DOS' for some advise on RxDOS's internal structure, although it describes an older version. You will need a deeper understanding for development though.
4. As described in section 9 the build process no longer uses tools that run and build on DOS, are considered non-free, or require non-free programs to build. The kernel and RxCMD have been modified to use nothing but NASM.

Section 7: Introduction

7.1 Where are all the programs?

RxDOS does only consist of the DOS system core (kernel), the command line interpreter and a few supplemental programs.

It's recommended to use the programs from the FreeDOS project: Most of these run on RxDOS as well and they're all free like RxDOS as well.

7.2 How do I install RxDOS?

If you're running some DOS (possible RxDOS from an installation disk) locate the executable `instsect.com`. Type `'instsect A:'` (without the quotes), substituting `'A:'` by the drive you want to install RxDOS on. In general you'll have to type `'instsect A:'` or `'instsect B:'` to install RxDOS on a floppy drive or `'instsect C:'` to install RxDOS on a hard drive. (Note that RxDOS can boot from any partition, including non-primary partitions and subsequent hard drives.) Then, copy the files `RxDOS.COM` and `RxCMD.EXE` to the drive you want to install RxDOS on. If you want, you can copy `RxDOSPAK.COM` to be used as your `RxDOS.COM`, which is a file packed with `BriefLZ` and given a boot-time unpacker.

7.3 What are the files RxDOS is made up of?

`RxDOS.COM`

The RxDOS (Disk Operating System) kernel core. Required to boot RxDOS.

`RxDOSPAK.COM`

The RxDOS kernel core, compressed with `BriefLZ`. Can be substituted as `RxDOS.COM` to boot RxDOS.

`RxCMD.EXE`

The `RxCMD` (Command line interpreter) program. This is the program that RxDOS will start by default after it booted. You can however use `RXCMD.EXE` like any other program. You can also instruct RxDOS to start another program after it booted.

`instsect.com`

This program installs `ldosboot` to a DOS drive. Copies of `boot12.bin`, `boot16.bin`, and `boot32.bin` are contained inside `instsect.com`, you don't need these files separately to use `instsect.com`.

`RXVDISK.SYS`

This system driver creates a virtual drive inside the low memory area. It is used inside

RXCONFIG.SYS with a line like 'DEVICE=RXVDISK.SYS'. WARNING: This device driver is outdated and should not be used.

Section 8: Changes of RxDOS version 7.20N since 7.1.5

8.1 Initial changes, individual programs' fates

1. Only Netwide Assembler (NASM) 2.13.02 used for assembling.
2. Changed names of these programs:

RxDOSCMD to RxCMD

3. Dropped some programs:

RxDOSBIO (renamed RxBIO, later integrated into RxDOS.COM)
RxD_BOOT (dropped, use ldosboot and instsect instead)
RxDOSMBR (dropped, use ldosmbr instead)

4. Format of the source files modified:

1. The source files now use tab characters to make editing them easier. The tab characters should be expanded to up to 8 spaces for displaying.
2. Removed the funny boxes drawn with ; ' - . characters.

5. Replaced COPYING and gpl.txt by LICENSE.TXT which is an up-to-date plain-text GNU GPL Version 2 from <http://www.gnu.org/>

6. Directory structure changed.

Former:

\	Binaries, documentation, license, makefile, link files
\sources	Actual source, include/macro files, license
\objs	Object files
\errors	Assembler and linker error messages

New:

\	Make scripts
\RxBIO	RxBIO source
\RxDOS	RxDOS source
\RxCMD	RxCMD source
\RxAPP	Sources of RxMBR, RxBOOT, RxD_BOOT, RxVDISK
\MACRO	Macro files
\doc	Documentation, license
\lst	Listings

\map	Map files
\obj	Object files
\bin	Binaries

7. Some new or corrected code for RxDOS 7.30 added to source files. This code is deactivated using NASM conditional assembly and it is getting step by step through the next releases after RxDOS 7.20N.

8.2 Changes to DOS

8.2.1 New interfaces

1. Int21.335E Get extended DOS version

```
OUT: ax != 33FFh and ax != 3300h and ax != 0001h
      ax = 7852h "Rx" for RxDOS
      cx = build year
      dh = build month
      dl = build day
      bx = DOS kernel version
      di = BIO kernel version
          (since RxDOS 7.23 always same as bx)
      si = reserved, currently unset
```

2. Int21.335F Get boot drive and filename

```
INP: dx = 0
OUT: dx != 0 if supported
      dx:ax -> ASCIIZ filename with drive
          (file in root directory), in DOSDATA
      cx, bx, di, si = reserved, currently unset
```

3. Int21.3360 Get DOS capabilities

```
INP: CY
OUT: NC, ax != 33FFh and ax != 3300h and ax != 0001h
      ax = DOS capability flags
      cx = reserved, set to zero
      dh = size of a DPB
      dl = reserved, set to zero
      bh = size of an SFT
      bl = size of a CDS entry
      di, si = reserved, currently unset
```

4. Int21.33FF Get DOS version string (FreeDOS-compatible)

```
INP: dx = 0
OUT: dx != 0 if supported
      (may or may NOT clear the Carry Flag)
      dx:ax -> ASCIIZ version string, in DOSCODE
Example strings:
"RxDOS version 7.24 [hg 2d2992dcc533+] [2018-04-06]"
"FreeDOS kernel - SVN (build 2042 OEM:0xfd) [compiled Sep 22 2017]"
```

8.2.2 New kernel layout and building with linker obsolete

1. Since version 7.21 (2017-09-01), both the RxBIO and RxDOS kernel parts can be built as flat binaries linked internally by NASM's "bin" output format. This means that no external linker is required any longer.
2. Since version 7.22 (2018-03-09), the RxCMD shell can be built as flat binary linked internally by NASM's "bin" output format.
3. As early as 1b22935d1222 2018-03-10, RxDOS.SYS is included by RxBIO.SYS's build process. Thus, a single-file kernel (similar to FreeDOS's) is first achieved this way.
4. Since version 7.23 (2018-04-03) (as early as 7484396956e4 2018-03-30), the combined-kernel RxBIO.ASM is included into IDOS init. As early as bce5d6c1fbcd 2018-03-31, RxDOS.ASM is separately included into IDOS init, and RxDOS.SYS is no longer included by RxBIO.ASM.
5. As early as 2a093d179ffd 2018-04-06, RxDOS.ASM is split into doscode.asm and dosdata.asm, and both are used respectively as DOSCODE and DOSDATA.
6. As early as 3cb408e4a258 2018-04-12, RxBIO.ASM was split into biocode.asm and bioentry.asm, and the former is put into the DOSCODE section. Also, all variables except dosentry_to_dosdata_segment are moved out of DOENTRY.
7. As of b5b67355b503 2018-04-24, the interrupt 19h handler is left in DOENTRY. This supports use cases where the HMA or UMA may not be available to the handler any longer.

8.2.3 Later kernel changes

1. As of 8b75542e6ba1 2018-04-26, DOSCODE can be relocated to the HMA.
2. As of af2c5e43be7e 2018-04-28, DEVICE and DEVICEHIGH directives in [RX]CONFIG.SYS are parsed in the same pass. Also, DEVICEHIGH correctly tries loading the device driver into the UMA if its executable image fits. This requires a two-step approach, and the allocation check in 21.4B03 "load overlay" that was added in 58b36366e8e5 2018-05-03.
3. As of e02d8abd452c 2018-05-03, IDOS memory.asm was implemented to replace RxDOSMEM (now all of the latter calls into IDOS).
4. As of 282e01e8ec3e 2018-05-05, UMB linking during device driver processing (in init2_relocate_device) is implemented, and 21.5802/.5803 (get/set UMB link state) is fully implemented.
5. As of b758fe23b89b 2018-05-07, UMB linking and DOSDATA UMA relocation and DOSCODE UMA or HMA relocation are disabled by default; use UMB=ON and DOSDATA=UMA and DOSCODE=UMA,HMA in [RX]CONFIG.SYS to enable these. (The compatible option, DOS=UMB, means UMB=ON, while DOS=HMA or DOS=HIGH means DOSCODE=HMA. DOSDATA=UMA and DOSCODE=UMA do not have any equivalent compatible options.)
6. As of 93ca7200fbc0 2018-05-12, UMA and HMA relocation can be turned on with UMA=ALL and HMA=ALL in [RX]CONFIG.SYS.
7. As of bc8dee4269a0 2018-05-15, DR-DOS-style RPLOADER broadcasts are made.

8. As of a9f4fa3bad20 2018-06-06, [RX]CONFIG.SYS is read using the loaded DOS like a normal user, not with disk accesses directly.

8.2.4 Misc

1. RxDOS and RxCMD now allow embedding the Source Control Revision ID to identify the source used to build them. mak.sh needs to be called with use_build_revision_id=1 to enable this. It currently hardcodes the command to get the ID as "hg id -i". (b429250a515e 2018-03-15)
2. RxCMD's VER command now includes support for the /R option, which in addition to the RxCMD version, Source Control Revision ID, build date, and copyright lines, also displays the reported DOS version and OEM code (Int21.3000), reported true DOS version (Int21.3306), and version string (Int21.33FF). (1be3f3e781ca 2018-03-14)
3. RxDOSINI's AllocateSystemSpace is now implemented atop Int21.48, and creates S MCBs (owner = 8, name = "S", type stored later in name field). This enables MCB walkers to locate all allocations. (ac38e4a7eba1 2018-03-17)
4. ldosboot and lmacros (b06c90a1009c 2018-03-28) are included as subrepos. ldos is included as subrepo. (8fa5fccbcf40 2018-03-30)
5. instsect and iniblz are included as subrepos. (cd662caf7879 2018-10-08)

Section 9: Building RxDOS 7.2x from the source code

9.1 Required tools

1. NASM 2.04 .. 2.13, other versions might work (2.13.03 tested most recently).
2. One of these linkers:
 1. WarpLink 2.70
 2. Other 16-bit OMF linkers, with trivial adjustments
 3. NASM's internal linker (output format `bin`)

Usage of NASM's internal linker made `exe2bin` unnecessary. The kernel can now only be build with NASM, object file output support is discontinued.

9.2 Warnings to expect

The "word data exceeds bounds" warnings that are displayed for `biocode.asm:455`, `biocode.asm:504`, and `doscode.asm:600` can probably be ignored.

`ldosboot` shows messages indicating how many bytes are left available in the boot sector loaders, these are:

1. Main boot sector for all FAT types.
2. `FSINFO` for the `FSIBOOT`-using `FAT32`.

`ldosboot iniload` shows messages indicating how many bytes are left available before the various fixed labels, these are:

1. `ms7_entry`: MS-DOS 7/8 load protocol endpoint, at 200h.
2. `ldos_entry`: IDOS load protocol endpoint, at 400h.
3. `end`: End of common early loader (for IDOS and MS-DOS 6 endpoints).
4. `end2`: End of extended early loader (for MS-DOS 7/8 endpoint).

`iniblz` shows a message indicating how many bytes are used by the depacker. Additionally, building `RxDOSPAK.COM` also shows the same messages for `ldosboot iniload` as the normal build, these will generally be the same.

`IDOS entry.asm` and `RxDOS dosdata.asm` show how many bytes are left in front of specific fixed labels, these are:

1. The Interrupt Restoration Table (IRT), at 70h:100h (DOSEENTRY:100h).
2. The pre-SDA (pre Swappable Data Area), in DOSDATA.

9.3 Scripts

The mak.sh script should be sufficient to create a makefile or simple script for your particular platform. It was created for a Linux bash environment.

The provided documentation is partly written for Halibut, a program that compiles its input to various other formats. To compile the documentation into one text file and one HTML file, use the script doc/makdoc. This script is now called by the mak.sh script too.