RADLab(tm) Version 1.05
By: Ron A. Dilley and David Andersen
RADSoft. 1992
August, 1992
Revised: 08-29-1992

!NOTICE!NOTICE!NOTICE!NOTICE!NOTICE!NOTICE!

RADLab(tm) is
not a freeware product. David and I have spent many months on this program, and we intend to keep improving and adding to it to keep it on the top. RADLab(tm) is a shareware (try-before-you-buy) program. Please register this product if you like and or use it! A registration form is included in this archive. If you would like further information on registration, and the benefits of registering this version of RADLab(tm), don't hesitate to call one of out systems for more information. Registration is only $10 dollars, a small price to pay for the quality of this package, and the time that we have spent making it. The better the registration response the more inclined we will be to make even better versions of this program in the future.

!NOTICE!NOTICE!NOTICE!NOTICE!NOTICE!NOTICE!

Disclaimer:

I hate doing these, but it would seem that the nature of this program forces me to. This package deletes, renames, edits, and in general messes with files. So we are not responsible for any problems with this program, problems caused by this program, or damage caused by this program. We have taken a lot of time making this program stable, but we offer no warranties expressed or implied on this package. Use it at your own risk. If you have a problem, give us a call and tell us what happened. We want you to use and enjoy this package, so we want to fix any problems or incompatibilities that are found.

Ron A. Dilley
David Andersen

Features:

RADLab(tm) will test the following archive and file formats:
ZIP(Version 1.0 & 2.0 when it comes out), ARJ, LHA, LZH, ARC/Crunch, ARC/Squeeze,
ARC/Crush, PAK/Distill, ZOO (Version 2.0), Self Extracting ZIP, ARJ, and LHA,
Teledisk and GIF files.

RADLab(tm) will compress any non-compressed GIF File using GIFLite(tm).

RADLab(tm) will re-compress any compressed archive using a user definable compressor.

RADLab(tm) will then re-comment the new archive using a user definable re-commentor.

RADLab(tm) will scan all files in an archive for virii using McAfee's SCAN.

RADLab(tm) will run a second-pass scanner that is user definable on all files in an archive.
This user definable may be a Trojan scanner.

RADLab(tm) may be used from the command line to scan whole directories at a time using wild cards.

RADLab(tm) will write a status line to the description file when using PC Board or ProDoor BBS software.

RADLab(tm) will add the contents of a FILE-ID.DIZ or DESC.SDI file found in an archive to the description file when using PC Board or ProDoor BBS software.

RADLab(tm) tests file dates of all files in the archive to determine the oldest file, then compares that file date with a user definable file date limit.

RADLab(tm) will move any failed files to a holding directory that is user definable.

RADLab(tm) supports PC Boards "TEST" option.

RADLab(tm) supports fossil controlled serial communication, allowing remote users to see what RADLab(tm) is doing.

RADLab(tm) writes the status of every run to a user definable log file.

RADLab(tm) is only 40k, and requires approximately 90k of free ram, not including the archive shells such as PKUnZip(tm) which requires about 80k to run. Total free ram necessary is about 170k.

Features in the planning stage:

RADLab(tm) will soon allow removal of user definable advertisement files from archives before they
are re-compressed.

RADLab(tm) will soon directly support WildCat.

RADLab(tm) will soon directly support Telegard.

RADLab(tm) will soon directly support WWIV.

RADLab(tm) will soon support Non-Fossil serial port control.

RADLab(tm) will soon support .DSM file formats for testing of Amiga files.

RADLab(tm) will soon support .LBR file formats.

RADLab(tm) will soon support multi-node operation, via file sharing, and multiple config files.

Why we are making this product:

I have been running a bulletin board system for about two years, and I realized as I am sure you have that file integrity is very important to the reputation of a bulletin board system. People don't call systems that are know for having bad files on them. And we are all paranoid about the change of passing or contracting a virus. So I decided some months ago to install an automated file checker on my PC Board system. I looked around and found ZIPLAB(tm). I installed it, and it ran twice, worked great for those two runs but it was pretty useless after that. After spending two weeks trying to get the program to work more than twice I gave up in frustration. I called David, and talked him into conspiring with me to build a better archive tester than the one I had attempted to use. If I was having problems getting ZIPLAB(tm) to work then others must be just as frustrated or even more, so there I definitely as market for this program. We spent some time looking at what ZIPLAB(tm)'s faults were. We had no intention of making the same mistakes that we thought Jeff Morley had made. ZIPLAB(tm), a great program for what it was, and what it was written in was too big, required to much ram and did not test archives to the degree that we paranoid folk would be comfortable with. Plus I found ZIPLAB(tm) hard to install, and run (I could not get it to run). So this is what we have made to give ZIPLAB(tm) and any other archive tester as run for it's money. I hope you are all satisfied with this product, and if you are not then please get in tough with us! We will be glad to hear from you and listen to your ideas.

Files included in this archive:

RADLAB.EXE	RADLab(tm) Executable
RADLAB.CFG	RADLab(tm) Config file
RADLAB.DAT	RADLab(tm) Data file
RADLAB.DOC	RADLab(tm) Documentation file MS Word Format
RADLAB.WRI	RADLab(tm) Documentation file Word Perfect 5.1 Format
RADLAB.TXT	RADLab(tm) Documentation file TEXT Format
INSTALL.EXE	RADLab(tm) Install Executable
INSTALL.DAT	RADLab(tm) Install Data file
INSTALL.DOC	RADLab(tm) Install Documentation file MS Word Format
INSTALL.WRI	RADLab(tm) Install Documentation file Word Perfect Format
INSTALL.TXT	RADLab(tm) Install Documentation file TEXT Format
REGISTER.TXT	RADLab(tm) Registration Form TEXT Format
PROUT1.BAT	ProDoor Example batch file
PROUTEST.BAT	ProDoor Example batch file
BBSAD.ASC	RADLab(tm) Example Comment Advertisement
FILE_ID.DIZ	RADLab(tm) Description File
DESC.SDI	RADLab(tm) Description File
ARCHIVE.EXE	RADLab(tm) Self Extracting Archive Including all necessary utilities made by other programmers for use with RADLab(tm). These programs are NOT part of the registration for RADLab(tm). I have included them for simplicity, nothing more, they are all very good programs, and you should register them with the companies and programmers that have made them.

I have included a tweak of PKUnZip 1.93 for one reason alone. There are people out there using this hack. S, I have included it for unzipping only. PKZip is version 1.1 (There is a method here to my madness). If someone uploads a file using the hack (of the beta version of PKZip, RADLab(tm) will de-compress it, and do all the testing, the RADLab(tm) will re-compress the file using PKZip 1.1. This will weed out the files uploaded to your system with the non-release version of PKZip.

PKUnZip, PKZip, PKXArc, PKUnPak, ARJ, LHarc, Zoo, TDCheck, GIFTest, GIFLight, McAffe's Scan, RECommenter, and X00 are all shareware, and require a registration fee separate from RADLab(tm). They are high caliber and the programmers deserve the recognition, and the funds.

A note about RADLAB.ZIP including extra utilities not written by myself. Version 1.05 is the last version of RADLab(tm) that will include the utilities that you will need to run RADLab(tm). After this version, RADLab(tm) will be archived alone, and I will provide all utilities that RADLab(tm) requires on my bulletin board system. In a conference specifically for RADLab(tm) service and support. All new updates to these utilities will be available on Boolean's Rebellion for download. This is to insure that all nescisary information that is included with the utilities that RADLab(tm) uses get to the end user. There are simply too many programs to include in a single archive. I am sorry that there is no other way to do this, and I appologize for forcing you to go hunting for these programs.

System requirements:

RADLab(tm) is approximately 40k in this version, and requires about 90k to run. I will in the future shrink this size down but for now 90k is small enough. PKUnZip requires about 80k to run, so in most situations, the most free ram you will need is 170k. A far cry from the 240k that ZIPLAB(tm) requires, and that does not include the ram that the archive utilities requite. So with ZIPLAB(tm) you are going to need about 320k free just to run the program. A ram-drive will make a large difference in the speed of file processing. I would recommend a 3MB ram-drive or larger. I use a 4MB ram-drive just to be on the safe side.

Running RADLab(tm):

RADLab(tm) Command Modes:

Mode	Syntax to run in mode	Features
LOCAL	RADLAB <fname>	Tests all files, checks file dates, moves files.
TEST	RADLAB <fname> TEST	Tests all files, does not check file dates, moves files.
PCB	RADLAB <fname> PCB <descfile>	Tests all files, checks file dates, writes PCBPASS.TXT or PCBFAIL.TXT, edits desc file, adds status line to desc file, moves files.
PCB/TEST	RADLAB <fname> PCB TEST	Tests all files, does not check file dates, writes PCBPASS.TXT or PCBFAIL.TXT, moves files.
PRO	RADLAB <fname> PRO <descfile>	Tests all files, checks file dates, writes PCBPASS.TXT or PCBFAIL.TXT, edits desc file, adds status line to desc file, moves files.

RADLab(tm) Configuration file:

RADLab(tm) uses a text based configuration file called RADLAB.CFG. This file MUST be in a directory that is listed in the path, or RADLab(tm) will not be able to find the configuration file, and all kinds of strange things will happen. The file format is as follows.

```
D:\FILES\
D:\FILES\FAILED.LST
D:\RADLAB\LOG\
01-01-91
E:\
YES
PKZIP -$ -R -P
.ZIP
RECOM -O -S
YES
F-PROT /COMMAND /SILENT /NOMEM /TROJAN
YES
2
2
38400
0
8
1
```

RADLab(tm) Explanation of configuration file:

This is a full explanation of the above example configuration file, line by line. There should be no other lines before the above lines, you may have as many lines of comments after these 17 lines.

D:\FILES\ * Full path to the directory that you want RADLab(tm) to
move failed

files to. BE SURE TO INCLUDE A "\" AT THE END OF THE

PATH!
D:\FILES\FAILED.LST * Full path including file name that you wish RADLab(tm)
to use when

writing the failed file log.
D:\RADLAB\LOG\ * Full path to the directory that you want RADLab(tm) to
write it's log

and error files to.
01-01-91 * Archives containing files older than this date will be
failed as "Too

Old". The date format MUST be MM-DD-YY.
E:\ * Full path to your working directory. A ram-drive for
example. (If you

use a ram-drive, I recommend at least 3MB).
YES * Software switch (Re-Compress archives YES/NO).
PKZIP -$ -R -P * Command string for re-compressing archives.
.ZIP * File extension (including ".") of the archive type you
are going to use

with the re-compress switch. Used only when running in

PCB or PRO

mode.
RECOM -O -S * Command string for re-commenting archives after they
have been

re-compressed.
YES * Software switch (Secondary File Scan YES/NO). This
can be used to

run a second form of scan on the archives and programs

in those archives.

A Trojan scanner for example.
F-PROT /SILENT * Command string for secondary file scanner.
YES * Software switch (FOSSIL support YES/NO).
2 * Serial Port (1 = COM1, 2 = COM2, 3 = COM3, 4 =
COM4).
2 * Flow Control (0 = No Flow Control, 1 = Software
(XON/XOFF) Flow Control, 2 = Hardware (CTS/DTE)
Flow Control, 3 = Both

Software and Hardware Flow Control).
38400 * Port Speed (300, 1200, 2400, 4800, 9600, 19200,
38400).
0 * Parity (0 = No Parity, 1 = ODD Parity, 2 = EVEN
Parity).
8 * Word Length (5 - 8).
1 * Stop Bits (1, 1.5 (Only when using 5 bit words), or 2).

Setting up RADLab(tm) for PC Board:

If you are running PC Board you will need to replace your existing PCBTEST.BAT file with the following example. This simple batch file will work both the standard upload test as well as the "TEST" option that is built into PC Board.

```
@echo off
RADLAB %1 PCB %3 %2
```

Where %1 is the full path to the file to be tested, PCB is the bbs type you are using (PC Board), %3 is the full path to your UPDESC.1 file, and %2 is the status argument passed from PC Board (UPLOAD/DOWNLOAD/TEST). If the argument is TEST then RADLab(tm) will not fail files because of date limits, and there will be no editing of the UPDESC.1 file.

Setting up RADLab(tm) for ProDoor:

If you are running ProDoor, you can use the three files included in this archive to run RADLab(tm). These files have been provided by Bill Thomson, SysOp of a ProDoor system that is using RADLab(tm). For clarification of the use of these batch file, please get in touch with Bill on his system, or I will route those questions to him.

```
PROUT1.BAT
PROUTEST.BAT
DIDTEST.COM          By: Dave Navarro
```

Setting up RADLab(tm) for LSD:

RADLab is running successfully on several LSD based systems as of this release. The batch file that is being used looks something line this.

```
cd \radlab
xu port:n:off        Where n is the port number.
radlab %1
xu port:n:on         Where n is the port number.
cd \lsd
```

Setting up RADLab(tm) for any Bulletin Board System:

RADLab(tm) was written to be used from the command line so to use RADLab(tm) with any BBS package you simply need to find out what your BBS program uses to call an archive tester (a batch file) or you can use RADLab with wild cards and test the whole directory that new uploads are sent to by the external protocols you use, and then use RADLab(tm) in the batch file that calls your external protocols, and have it run after the upload. Then the error level that is returned is the error level that RADLab(tm) returns. 0 for pass, and 1 for fail. I would like to include as many setups for different BBS packages in the documentation. If you have a BBS program, and you get RADLab(tm) setup, please take the time to get back to us with your setup so I can include it in the documentation, so other users of the same BBS program can more easily set it up.

ATTENTION!

All the files that are included with RADLab(tm) MUST be in a directory that is listed in the PATH statement in your AUTOEXEC.BAT file. Failure to do so may result in some every interesting runs.

How to tell if RADLab(tm) has passed or failed a file:

RADLab(tm) returns error levels to signal programs whether or not the file has passed. 0 if there were no problems with the file, RADLab(tm) only returns error level 1 if the problem with the file is known (i.e. the file has errors, the file fails CRC testing, the file is too old.) any other problems with the software returns error level 0 to make sure that no files are lost due to software problems. All errors encountered during program run are written to a file called RADLAB.ERR.

RADLab(tm) will also create a file called PCBPASS.TXT or PCBFAIL.TXT depending on whether or not the file failed. These files are used by PC Board and ProDoor to determine file status. These files are created in the directory that RADLab was executed from, and if you are running PC Board or ProDoor, will be deleted after the upload is finished. All failed files are copied into a holding directory specified by the RADLAB.CFG file.

FOSSIL Support:

I have added FOSSIL support to RADLab(tm) starting with Version 1.03. If you wish to use the FOSSIL support to send information to the remote user you will need to select YES in the RADLab(tm) configuration file for FOSSIL support. If you already have a fossil driver loaded when RADLab(tm) is executed, RADLab(tm) will detect it and start it's run. If you don't have a FOSSIL installed when RADLab(tm) starts it's run, it will detect the lack of a loaded FOSSIL, and load X00(tm) and then run and at the end of RADLab(tm)'s run it will unload X00(tm). I will in time as support for non-FOSSIL serial communication, but to get it to run faster that 9600 baud will require some time on my part, the FOSSIL for now is the quick fix to the problem of users not knowing what RADLab(tm) is doing.

Making RADLab(tm) Better:

We have written RADLab(tm) in such a way that we can with next to no increase in file size, or decrease in run speed, add further support for different archive formats. I believe that if there is demand for support of an archive type that RADLab(tm) doesn't presently recognize, then I will add it to RADLab(tm). If there is an archive or file type that you wish to see added to RADLab(tm) you need to do three things. First, find a decompressor/tester for the file, and an example of the file (So I can look at the file header to find something to use as a reference). And second, call Boolean's Rebellion and upload those files to us. And lastly, leave us a message telling us why it should be added to RADLab(tm).

If you think that RADLab(tm) is lacking in some respect, or you think that there is something that you could use in an archive tester that RADLab(tm) does not do, then take the time to call us and tell us what you want or need. You are the people that we are writing this program for. It is you that will make or break this program. The more effort you expend to make this product better, the better it will become.

In Closing:

David and I have spent many months working on this program. We found that one of the big failings of ZIPLAB(tm) is that is it very unstable and works here and there. We don't want to hear people saying that about RADLab(tm), I should say, we want to hear about it if there is something in RADLab(tm) that is not working so we can fix it. So if you come across a glitch, bug, incompatibility, or problem then please make an effort to get back to use to report it. All the ideas and help that we get is appreciated and if we add something to the program that you though of we will include your name in the list of contributing users.

All BETA-Test systems will get all the new updates of RADLab(tm) free, with no registration required. We reserve the right to choose, terminate, suspend and of all BETA-Test sites.

The Plug:

The Bauhaus
619.670.3939 DS/14.4k
300 Megabytes On-Line (HD's)
Supporting: FIDONet
SysOp: David Andersen /aka Dimensional Beast

Boolean's Rebellion
619.465.1305 DS/16.8k
619.465.1357 HST/14.4k
619.465.2417 HST/14.4k
619.465.5118 HST/14.4k
1.6 Gigabytes On-Line (HD's)
1.1 Gigabytes On-Line (CD-Roms)
Home of RADSoft. 1992
Supporting: OOFNet (99:700/102)
SysOp: Ron A. Dilley /aka Motivator

BETA-Test Sites:

The File Cabinet
619.426.0740 HST/14.4k *PRIVATE*
619.426.7760 9600/ Compucom
Running: ProLogon/ProDoor
SysOp: Bill Thomson

The CLICK BBS
619.633.1288 DS/14.4k (3 Lines Auto Ring-Down)
2.5 Gigabytes On-Line (HD's)
CD-Rom Support
Supporting: OOFNet/*PremiereNET*
Running: Latest Version of LSD
SysOp: Hackster

The Kludge
619.596.7668 DS/14.4k
1.8 Gigabytes On-Line (HD's)
Supporting: OOFNet
Running: LSD
SysOp: Dick Goesinia

Problem Reports:

This is a new section in RADLab(tm)'s documentation to help you if you are having problems getting RADLab(tm) to work with your system. This list is compiled from all bug reports and their situations (to the best of my knowledge).

PROBLEM: User reports a problem when the program runs, the program throws garbage all over the
screen, but the program works just file other than that.
SOLUTION: The problem is that the user did not have the RADLab(tm) directory listed in the path entry found in the AUTOEXEC.BAT file. Make sure that all files dealing with RADLab(tm) are listed in the path statement for RADLab(tm) to work properly.

PROBLEM: User reports that when RADLab(tm) runs on a Hercules compatible monochrome system that RADLab(tm) does not switch video pages when it spawns to any of the external utilities. The User has tried this on two of his machines with the same problem. The program runs just file, it simply messes up the screen.
SOLUTION: To date I still don't know what is going on with this users systems. RADLab(tm) has been tested on several Hercules compatible systems with no errors. The page switching routine uses a BIOS level call to switch pages and there are no known incompatibilities. So I have no solution for this, I would look for TSR's that are running on the machine that might modify the page switching interrupt, or perhaps something that configures the bit map address. I have heard of that interrupt causing problems.

PROBLEM: Two users report that no FOSSIL output is occurring on their systems though the program is running just fine.
SOLUTION: I look kinda funny on this one. I found out what the problem was. I wrote the program to default to local mode if only one command line argument was passed to RADLab(tm). This was a great idea from my standpoint, my system only uses the PC Board enhanced features, so I always pass more than one argument. BUT, not all BBS programs are gonna use that and they would run RADLab(tm) using only one argument, and no matter what they did it would default to local mode, and shut the FOSSIL down. I re-wrote that portion of the code. Problem fixed in all versions of RADLab(tm) compiled after 08-16-1992.

PROBLEM: User reports that his system is running just fine, but the only thing that he gets out of the serial port is garbage. Program runs just fine.
SOLUTION: User was calling his RADLAB.CFG file RADLAB1.CFG. So RADLab(tm) could not find the configuration file, and was configuring the FOSSIL port for 5 bit words. You must be sure that the RADLAB.CFG file exists, and is in the path.

PROBLEM: User is getting nothing out of the serial port. Program works just fine.
SOLUTION: User had flow control and port lines switched in the RADLAB.CFG file. Much to my embarrassment, I found that in the documentation for 1.04 Revised on 8-16-1992, there were several syntax errors dealing with the configuration file that did not match the true setup, or the installer. So I re-wrote the documentation file. Sorry to all of you who cam across that one.

PROBLEM: User reports that RADLab(tm) is not saving the sub-directory information.
SOLUTION: Well, once again I have found an error on my part. The command to get PKZip to compress
a file with sub-directories, and recursion is PKZip -$ -R -P, NOT PKZip -$ -R which only

recurses the directories, and saves the files, but not the directories. I also had to make a small change to the internal syntax that RADLab(tm) uses to execute the re-compress routine. RADLab now moves into the temporary directory before it executes the compress. The reasong being, if you call PKZip with PKZIP -$ -R -P C:\TEMP\*.* and you are just interested in the files and directories inside the directory TEMP, PKZip includes TEMP in the stored path. So to cure this RADLab(tm) moves into the temporary directory and executes PKZIP -$ -R -P *.*
Problem fixed in version 1.05, compile date: 08-29-1992.

RADLab(tm) History:

Version 1.05:

Apon suggestions from several users, RADLab(tm) will now run a second pass file tester on all files. This test can be used to run a Trojan scanner on files after the virus scanner has run. This option is removable, and the tested is user selectable. I have added support for ZOO version 2.0 files. I have done some large scale re-writing of some of the modules to get that extra speed out of the code. I have also shrunk that code down. I fixed a problem with the re-commenter, please read this file to see the layout of the RADLAB.CFG file it has changed.

Version 1.04:

I finished the FOSSIL support, ready for the public after a long period of testing (about a week). I have been getting so much positive feedback about the program, not to mention demands for the FOSSIL support version to be made available that I put it out. I have added an installer to the archive to make installation of the program simpler. Yell if there are any problems.

Version 1.03:

First attempt at FOSSIL support, spent a long time getting my fossil library to work correctly. Re-wrote several of the routines to get more speed with less memory usage. I made many small tweaks to get the new FOSSIL support working 100%.

Version 1.02:

Fixed a problem with running re-compression & the re-commenter. Added PRODoor support for desc. file(s). Added a config switch to turn re-compression on and off. Added support for PC Boards internal "TEST" option. Added FILE_ID.DIZ and DESC.SDI support by request.

Version 1.01:

Fixed a problem with the stack. Version 1.0 would not test wild card files without running out of stack space. Added a temp working directory for use with a RAM-Drive to increase speed of decompression, scanning and re-compression.

Version 1.00:

A quick release to fix a problem in the software that I had not found while running it on this system. Unfortunately, rushing this release ended up putting the program out with more bugs that I had not found in testing.

BETA Versions .00 - .14:

There were MANY betas of this program as David and I fooled around with the code to make it fast and take up as little memory as possible. I didn't think it is important to mention any of the special events during the beta phase of this program. Lets just say that the program has been much improved in the past months.