

How can I run a REXX program without typing "REXX"?

About the problem

When running REXX programs in environments other than OS/2, PC DOS 7, or PC DOS 2000, and in the absence of special preparation, it is necessary to include `REXX` in the command. For example, if you have a REXX program named `MYPROG`, which accepts a filename as a parameter, you could run it under OS/2 or PC DOS 2000 by entering:

```
MYPROG D:\MYPATH\MYFILE.DAT
```

When using Object REXX under Windows 95, 98, or NT, or DOS 7 REXX under a version of PC DOS other than 7 or 2000, or under Windows 95, 98, or NT, you would have to enter:

```
REXX MYPROG.REX D:\MYPATH\MYFILE.DAT
```

This is awkward, because you have to mentally track which commands are REXX programs, and which are not. This would be as unnatural as having to keep track of which commands are `.BAT`, `.COM`, `.EXE`, or internal.

Solutions

There are six approaches to running REXX programs without having to type `REXX` on Windows 95/98, Windows NT, and DOS other than IBM PC-DOS 7.0:

- A. Front-end `.BAT`ch files
- B. DOSKEY macros
- C. Bilingual `.BAT`/`.CMD` files
- D. Filetype Association
- E. Slash-star as the first two characters on the first line of a standard procedures command file (`.BAT` and/or `.CMD`, depending on the operating system in use)
- F. Alternative command shell

Not all of these approaches will work with all REXX language processors on all of these platforms. The following tables summarize the availability of these solutions in the various computing environments, and the pros and cons of each approach. A detailed discussion follows the tables. The discussion here centers around Object REXX under the Windows 95/98/NT operating systems, native REXX and Object REXX under OS/2, native REXX under PC-DOS 7. Quercus Personal REXX and the use of PC-DOS 7 REXX under Windows 95/98/NT is touched upon (note: PC-DOS 7 REXX is also part of PC-DOS 2000).

Applicability of Various Solutions to each Environment

Environment	Front-end .BAT	DOSKEY macros	Bilingual files	FType Assoc	"/*" first	Alt{2} Shell
DOS with Personal REXX	+Yes	+Yes	+Yes	-No {1}	-No	+Yes

DOS with PC-DOS 7/2000 REXX	+Yes	+Yes	+Yes	-No{1}	+Yes	+Yes
Windows 95/98 with Object REXX	+Yes	+Yes	+Yes	-No	-No	+Yes
Windows NT with Object REXX	+Yes	+Yes	+Yes	+Yes	-No	+Yes
Windows NT with OS/2 v1.3 REXX{4}	+Yes	-No	+Yes	-No	+Yes	=N/A
OS/2 with OS/2 Classic REXX	+Yes	- No {3}	+Yes	-No	+Yes	+Yes {2}
OS/2 with Object REXX for OS/2	+Yes	-No {3}	+Yes	-No	+Yes	+Yes {2}
OS/2 with PC-DOS 7 REXX	+Yes	-No {3}	+Yes	-No	-No	+Yes

Legend: + is a good thing; - is a bad thing; = is neutral.

Notes:

1. Quercus Personal REXX provides this capability for all levels of DOS; IBM PC DOS 7 and PC DOS 2000 support REXX programs in .BAT files
2. Yes, but not required if file type is .CMD
3. DOSKEY works in a DOS Window (under OS/2 MDOS), but not at an OS/2 command prompt.
4. For details on how to do this, see the Microsoft knowledgebase article entitled "Running REXX Scripts on Windows NT", at <http://support.microsoft.com/support/kb/articles/q96/3/26.asp>

Pros and Cons of Various Solutions

Characteristic	Front-end .BAT	DOSKEY macros	Bilingual files	FType Assoc	"/*" first	Alt{2} Shell
.CMD Runs in both OS/2 and Windows NT	+Yes	+Yes	+Yes	-No	-No	+Yes
.BAT Runs in both DOS and Windows 95/98	+Yes	+Yes	+Yes	-No	-No	+Yes
Can be both command & subroutine	+Yes	+Yes	+Yes	+Yes {1}	+Yes	+Yes
Can be redirected/piped	+Yes	+Yes	+Yes	-No	+Yes	+Yes /font>
Produces spurious output line	+No	+No	-Yes	+No	+No	+No
Can be passed ; , =	-No	+Yes	-No	+Yes	+Yes	= {3}
Must be set up each session	+No	-Yes	+No	+No	+No	+No
Is space efficient	-No	+Yes	=Some	+Yes	+Yes	+Yes

Implied Create/Copy/Rename/Delete	-No	-No	+Yes	+Yes	+Yes	+Yes
Can permanently set environment variables & change directories in Windows 95/98/NT & DOS < 7 {1}	-No	-No	-No	-No	-No	+Yes
Honors PATH search order	+Yes	-No	+Yes	+Yes	+Yes	+Yes
Cost (not counting your time)	+\$0	+\$0	+\$0	+\$0	+\$0	-> \$0

Legend: + is a good thing; - is a bad thing; = is neutral.

Notes:

1. This works fine for Object REXX under Windows NT. However, calling subroutines or functions with an extension of .REX would require that the subroutine name include the extension explicitly if you were using either the PC-DOS 7/2000 or OS/2 v1.3 REXX interpreters under Windows NT.
2. In this case, those from J.P. Software (4DOS, 4NT, 4OS2)
3. The J.P. Software alternate shells treat these characters the same as their IBM or Microsoft counterparts.

A. Front-end .BATch files

In this approach, a .BAT file is created for each REXX program. When a command is issued with the name of the REXX program, the .BAT file is invoked by the operating system, and it, in turn, invokes the REXX program.

To simplify the creation of a batch program for each REXX program, you can simply copy the following template with the name of the REXX program and an extension of .BAT:

```
@ECHO OFF
SET REXXPROG_NAME=%0
SET REXXPROG_PARMS=
:GET_PARMS
SET REXXPROG_PARMS=%REXXPROG_PARMS% %1 %2 %3 %4 %5 %6 %7 %8 %9
FOR %%A IN ( 1 2 3 4 5 6 7 8 9 ) DO SHIFT
IF NOT '%1' == '.' GOTO GET_PARMS
REXX %REXXPROG_NAME%%REXX_EXT% %REXXPROG_PARMS%
SET REXXPROG_PARMS=
SET REXXPROG_NAME=
```

The extension used to name your REXX program will depend on what your REXX language processor expects. Be sure to SET REXX_EXT= in your AUTOEXEC.BAT, STARTUP.CMD, CONFIG.SYS, or Registry. Be sure to include the leading period in the SET command (i.e. SET REXX_EXT=.REX).

Advantages

- This approach is probably the most portable of all those detailed here.
- It is possible to automate the initial creation of .BAT files for each REXX program.

- There is no need to set up the front-end batch files each session.
- Can be both command & subroutine.
- Commands implemented in this fashion can be redirected/piped.
- This method does not produce any spurious output.
- This method works properly with the normal PATH search order.

Disadvantages

- This approach is the most wasteful of disk space. On large partitions using the FAT16 file system, the wasted space can be up to 32KB/file. (This is because, under these conditions, the smallest quantity of disk space which can be allocated is 32KB, even though the front-end .BAT file is only 300 bytes long.)
- Commands invoked this way cannot permanently set environment variables and change directories in Win95/98/NT and DOS {See note 1, above}.
- Commands invoked in this manner cannot be passed semicolon, comma, or equal sign characters in their parameters.
- When a REXX program is created, copied, renamed, or erased, its corresponding .BAT file must be created, copied, renamed or erased as well.
- When an Object REXX or PC-DOS 7 REXX program sets an environment variable in 95, Windows 98, Windows NT, or changes a directory, that alteration to the machine state does not persist past the termination of the program. This also applies to the execution of PC-DOS 7 REXX in non-IBM versions of DOS, and versions of IBM PC-DOS prior to 7. This does not apply to REXX in OS/2, or PC-DOS REXX in DOS 7 or later, or to Quercus Personal REXX under DOS.

B. DOSKEY Macros

This approach sets up a DOSKEY macro for each REXX program. The syntax to be used is:

```
DOSKEY rexfname=d:\path\rexx.ext rexfname $*
```

Where:

rexfname

is the name of the REXX program

d:\path

is the drive and directory where the REXX language processor resides. (This may be omitted if the REXX language processor resides in a directory listed in the PATH environment variable.)

rexx.ext

is the name and extension of the REXX language processor. If there is only one executable file for the REXX language processor (e.g. a .EXE, but no .COM), the extension may be omitted.

Advantages

- Can be passed semicolon, comma, equal sign.
- This approach is space efficient: it does not require any disk space.
- This technique allows REXX programs to be as both commands and subroutines.
- Commands invoked this way can be redirected/piped.
- This technique does not produce any spurious output.
- It is possible to automate the initial creation of DOSKEY macros for each REXX program.

Disadvantages

- Commands invoked in this manner cannot be passed semicolon, comma, or equal sign characters in their parameters.

- When an Object REXX or PC-DOS 7 REXX program sets an environment variable in 95, Windows 98, Windows NT, or changes a directory, that alteration to the machine state does not persist past the termination of the program. This also applies to the execution of PC-DOS 7 REXX in non-IBM versions of DOS, and versions of IBM PC-DOS prior to 7. This does not apply to REXX in OS/2, or PC-DOS REXX in DOS 7 or later, or to Quercus Personal REXX under DOS.
- The DOSKEY macros must be set up each session.
- When a REXX program is created, copied, renamed, or erased, its corresponding DOSKEY macro must be created, copied, renamed or erased as well.
- This method does **not** work properly with the normal PATH search order. If a drive and path are specified for the REXX program as part of the DOSKEY macro, any other file with the same name, even if it appears earlier in the PATH search order, will not be executed. If a path is specified, but no drive, then the given drive must exist for any current drive when the command is executed. If a drive is specified, but no path, the command file must be found in whatever the current directory is when the command is issued. If neither a drive nor directory is specified for the REXX program, then it must exist in the current directory on the current drive whenever the command is issued.

C. Bilingual .BAT/.CMD Files

A "bilingual" REXX program has a format which allows it to be processed as either a batch file or as a REXX routine:

```

/* 2>NUL:
@ECHO OFF
SET REXXPROG_NAME=%0
SET REXXPROG_PARMS=
:GET_PARMS
SET REXXPROG_PARMS=%REXXPROG_PARMS% %1 %2 %3 %4 %5 %6 %7 %8 %9
FOR %%A IN ( 1 2 3 4 5 6 7 8 9 ) DO SHIFT
IF NOT '%1' == '.' GOTO GET_PARMS
REXX %REXXPROG_NAME%%REXX_EXT% %REXXPROG_PARMS%
GOTO DONE

REXX program follows... */

SAY 'HELLO,' ARG() 'ARGS=>'ARG(1)'<' /* Replace this line with */
/* your REXX program */
EXIT

/* The following lines must come at the very end of the file, */
/* and must follow an unconditional EXIT or RETURN. */
:DONE
SET REXXPROG_PARMS=
SET REXXPROG_NAME=

```

Notes:

- The extension used to name your REXX program will depend on what your REXX language processor expects. Be sure to SET REXX_EXT= in your AUTOEXEC.BAT, STARTUP.CMD, CONFIG.SYS, or Registry.

- As you would expect, the redirection and piping symbols, "<", ">", and "|", are not passed through to the REXX program unless they are enclosed in quotes, just as with any other batch file. (In which case, the quotation marks are passed, as well).

Advantages

- This scheme will work under all versions of DOS, Windows, OS/2, CMS, and TSO (with the caveat of a spurious error message under some DOS versions and Windows 95/98), provided that the file has a name with the appropriate extension. (The first line must be different under AIX or Linux.)
- REXX programs implemented in this fashion can be used as both commands and subroutine.
- Commands invoked in this manner can be redirected/piped.
- When a program using this mechanism is created, copied, renamed, or deleted, there are no additional tasks to be performed.
- There is no required additional initial set up for each session.
- This method works properly with the normal PATH search order.

Neutral

- This technique is fairly space efficient: it adds about 300 bytes to each program file.

Disadvantages

- Any commas, semicolons, or equal signs which appear in the command line will not be passed to the REXX routine. This is just the way DOS/Windows handles parameters to Batch files. Commas and semicolons may be enclosed in quotation marks (") to avoid this problem, but equal signs cannot. Note also, that the quotation marks will be passed to the REXX program. Note also that under Windows NT, you may have to use the editing escape character (^) with ampersand (& or &&) and parentheses.
- Piping and redirection have no effect!
- Groups of sequential multiple spaces on the command line will be reduced to a single space (its DOS once again).
- The leading "/" will result in a diagnostic message under DOS and Windows 95/98, but the REXX routine will run just fine. (Under Windows NT, the message will be redirected and not display). If you don't like the spurious Error message, you could place a CLS command in the file before or after the @ECHO OFF. On Windows NT, the spurious error message can be redirected away, by appending "2<NUL:" to the first line. If you don't like the spurious REM echo, you could place a CLS command in the file at the start of the REXX code. Or you could use ANSI.SYS capabilities to move the cursor up a line or two and erase the unwanted text.
- When an Object REXX or PC-DOS 7 REXX program sets an environment variable in 95, Windows 98, Windows NT, or changes a directory, that alteration to the machine state does not persist past the termination of the program. This also applies to the execution of PC-DOS 7 REXX in non-IBM versions of DOS, and versions of IBM PC-DOS prior to 7. This does not apply to REXX in OS/2, or PC-DOS REXX in DOS 7 or later, or to Quercus Personal REXX under DOS.

(Given these restrictions, you may still want to type "REXX" ;-)

D. Filetype Association

The CMS, TSO, AIX, Linux, OS/2, and IBM PC-DOS operating systems all have built-in mechanisms for recognizing REXX programs without the need to type "REXX" at the beginning of the command line.

Quercus Personal REXX for DOS has a mechanism for recognizing implicit REXX commands which can be installed at the user's discretion.

Windows 95/98, and versions of DOS other than IBM PC-DOS 7.0 and PC-DOS 2000, do not have a mechanism for implicit REXX from a command prompt.

Windows NT has a method of specifying the processor for any extensions other than .EXE, .COM, .BAT, and .CMD. Setting this up is a three-step process: first you have to define a "file type" and the command to process it; second, you associate one or more extensions with this file type; finally, you inform Windows NT that it should search for the extension(s) you want when it processes a command.

Step 1. Specify how REXX scripts are executed:

```
FTYPE REXXscript=C:\ObjREXX\REXX.EXE "%1" %*
```

Step 2. Specify that files with the .REX extension contain REXX scripts:

```
ASSOC .REX=REXXscript
```

Note: associating .CMD, .BAT, .EXE, or .COM files with the REXXscript file type will not have any effect.

Step 3. Specify that files with the .REX extension are to be searched for when a command is issued, and what order to consider them:

```
SET PATHEXT=.REX;.COM;.EXE;.BAT;.CMD
```

In the above example, the search for a command will look for .REX files first in each directory examined, then .COM files, then .EXE files, then .BAT files, and finally .CMD files. (Which directories will be searched, and in what order, is determined by the PATH environment variable.) You can specify whatever order you like, but traditionally, REXX files are searched for first. You can even arrange the order of the built-in extensions (.EXE, .COM, .BAT and .CMD) to suit your own preferences. You can have as many user-defined extensions as you like, and they need not all be associated with REXX (as long as they're associated with *something*).

For instance:

```
SET PATHEXT=.REX;.COM;.EXE;.BAT;.CMD;.PDF;.REXX
```

This arrangement also allows for automatic viewing of Adobe Acrobat (.PDF) files (assuming an appropriate file type and association for it), as well as another REXX extension which will be searched for **last** (note that file extensions in Windows NT are not limited to 3 characters, so .REX is *different* from .REXX).

Advantages

- This technique does not produce spurious output.
- Commands invoked under this scheme can be passed semicolon, comma, and equal sign characters.
- There is no special initial set up required during each session.
- This method is space efficient: no additional disk space is used.
- Files with an extension of .REX can be both command and subroutine when using Object REXX.
- When a program using this mechanism is created, copied, renamed, or deleted, there are no additional tasks to be performed.
- This method works properly with the normal PATH search order.

Disadvantages

- Under Windows NT, .CMD and .BAT files cannot contain a REXX program, even if the appropriate association with a REXX language processor is made.
- This technique is not available under Windows 95/98, or DOS.
- Commands invoked using this technique cannot be redirected/piped.
- Files with an extension of .REX can be both command and subroutine when using PC-DOS 7/200 or OS/2 v1.3 REXX under Windows NT, provided the .REX extension is explicitly coded in the CALL instruction (e.g. CALL "MYSUBR.REX" parm1, parm2)
- When an Object REXX or PC-DOS 7 REXX program sets an environment variable in Windows NT, or changes a directory, that alteration to the machine state does not persist past the termination of the program. This also applies to the execution of PC-DOS 7 REXX in non-IBM versions of DOS, and versions of IBM PC-DOS prior to 7. This does not apply to REXX in OS/2, or PC-DOS REXX in DOS 7 or later, or to Quercus Personal REXX under DOS.

E. "/"* First

Some operating systems will automatically recognize a REXX program if it is found within a file with an extension designating standard procedures files. In PC-DOS 7/2000, this extension is .BAT; in OS/2, it is .CMD. The alternate shells available from J.P. Software (4DOS, 4NT, 4WIN, and 4OS2) also have this capability for .CMD files, but not .BAT files. This capability must be provided by the operating system, there is no other way to obtain this behavior.

Advantages

- The operating systems which support this mechanism dispatch the REXX language interpreter in such a fashion that changes to environment variables and directory changes persist after the REXX program completes.
- This implementation is space efficient: no additional space is required for REXX programs.
- When a REXX program is created, copied, renamed, or deleted under these operating systems, there are no additional tasks to be performed.
- REXX program files run under these operating systems can be used both as commands and subroutines.
- Command output can be redirected or piped under these operating systems.
- These shells do not produce spurious output when REXX programs are executed.
- There is no special initial set-up which must be performed each session to support each REXX program.
- This method works properly with the normal PATH search order.
- These operating system passes semicolon, comma, and equal sign characters to REXX commands.

Disadvantages

- These operating systems may not be the one(s) you need to use. Many applications require one of the 32-bit Windows operating systems.

F. Alternative Shell

It is possible to install an alternative shell to replace the one shipped with DOS, OS/2, Windows 95/98, or Windows NT. It is not necessary to use the same shell on all platforms. This section explicitly refers to the alternate shells available from J.P. Software: 4DOS, 4NT, 4WIN, and 4OS2. See the product site at <http://www.jpsoft.com/>. These products have won several awards. I would be interested to learn of alternative shells from other sources which can address the problems of executing REXX commands implicitly. These shells seems to do everything well (at least with respect to REXX -- I have no comment on other aspects).

Those J.P. Software shells which run under Windows support the Windows-based association mechanisms. All of these shells provide their own capability to associate an extension.

Advantages

- One of the nicest things about these shells is that they check .CMD and .REX files to see if they start with "/*". If one does, it is passed to the REXX interpreter.
- These shells are space efficient: no additional space is required for REXX programs.
- When a REXX program is created, copied, renamed, or deleted under these shells, there are no additional tasks to be performed.
- REXX program files run under these shells can be used both as commands and subroutines.
- Using these shells, your REXX program can permanently set environment variables and change directories under Windows 95/98, Windows NT and both older versions of PC-DOS and non-IBM versions of DOS.
- Command output can be redirected or piped.
- These shells do not produce spurious output when REXX programs are executed.
- There is no special initial set up which must be performed each session to support each REXX program.
- This method works properly with the normal PATH search order.

Neutral

- There is no difference in the handling of semicolon, comma, and equal sign characters under these shells than their IBM or Microsoft counterparts.

Disadvantages

- These shells are not free. They are the result of a great deal of work, and contain many features beyond facilitating the execution of REXX.
- These shells do not support the PATHEXT facility, even under Windows NT.