



Release 3.0

1) L'**ARJ** è un compattatore, cioè un programma capace di comprimere i dati in modo da renderli più maneggevoli e meno ingombranti (dove per ingombrante si intende in termini di spazio in Kilobytes nel Disco Rigido). Facciamo un esempio per capirci: mediante l'opportuna sintassi (che spiegherò poi) supponiamo di lanciare l'**ARJ** su un gruppo di, ad esempio, 50 files che occupano in totale 30 MegaByte, ...ebbene in risposta al nostro comando noi otterremo un SOLO file che occupa (ad esempio) 10 Megabyte!!: ecco cosa vuol dire usare un COMPATTATORE = risparmiare spazio e tempo; in tal caso il fattore medio di compressione sarebbe stato di 33% (= 10mega/30mega %), cioè le dimensioni attuali sono il 33% di quelle originali. Si noti che alle volte tale rapporto può essere migliore o peggiore: se ad esempio il mono-file fosse stato di 29 MegaByte il rapporto medio di compressione sarebbe stato di 96% (=29/30 %), cioè il mono-file compresso occuperebbe praticamente lo stesso spazio di tutti quelli non compressi. L'unico guadagno in casi simili é quello di aver raccolto sotto un solo file tanti altri, cosa che comunque ha i suoi vantaggi ...facciamo un esempio: abbiamo 500 ricette da cucina in formato Word e dobbiamo darle ad un nostro amico; abbiamo due scelte:

- A) copiamo, sempre che ci stiano , tutti e 500 i files su un dischetto (pensate a quanto tempo ..1 file copiato.. 2 file copiato500..)
- B) usiamo l'ARJ e li comprimiamo-raccogliamo in un unico file che chiameremo (ad ex.) RICETTE.ARJ, cosicché occuperà meno spazio (forse riusciamo anche a metterlo in un solo dischetto) e inoltre sarà sicuramente più "maneggevole" (in senso metaforico).

2) Pensate che per fare tutto quanto ho accennato sopra, basta un solo piccolo file eseguibile (= file con estensione EXE) di nome **ARJ.EXE**; detto file va visto come se fosse un nuovo comando DOS e quindi è necessario conoscerne la grammatica per poterlo utilizzare pienamente. Innanzitutto il file andrà copiato nella directory **C:\DOS** (per sistemi Win3.x) o nella directory **C:\WINDOWS\COMMAND** (per sistemi Win9x).

----- per ESPERTI -----

Infatti, presumibilmente, tale directory è inserita nella stringa **PATH=...** del file **AUTOEXEC.BAT** presente nella Root del vostro hard disk; pertanto quando digiterete **ARJ + INVIO**, ovunque voi siate, otterrete la risposta dell'**ARJ**, come se fosse un comando del **DOS**, appunto!!.

A questo punto provate a battere:

ARJ -? + INVIO

ed in risposta otterrete 7-8 pagine di "**Help in Linea**" (=termine tecnico) a cui potrete sempre riferirvi in futuro mediante la procedura sopra descritta.

3) La struttura del comando da impartire, quando vogliamo **COMPRIERE**, è:

C:\DIRECTORY_VOLUTA> ARJ [COMANDO] [DOVE,CHECOSA] [FUNZIONE]

Ove per **DIRECTORY_VOLUTA** si intende che il comando deve essere lanciato **NELLA** directory che si intende comprimere; **DOVE** indica la directory ove troveremo il file compresso (risultato del comando) e **CHECOSA** indica il nome del suddetto file (lo decidiamo noi).

In [COMANDO] metteremo la lettera " A " che significa che vogliamo comprimere (i comandi disponibili sono molti, basta digitare ARJ/? per vederli, per ora basta sapere che per comprimere ci vuole " A ", poi vedremo gli altri), mentre [FUNZIONE] indica le modalità con cui vogliamo che la compressione sia realizzata (le vedremo).

Se invece vogliamo DECOMPRIERE, allora il comando è:

C:\DIRECTORY_VOLUTA> ARJ [COMANDO] [DOVE,CHECOSA] [FUNZIONE]

DIRECTORY_VOLUTA indica la directory ove vogliamo trovare il risultato della nostra decompressione; DOVE,CHECOSA rappresentano rispettivamente la directory ove si trova il file da decomprimere, e il nome di questo file; il [COMANDO] questa volta sarà la lettera " X " (vale il discorso fatto per la compressione). Infine [FUNZIONE] indica nuovamente le modalità con cui la decompressione sarà realizzata.

4) L'esperienza è la miglior maestra in questo campo quindi ritengo che il miglior aiuto che si possa dare è fornire il lettore di una serie di esempi polivalenti; per tutti gli altri scopi è sempre possibile leggere l'Help in linea e comunque, tanto per averne un'idea di massima, vi elenco alcuni comandi disponibili:

ARJ T testa l'integrità degli archivi, cioè verifica che non siano danneggiati (ad esempio per colpa di un floppy-disk marcio.
ARJ I verifica l'integrità di sé stesso, cioè l'ARJ controlla che ARJ.EXE non sia danneggiato .
ARJ L GATTO.ARJ lista il contenuto dell'archivio GATTO.ARJ.
..... etc

Spero sia ora chiaro come i comandi disponibili siano davvero tanti.

5) Vediamo adesso qualche esempio educativo! ;-)

C:\PIPP0> ARJ A C:\ZORRO\PROVA.ARJ

Comprime tutti i files, ma NON le subdirectory, che si trovano nella directory C:\PIPP0 , e il risultato della compressione è un file di nome PROVA.ARJ che troveremo nella directory C:\ZORRO (si noti che se la directory C:\ZORRO non esiste, l'ARJ sceglie per default (=automaticamente) la root (cioè C:\), cioè il file PROVA.ARJ lo troviamo nella directory C:\ , anziché C:\ZORRO).

C:\PIPP0> ARJ A C:\ZIPPO\PROVA.ARJ -r

Comprime tutti i files, COMPRESSE le subdirectory, che si trovano nella directory C:\PIPP0 (è ora chiaro l'effetto della funzione " -r ") e il risultato della compressione è un file di nome PROVA.ARJ che troveremo nella directory C:\ZIPPO .

C:\TOPO> ARJ A C:\ZEBRA\PINOLO.ARJ -r -m3

Comprime tutti i files, COMPRESSE le subdirectory, che si trovano nella directory C:\TOPO e il risultato della compressione è un file di nome PINOLO.ARJ che troveremo nella directory C:\ZEBRA . L'effetto della funzione " -m3 " è di comprimere leggermente meno i dati in cambio di una maggior velocità nell'eseguire la compressione. Cioè senza la funzione -m3 ,l'ARJ comprime in modalità automatica che è veloce e mediamente buona. Con -m3 sarà velocissimo ma con un fattore di compressione leggermente peggiore.

C:\MANO> ARJ A A:\PAOLO\GATTO.ARJ -r -va

Comprime tutti i files, COMPRESSE le subdirectory (-r), che si trovano nella directory C:\MANO e il risultato della compressione è un file di nome GATTO.ARJ che troveremo nella directory A:\PAOLO nella unità A: . L'effetto della funzione " -va " è il seguente: se il risultato della nostra compressione, cioè il file GATTO.ARJ , occupasse 2 MegaByte e noi non avessimo messo -va , poiché un floppy disk contiene 1,44 MegaByte, dopo un po' l'ARJ si sarebbe interrotto dicendo: "DISK FULL !!" (= Disco Pieno !!). Invece se avessimo messo -va , non appena il floppy-disk si riempiva (=non c'era più spazio disponibile per copiarci ulteriori dati), l'ARJ si sarebbe fermato dicendo : "OK TO PROCESS NEXT VOLUME (1)" (=Sono pronto a realizzare la prossima parte) ,ove tra parentesi (nell'esempio sopra (1)) è il numero che, vedremo, avrà l'archivio creato come ultimo carattere dell'estensione).

Il concetto è semplice: se l'**ARJ** si trova a dover comprimere un grosso quantitativo di dati, è molto probabile che non riuscirà a creare un archivio più piccolo di un floppy-disk (cioè 1.44Mb), ma allora come fare?? Era necessario trovare un sistema per potere creare archivi su molti dischetti (= ARCHIVI MULTIPLI) senza confusione. A mio avviso fu scelto un sistema geniale: per poter distinguere i files, l'ARJ nomina automaticamente gli archivi multipli nel modo seguente: GATTO.ARJ GATTO.A01 GATTO.A02 GATTO.A03 che corrispondono rispettivamente al primo, secondo, terzo, quarto, ..dischetto !!. E' ora chiaro cosa intendevo per ultimo carattere dell'estensione: A01, A02, ..3, ..4, ...etc.. Si noti che se tutti i file si chiamassero GATTO.ARJ non saremmo capaci di distinguerli tra di loro, inoltre non potremmo neanche copiarli nella stessa directory dell'Hard-Disk perché si sovrascriverebbero l'un l'altro cancellandosi !!!.

Quindi, ritornando al nostro esempio: quando l'**ARJ** si interrompe dicendo "OK TO PROCESS NEXT VOLUME (1)", bisognerà cavare il dischetto contenuto nel drive A: (ed eventualmente scrivere a matita sopra l'etichetta GATTO.ARJ), sostituirlo con uno VUOTO, e premere INVIO : l'ARJ continuerà la sua opera di compressione sul nuovo floppy e si fermerà, quando anche quest'ultimo sarà pieno (è chiaro che il dischetto appena riempito contiene GATTO.A01), con il messaggio "OK TO PROCESS NEXT VOLUME (2)"...e così via fino all'ultimo dischetto che servirà all'ARJ per comprimere quanto richiestogli !! (NOTA BENE: bisogna stare un po' attenti, infatti non si può sapere prima quanti dischetti occorreranno all'ARJ per realizzare un archivio, l'unica cosa certa è che il totale non supererà le dimensioni di ciò che si sta' comprimendo; per capirci, se vogliamo archiviare un totale di 14 Megabyte, al massimo serviranno 10 dischetti !!).

C:\LETTO> ARJ X A:\POLLO\PICCHIO.ARJ

Decomprime nella directory C:\LETTO il file compresso PICCHIO.ARJ che trova nella directory A:\POLLO nella unità A: .

C:\LAMA> ARJ X A:\TGM.ARJ -va

Decomprime nella directory C:\LAMA il file compresso TGM.ARJ che trova nella directory A:\ (cioè la root) della unità A: . L'effetto della funzione -va è il seguente : se l'archivio TGM.ARJ è unico non succede nulla, se invece il file TGM.ARJ è il primo di un Archivio Multiplo (TGM.ARJ TGM.A01 TGM.A02 ...) allora le cose cambiano parecchio: se non avessimo messo -va , una volta processato TGM.ARJ, l'ARJ si sarebbe interrotto dicendo "CONTINUE ON NEXT VOLUME" (=continuo a decomprimere col prossimo dischetto che mi fornisci) e a noi sarebbe toccato, dopo aver cambiato dischetto, di riscrivere tutto il comando con il nome TGM.A01 e così via (avremmo cioè dovuto riscrivere ARJ X A:\TGM.A01 + INVIO). Se invece avessimo messo -va , una volta processato TGM.ARJ, l'ARJ si sarebbe interrotto dicendo "OK TO PROCESS NEXT VOLUME (1)" assumendo che il file successivo terminasse con il numero 1 (cioè TGM.A01), a noi sarebbe così toccato semplicemente di cambiare dischetto nel drive e premere Y (di YES) e INVIO, e l'ARJ sarebbe ripartito a processare il nuovo dischetto, evitando a noi di dover riscrivere per OGNI disco la sintassi necessaria, ... e così via fino all'ultimo dischetto.

C:\DESCENT> ARJ A C:\FLOP\DESC.ARJ -r -v1440

Comprime tutti i files e le subdirectory (-r) della directory C:\DESCENT in un archivio di nome DESC.ARJ che si troverà nella directory C:\FLOP . Ipotizziamo che la directory C:\DESCENT contenga un sacco di file: è molto probabile che l'archivio DESC.ARJ sia più grande di 1.44 Mb (= le dimensioni di un floppy-disk), pertanto se non mettessimo alcuna opzione (tranne -r) ci ritroveremmo un archivio che non sarebbe spostabile dall'hard disk (sempre che l'hard disk sia riuscito a contenerlo !!).

Se avessimo messo " -va ", l'**ARJ** avrebbe semplicemente controllato che la unità di destinazione (cioè quella che contiene la directory in cui troveremo l'archivio: nel nostro caso quindi C:) avesse abbastanza spazio libero per contenerlo; pertanto potevano succedere due cose:

A) L'hard disk ha molto spazio libero e alla fine troviamo il nostro archivio DESC.ARJ dove lo volevamo (tuttavia così grosso da non poter essere spostato su un floppy disk).

L'hard disk é quasi pieno e così, mano a mano che l'**ARJ** crea l'archivio DESC.ARJ, lo spazio libero diminuisce. Quando l'hard disk é pieno allora l'**ARJ** si interromperà chiedendo di cambiare volume (é l'effetto della opzione -va) ma noi non possiamo sostituire un nuovo disco rigido !!! (nota che se non avessimo messo neanche " -va " allora l'**ARJ** si sarebbe fermato dicendo DISK FULL !!).

L'opzione " -v1440 " costringe l'ARJ a creare archivi di 1.44Mb (=dimensioni di un Floppy): cioè quando DESC.ARJ diventa di 1.44 Mb , allora l'**ARJ** si ferma, crea DESC.A01, e continua; quando DESC.A01 diventa 1.44 Mb allora l'ARJ si ferma, crea DESC.A02, e così via finché tutti i dati sono stati compressi !! L'utilità di questo comando é che la compressione da C: a C: é molto più veloce di quella da C: ad A: , e così si creano gli archivi sull'hard disk e poi li si copia, uno ad uno, sui Floppy Disk, con un semplice comando COPY del DOS!!. Ora supponiamo di aver ottenuto, dalla compressione descritta nell'esempio sopra, 10 archivi .ARJ (che quindi abbiamo provveduto a copiare su 10 Floppy Disk, visto che l'esempio mostrava come questi archivi venissero creati in C:, cioè l'hard disk) e che vogliamo decomprimerli nell'hard disk di un nostro amico.

Supponiamo inoltre di volerli decomprimere da C: a C: , poiché A: , cioè il floppy, è lento. Decidiamo di copiare tutti i 10 floppy nella directory (ad ex.) C:\AAA : a questo punto ci ritroviamo una directory (C:\AAA) con al suo interno 10 archivi .ARJ siffatti: DESC.ARJ , DESC.A01 , DESC.A02 , , DESC.A09 (sono dieci perché il primo, DESC.ARJ , vale come numero zero !!). Volendo ripristinare il programma archiviato nella directory C:\DESCENT (perché ci piace così !) innanzitutto la creeremo (con il comando DOS: " MD DESCENT " digitato al prompt dei comandi C:>) poi entreremo in C:\AAA (con il comando DOS : " CD ") e digiteremo:

C:\AAA> ARJ X DESC.ARJ C:\DESCENT -va -jyacv

Con l'opzione " -va " costringiamo l'ARJ, quando finisce di scompattare un archivio, a chiederci quello successivo: ma poiché tutti gli archivi necessari alla decompressione di DESC.ARJ sono già disponibili nella directory C:\AAA (glieli abbiamo copiati noi), noi dovremmo semplicemente digitare " Y " (= yes) a tutte le richieste dell'ARJ ! Ebbene l'opzione " -jyacv " ha il significato di evitare che l'ARJ ponga le domande seguenti:

- APPEND AT POSITION XYZ
- CONTINUE ON NEXT VOLUME
- CREATE DIRECTORY

Infatti con l'opzione " -jyacv " facciamo sì che l'ARJ si risponda " YES " da solo!! (e noi risparmiamo tempo).

N.B.1 = I nomi scelti per le directory o gli archivi sono completamente inventati da me!! é chiaro che il comando andrà adattato caso per caso alle singole necessità!!.

N.B.2= Quando scrivo la prima riga di un esempio, la imposto così:

C:\NOME> ARJ etc

Sia chiaro che scrivere C:\NOME> ARJ.... vuol dire che il comando ARJ deve essere impartito all'INTERNO della directory C:\NOME e NON che C:\NOME> faccia parte del comando stesso!! Se il prompt dei comandi indica (ad ex.) " C:\> " e voi dovete andare nella directory C:\GIOCHI allora digiterete al prompt dei comandi:

" CD GIOCHI + INVIO "

e il prompt indicherà " C:\GIOCHI> " : ora siete all'INTERNO della directory C:\GIOCHI !!.

6) Sono disponibili una barca di funzioni, basta vedere l'Help in linea per rendersene conto; tanto per capirci ecco qualche esempio:

- i1 = dà una barra grafica, anziché la percentuale, che mi indica il "progredire della compressione" (é più facile a vedersi che a dirsi).
- jya = sopprime la domanda di append che viene posta dall'ARJ in caso di archivio multiplo.
- je = crea un archivio Auto-Estraente, cioè un eseguibile che, una volta lanciato si auto-decomprime senza bisogna di avere l'ARJ o di darne alcun comando.
- m0 = non usa la capacità di compressione: così facendo l'ARJ si comporta, né più né meno, come un normale programma di backup.
- jt = verifica l'integrità dell'archivio mano a mano che lo crea: in pratica serve quando si archivia direttamente su floppy-disk e non si é sicuri che i dischetti siano buoni.
- a1 = include nell'archivio ogni file e subdirectory che incontra, sia cioè quelli/e vuote che quelli/e nascoste o di sistema.
- ? = visualizza l'Help in linea (l'abbiamo già visto !)

.....etc.....

7) Riporto per semplicità l'Help in linea dell'ultima versione dell'*ARJ* così che possa essere letto con calma nell'ipotesi che questo documento venga stampato:

ARJ -? + INVIO:

Manuale ARJ

ARJ (tm) 2.70 Copyright (c) 1990-99 ARJ Software, Inc. Sep 30 1999
*** This SHAREWARE program is NOT REGISTERED for use in a business, commercial,
*** government, or institutional environment except for evaluation purposes.

Long filenames enabled
Processing archive: C:\WINDOWS\COMMAND\ARJ.EXE
Archive created: 1999-09-30 15:28:22, modified: 1999-09-30 15:28:22

Example ARJ/ARJ32 commands:

Add files minus directory structure:	ARJ a -e archive *.*
Add two files to archive:	ARJ a archive name1 name2
Add files to archive with verification:	ARJ a -jt archive *.*
Add files with archive damage protection:	ARJ a -hk archive *.*
Add files with maximum compression:	ARJ a -jm archive *.*
Add several directories of files:	ARJ a archive ... dir1 dir2 dir3
Add thousands of files to an archive:	ARJ a -hml -r archive *.*
Add a chapter to an archive:	ARJ ac archive c:\dir*.* -r
Comment archive header only:	ARJ c archive -zcmt.fil
Convert standard archive to chapter:	ARJ cc archive
Convert chapter archive to standard:	ARJ y archive -jbo
Delete the last chapter from an archive:	ARJ dc archive *.*
Strip archive comment only:	ARJ c archive -zNUL
Extract files from archive:	ARJ e archive
Extract maintaining directory structure:	ARJ x archive
Extract new and newer files without query:	ARJ e archive -u -y
Extract subdirectory from archive:	ARJ e archive subdir*.* -pl
Extract files to directory:	ARJ x archive destdir\ *.*
Extract files to directory:	ARJ x archive *.* -htdestdir
Extract files to directory with space:	ARJ x archive "Program Files\ "
Extract files to directory with space:	ARJ x archive "-htProgram Files"
Extract a Win95 archive to plain DOS:	ARJ x archive -hf2
List files in archive:	ARJ l archive
Move files to archive:	ARJ m archive *.doc
Move files from archive:	ARJ e -d archive *.doc
Recover files from a damaged archive:	ARJ x archive -jr
Test integrity of files in archive:	ARJ t archive
Test multiple ARJ archives:	ARJ t *.ARJ -hr
Add files to a multiple volume archive:	ARJ a -va a:archive *.*
Create up to 999 archive volumes:	ARJ a -va a:archive.001
Extract from a multiple volume archive:	ARJ x -v a:archive
Create a multiple volume self-extractor:	ARJ a -va -je a:archive *.*
Convert archive to self-extractor:	ARJ y -jel archive

Backup drive C to drive A:
ARJ a A:\backup C:\ -a1 -b2 -r -vvas -wC:\ -js -jt -hk
Restore backup on drive A to drive C: ARJ x A:\backup C:\ -vv -y

More detailed ARJ/ARJ32 information:

Usage: ARJ <command> [{/|-}<switch>[-|+<option>]...] <archive_name>[.ARJ]
[<base_directory_name>\] [<!list_name>|<path_name>|<wild_name>...]

or

ARJ @response_filename [-& | -+ | -g? | -hy | -gpassword]

<Commands>

ac: Add Chapter to chapter archive	m: Move files to archive
cc: Convert archive to Chapter archive	n: reName files in archive
dc: Delete last Chapter from archive	o: Order files in archive
a: Add files to archive	p: Print files to standard output
b: execute Batch or dos command	q: repair ARJ-PROTECTED archive
c: Comment archive files	r: Remove paths from filenames
d: Delete files from archive	s: Sample files to screen with pause
e: Extract files from archive	t: Test integrity of archive
f: Freshen files in archive	u: Update files to archive
g: Garble files in archive	v: Verbosely list contents of archive

Manuale ARJ

i: check Integrity of ARJ.EXE	w: Where are text strings in archive
j: Join archives to archive	x: eXtract files with full pathname
k: remove obsolete bacKup files	y: copY archive with new options
l: List contents of archive	

<Main defaults>

Save path information in archive
Save long filenames
Save/restore file date-time created and accessed in Win95/98
(ARJ32) Save/restore file date-time created and accessed in Win95/98/NT
Don't save drive and root in path information
Display comment ANSI sequences with built-in display handler
Prompt before overwriting output files
Use method 1 compression
Use binary mode
Use ! as list file symbol
The "ac" command is an abbreviation for "u -u2 -jb"
The "cc" command is an abbreviation for "y -jb"
The "dc" command is an abbreviation for "d -jb"

(ARJ32) Save/restore file date-time created and accessed in Win95/98/NT

ARJ32 uses OEM codepage by default. The "-hy" option sets ARJ32 to use the ANSI codepage which is the Windows graphical environment default.

<User prompt responses>

Yes - yes
No - no
Quit - abort out of ARJ
Always - always assume yes for current type of query
Skip - always assume no for current type of query
Global - always assume yes for all queries except diskette volume prompts
Command - prompt for and execute one DOS command

<Switches>

-: disables switch char	p: match using full Pathnames
+: inhibits ARJ_SW usage	p1: match Pathname with subdirs
+var: set environment variable	q: Query on each file
@: set 1 token per response file line	r: Recurse subdirectories
!: set list char (!)	s: set archive time-Stamp to newest
&: set batch critical error handler	s1: save original time-Stamp
#: select files by number	s2: set archive time-Stamp
\$: add/extract volume label	s3: save both original time-Stamps
\$A: add/extract label to drive A	t: set file Type (default 0)
a: allow any file Attribute	t0: set binary file type
a1: any files and directories	t1: set C text file type
b: Backup changed files	tlf: force C text file type
b1: Backup changed, reset archive bits	tlg: set C text with graphics
b2: only reset archive bits	u: Update files (new + newer)
b3: reset archive bit during restore	u1: Update files (new + OLDER)
b4: do not restore bits, reset arc	u2: Update files (new + different)
b5: do not restore any file attributes	u3: Update files (new + CRC mismatch)
c: skip time-stamp Check	v: enable multiple Volumes
d: Delete added files	v360: build 362000 byte volumes
asks permission before deleting	v50K: build 50000 byte volumes
e: Exclude paths from names	va: auto-detect space available
e1: Exclude base dir from names	vda:*.*: delete a:*.*
f: Freshen existing files	vi: inhibit diskette change test
f1: Freshen with OLDER files	vr50K: reserve 50000 bytes of
f2: Freshen with DIFFERENT files	space on first volume
f3: Freshen with CRC mismatched files	vs: provide DOS command prompt
g: Garble with password	vsCMD: execute CMD before each vol
gstew: garble with password stew	vv: beep between volumes
g?: prompt for password	vw: keep Whole files in volumes
i: show no progress Indicator	vz: provide command with no echo
i1: show bar graph Indicator	v360,v720,v1200,v1440: abbrevs
i2: show percentage and bar graph	Volume options may be in any order

i3: total progress percentage
i4: total progress bar graph
i5: total percentage and bar graph
k: Keep a .BAK of ARJ archive
l: create List_name file
 lNAMES.LST: create NAMES.LST
m: with Method 0, 1, 2, 3, 4
 m0: store (no compression)
 m1: good compression (default)
 m2: less memory and compression
 m3: FAST! less compression
 m4: FASTEST! least compression
n: only New files (not exist)
o: On or after YYYYMMDDHHMMSS
 o: on today
 o19901225: on/after 12/25/1990
ob: Before YYYYMMDDHHMMSS
 ob: before today
 ob19901225: before 12/25/1990
od: no older than N Days
 od5: five or less days old
oa: After YYYYMMDDHHMMSS (last access)
 oa: on today
 oal19901225: after 12/25/1990
oab: Before YYYYMMDDHHMMSS (last access)
 oab: before today
 oab19901225: before 12/25/1990
oc: After YYYYMMDDHHMMSS (created)
 oc: on today
 oc19901225: after 12/25/1990
ocb: Before YYYYMMDDHHMMSS
 ocb: before today
 ocb19901225: before 12/25/1990

<Shifted switches>

h#: append date string to name	hi: detailed display in Index files
h#1: append time string to name	hk: enable ARJ-PROTECT damage protection
h#2: append DHHMMSS to name	hk: default amount of recovery data
ha: ignore readonly Attribute	hk1: more recovery data
hb: select by file attribute/type	hk9: maximum amount of recovery data
a - archive bit set	disable ARJ-PROTECT when used with
b - archive bit clear	the "y" command
r - readonly	hl: return error for Listfile error
s - system	hm: increase file list capacity
h - hidden	hm!: redirect -hm temp file
d - directory	hm! : move temp file to XMS mem
n - normal file, not d,h,r,s,c,l	hm!e:\ : move temp file to e:\
c - chapter label	hn: specify non-existent filespec
l - volume label	hn####: set #### as filespec
f - abbrev for n,d,r,s,h	ho: Only extract files with matches
w - win95 long filenames	ho1: Only extract files without matches
hb: select all types	hq: ignore open access error on adds
hbsh: select hidden, system only	hq1: ignore open not found error on adds
hbd: select dirs only	hq2: ignore access and not found on adds
hc: execute DOS Command at ARJ start	hr: try to ignore archive errors
hcCLS: execute CLS	hs: disable file Sharing
he: skip test of security Envelope	ht: set Target directory
hel: set error on security Envelope	htD:\DESTDIR
hf: set short filenames only	hu: allow Update of volume archives
hf1: set longnames in file comment	hv: check for ARJ version
or extract with W95LNAME.nnn	hw: scroll save search filename display
hf2: use long filenames in DOS	hw1: display only filenames with matches
hf3: set longnames in file comment	hx: set default archive extensions
using shortname wildcards	hx.arj.sdn
hf4: set all files archived in Windows	hy: set ARJ32 to use ANSI codepage
as Windows long filenames	set ARJ to extract ANSI codepage
hg: specify garble module filename	archive
hgGARBLE.COM: set GARBLE.COM as	
name of garble module	

hg!: specify use of 40 bit encryption

j#:	store only CRC and not the file	jp:	Pause after each screenful
j#1:	store only date, size, attributes	jp50:	pause, set page size to 50
j#2:	restore file attributes	jq:	set string parameter
j#3:	restore file dates	jqstring:	set parameter to string
j#4:	restore file attributes and dates	jr:	Recover broken archive files
j\$:	do not process file DTA and DTC	jr1:	Recover badly broken archive files
ja:	set P command to display ANSI	js:	Store archives by suffix
ja1:	inhibit the display of comments		default is arj, arc, lzh, pak, rar,
jb:	set chapter backup processing		zip, zoo, j, and uc2.
jb0:	clear chapter archive status	js.zoo:	store only .zoo files
jb*:	select all chapter backup files	jt:	Test temporary archive by CRC
jc:	exit after Count of files	jt1:	Test CRC and file contents
jc5:	exit after 5 files	jt2:	Test only added file contents
jc nam1 nam2:	exit after 2 files	jt3:	Test only file date-time and size
jd:	ensure free Disk space	ju:	translate UNIX style paths
e -jd50K:	skip file if<50000 free	jv:	set Verbose display
l -jd1000:	set error if<1000 free	jv1:	set special verbose list mode
je:	create self-extracting archive	jw:	set extract output filename
je1:	create SFXJR archive	jwNEW.FIL:	output to NEW.FIL
jf:	store/use Full specified path	yx:	start at eXtended position
jf1:	store/use path minus drive	jx10000:	start at position 10000
jg:	[OBSOLETE]	jy:	suppress queries assuming Yes
jg1:	[OBSOLETE]	a -	skip append query
jh:	set Huffman buffer size	c -	skip create directory query
jh65535:	set to 65535 bytes (max)	d -	skip delete files query
jh2048:	set to 2048 bytes (min)	k -	skip disk space available query
ji:	create Index file	n -	skip new filename prompt
jiINDEX.FIL:	create INDEX.FIL	o -	skip overwrite file query
jk:	Keep temp archive on error	r -	erase all type-ahead before query
jl:	display only filespecs	s -	skip scanned enough text query
jm:	set Maximum compression mode	v -	skip proceed to next volume query
jm1:	set faster Maximum compression	y -	accept single character Y/N/A/Q
jn:	restart volumes at fileName	jycny:	skip create, new name queries
jnBIN\X.COM:	restart at BIN\X.COM		in single character input mode
jn:	get restart information from	jz:	supply file for file comment
	previously written index file	jzFIL.CMT:	use FIL.CMT for comments
jo:	query when updating archive file	jzNUL:	use to strip file comments
jo:	extract to unique Output names		
jol:	extract to unique Output names		
	keeping file extension		

Environment variable setting:

```
set arj_sw=-jyry -jv -il
set arj_sw=c:\arj.cfg
```

(ARJ32 uses ARJ32_SW instead of ARJ_SW)

```
set arj32_sw=-hy
```

ARJ DOS errorlevels:

```
0 -> success
1 -> warning (specified file to add to archive not found,
            specified file to list, extract, etc., not found,
            or answering negatively to "OK to proceed to next volume..."
            prompt)
2 -> fatal error
3 -> CRC error (header or file CRC error or bad password)
4 -> ARJ-SECURITY error or attempt to update an ARJ-SECURED archive
5 -> disk full or write error
6 -> cannot open archive or file
7 -> simple user error (bad parameters)
8 -> not enough memory
9 -> not an ARJ archive
10 -> XMS memory error (read or write)
11 -> user control break
```


Manuale ARJ

12 -> too many chapters (over 250)

The batch statement IF ERRORLEVEL 0 ... means
if the errorlevel is EQUAL TO or GREATER THAN 0.

If the first two characters of an ARJ comment are "))" in a self-extractor,
then the self-extractor will accept the appended line as its command line.

)) -o
will force the self-extractor to overwrite existing files.

This ARJ help information is not intended as a replacement for the ARJ user
documentation. See the files ARJ.TXT, INTRO.TXT, ERRORS.TXT, ... for detailed
information about using ARJ and ARJ32.

ARJ/ARJ32 file archiver utility. Copyright (c) 1990-99 ARJ Software, Inc.
All Rights Reserved.
ARJ is a trademark of our company.

ARJ IS DEDICATED TO GOD AND TO THE JUNG FAMILY. John 15:5

ARJ/ARJ32 LICENSE POLICY:

ARJ SOFTWARE, INC. DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ALL SUCH WARRANTIES
ARE EXPRESSLY and SPECIFICALLY DISCLAIMED.

READ THE ACCOMPANYING DOCUMENTATION FOR FURTHER DISCLAIMERS AND INFORMATION.

YEAR 2000 COMPLIANCE STATEMENT:

The programs ARJ and ARJ32 are YEAR 2000 compliant and will not produce errors
processing date data in connection with the year change from December 31, 1999
to January 1, 2000 or with the leap year factor from February 28, 2000 to
February 29, 2000 when used with accurate date data. This statement applies
to ARJ versions 2.30, 2.41a, 2.50a, 2.60, and versions under development with
one exception.

In ARJ versions 2.60 and lower, the command option "-h#" will not generate the
expected archive name in the year 2000. This is not a problem with releases
of ARJ and ARJ32 since 2.60.

No special user changes or activities are required for this compliance. No
programming or configuration changes are required either.

Disclaimer: The Statement of Compliance refers to ARJ products as delivered
by ARJ Software, Inc. The Compliance Statement does not apply to third party
add-on features or products. The Compliance statement does not constitute a
warranty or extend the terms of any existing warranty.

This statement supersedes all previous ARJ year 2000 compliance statements.

ARJ and ARJ32 are separate shareware products and their use in a business,
commercial, government, or institutional environment require separate
registrations. The inclusion of the ARJ or ARJ32 software (ARJ, ARJ32,
DEARJ, ARJSFX, ARJSFXJR, ARJSFXV) as part of a software and/or hardware
package to be distributed always requires a special license. The
registration of ARJ or ARJ32 for personal at-home use is NOT required.
Instead, we ask that these at-home users share ARJ with a friend. The
registration fee for the use of either ARJ or ARJ32 (not both) on one
computer for internal use only is \$45 including shipping. This price is
valid until January 1, 2001. Site licenses and distribution licenses are

available. Business, commercial, government, and institutional ARJ/ARJ32 users are permitted a free 30 day period for the evaluation of ARJ/ARJ32.

For more information concerning ARJ/ARJ32, see the accompanying documentation or contact:

ARJ Software, Inc.
P.O. Box 249
Norwood MA 02062
USA

Fax : 1-781-769-4893
Email: robjung@world.std.com
Web : <http://www.arjsoftware.com>

*** NOTE NEW MAILING ADDRESS ***

8) In commercio esistono moltissimi altri compattatori, quali ad esempio PKZIP.EXE, LHA.EXE, RAR.EXE, PAK.EXE, ZOO.EXE ... etc... tuttavia mi sento di consigliare vivamente l'**ARJ** in quanto lo ritengo essere il più completo per quel che riguarda le opzioni disponibili e la semplicità di utilizzo delle stesse. E' chiaro che questo manuale, distribuito in forma completamente gratuita, non ha assolutamente la pretesa di voler in alcun modo sostituire la documentazione specialistica ma solamente affiancarla, specie nei confronti dei primi utenti di PC che si trovano generalmente spiazzati quando si parla di Compressori & Company.

9) Mi auguro, ove possibile, di essermi reso utile. In tal caso vi invito a distribuire liberamente questo documento accompagnandolo all'**ARJ** stesso in modo che l'uso del compressore diventi una pratica diffusa e familiare quanto il mouse.

Se volete scrivermi, un mio indirizzo e-mail è: io@stordito.it , accetto tutto tranne le critiche! ;-)

Paolo ® 2000