■ DISTRIBUTION INFORMATION

DBE.EXE Directory Backup Extension PROGRAM VERSION 4.40

DBE
Release 4.40
Apr. 11, 2011

Copies are available
from the author or
at the links below:

GAMerritt1@GMail.com
GAMerritt@HotMail.com

Notes and general information are available at:

http://sites.google.com/site/glennmerritt1/dbe
http://cqvis.yolasite.com/gamerritt
http://cqvis.yolasite.com/discqdir

DBE DOCUMENTATION FILE

CURRENT OPTIONS 041111

■ KEYBOARD OPTION KEY NAMES/FUNCTIONS

The numeric keypad keys are symbolized by an upward-pointing arrow followed by a number. This signifies that, assuming NumLock is OFF, one of the Shift keys is pressed and held down while the number key is pressed. (If NumLock is on, pressing Shift cancels the effect of NumLock; the Shift key should not be pressed if this is the case.)

The caret symbol ^ represents the Ctrl key, which is held down while the key having the name following the caret symbol is pressed. The same principle holds for the ª symbol, which represents the Alt key, and the ↑ symbol, which represents either one of the Shift keys.

The Ins and Del keys, as well as the Plus and Minus keypad keys, are used in various situations for tagging files or groups of files.

The cursor and paging keys normally page or scroll up or down through file lists, but when NumLock is on or Shift is used, the SelectDisk option is activated, and this option is primarily oriented toward the use of the numeric keypad, as described below.

When the state of the Shift and NumLock keys are different (not both on or off) the keypad keys work as numbers. The 5 keypad key is only sensed if these states differ (except by certain TSR programs which contain special keyboard interrupt handlers).

The central keypad keys (numbers 1..9) will have different responses by DBE, depending on BOTH the NumLock and Shift key states, but the Plus, Minus, and Insert keypad keys' functions depend on WHETHER either of the Shift keys are pressed: If pressed, the <SLTOI>, <SLTOP>, and <CTOM> flag states are set; otherwise these keys operate on file tags.

In general, the 4..9 keys are used to select directories and the 1..3 set the number of files per line displayed in each window. If NumLock is off, one of the Shift keys must be pressed to enable these responses; otherwise these keys act like the normal cursor and paging keys.

Some duplication (or similarity) of functions exists between the text and the keypad areas of the keyboard, based on the assumption that the keypad area is used more for navigation and individual file/directory operations and the text area is used for the more complex processes of selection, grouping, sorting, and setting targets, but because navigation is necessary no matter what one is doing, the < , . and > keys are also available for this, and the \ key is near enough to them that opening/closing the directory window and navigating both the directory list and the normal file lists is a bit quicker.

■ CURRENT FILE POINTER / VIEWING MODES

The current file is indicated by a selection bar and a small pointer which travels up and down along the divider between the two windows. If you wish to center the current file vertically in the window, press CtrlPgUp or CtrlPgDn, which will also scroll the window up or down, keeping the current file in the middle of the window if this is possible. Press Space to reset normal selection bar movement.

From one to three files may be displayed per window line. If the ▲ or ▼ keys are pressed, the current file is changed by the number of files per line, and if the ◄ or ► keys are used, the change will always be one file, forward or backward.

The selection bar and the small pointer change color for Selected files; in the Selected window (turn viewing of it on/off with the * key) the bar turns into a bright green highlight, except for when the Queue View mode is toggled on, when it becomes bright magenta.

The active window is pointed to by a ◄] or an [► because both windows will show a selection bar. (However, if an undocumented internal loop state control is set to terminate certain list traversals, this pointer will change to <- or -> and its color will be cyan instead of gray. There is a method of setting and clearing this control, but the 'screen background color' people who calumniated me and disputed my authorship of DBE don't know it.) The active window's selection bar has a white background, and the other window's selection bar has a cyan background. One may ask why two selection bars exist: Well, it's because every directory with files in it has a selection bar resting on either the first file or on the file it was on the last time an operation took place involving that directory. That is; every directory has a current file, as well as a starting file for the view windows, and DBE keeps track of which ones these are. Every time you navigate out of or back into a directory, that is where the selection bar will be until you move it or until the processing loop updates it (it lets you see which files are being processed by updating the selection bar to point to the current file ready for processing; sort of like a play-by-play of the action; although if you are using a system which queues disk/file write operations and performs them in burst mode, the actual file operations will not be in sync with the screen updates, and the data for the screen updates may never be sent to the video buffer because it was changed even before the buffer had a screen-full of data to be shown).

If you press ᵃA to associate files, however, the opposite window will have no selection bar, because associated files will be shown opposite each file in the current window; nothing will appear there opposite files for which no matches were found. Pressing / toggles between the Associate view and the normal view.

If you save a DBSYSCON.FIG configuration file on exit from DBE, the opposite window will be empty when you start up again if DBE reads it in at startup. Normally, unless you named two file specifications on the command line, DBE will show the same directory in both windows at startup (or one window will display the key help mnemonics); so this is a reminder that DBE hasn't actually read the disk but has read in a saved file/dir/disk list. If you had tagged any files or assigned targets to them, they would appear as such- one more clue to the fact that a configuration exists and is in use. Note that if you had read a directory and then cut files from the list before saving the configuration, only the files still remaining on the list would be present when you started DBE again, and you would have to re-scan the disk to see the complete list of files actually present on the disk.

For versions 4.3N+, you can always turn on CapsLock, NumLock, or ScrollLock, to be sure when starting up whether a FIG file is being loaded. If you want to start without one, run DBE with the command "DBE /", and DBE will then list the FILESET files (*.DBE) in the default directory if you do not enter a file specification at the prompt which appears at startup.

The Selected list appears as though it were a normal directory in a normal window, except that the Level name at the bottom of the screen changes to reflect the home directory of the current file. The * key toggles viewing of the Selected list on and off. If you are viewing the Associated list, the Selected list turns off in the window you are using, but not necessarily in the other (although it won't appear when Associated files are shown there).

If you press ᵃ= (AltEquals) the normal file view is replaced by one in which files and targets are shown together with their respective numbers according to the original ordering as they were added to their respective lists. The files are numbered separately for each directory, and this numbering does not change if you sort or rearrange the files; only re-indexing them can change this ordering. If you are using a number of different targets and need to check or reassign any of them, the ᵃ= file/target list view comes in handy.

Pressing ^◀┘ (CtrlEnter) shows the numbered target list. If you type a number when this list is being viewed, the target having that number will be set for the current file. If you only want to look at the target list but don't want to set a target, use Esc or ^◀┘ to exit the target list view. Pressing Enter sets the target to the current one (indicated by a selection bar).

These views normally stay on until you turn them off; the Selected list view turns on automatically every time you use S, ^S, U, or ^U to select or unselect files. This is to make sure you see whether there are files on it, because it can be global, with files from any directory or disk listed on it. The target list turns off if you select a target, because you really do need to see which file the target has been assigned to, and because multiple targets for single files are not supported.

The \ and ↑5 keys turn the SelectDisk window on and off. This sets the view in the current window to the directory you choose. If you set a target with the T option S suboption the SelectDisk window appears automatically, allowing you to set the current target to the directory you choose.

The '|' key, the shifted keypad '8' key, or the = key followed by PgDn, displays the Disk List, which shows the 'dingbats' representing occupied and projected amounts of disk space used. These blocks will appear when you have read a disk and assigned a target for one or more files to it which are also tagged for COPY or MOVE.

These viewing modes are listed when you press L as follows:

```
LIST:
        NORMAL   FILE/TARG   TARG   LEV   SEL   DSK   JMP   ASSOC
                    ª=       ^◀┘     \     *     =     J     /
```

You may then press NFTLSDJA to choose which mode to turn on.

If you press ? or turn ScrollLock on, the bottom of the screen shows the file flag status. This stays on if ScrollLock is on. It is accompanied (if there is enough room) by a full current file name/path including the DiskID or Volume Label shown at the top of the screen; and if there is a target assigned to the file, an ▶ character followed by the target disk ID or Volume Label and the name of the target directory will appear, provided the target disk was added to the memory list before the target was defined for the file AND the target type is <ToDir>. These displays stay on when the ª= view is being shown.

■ LIMITATIONS OF 8.3 FILENAMES

DBE is limited to about 3,000 files/directories on its memory list because of the program size and the limits of real-mode memory.

DBE is limited to file operations on files with pathnames no longer than 67 characters plus a 12-character filename.

DBE will copy long filenames much longer than that, but the path itself has to be within the 67-character length limit. The long filename is transferred after the file itself has been copied using its short filename, and the new file with the short name is simply renamed to possess the long filename the source file had. Because DBE can read and display long file and directory names (at least if the short form of the directory portion of the name is of length less than 128 characters), does not mean that it can open the files or perform any operations on them. Only if the short form of the directory name is of length less than 68 characters can DBE assign and open (or create) them. On average, this means that it will read and write files in directories up to 8 levels from the root. An obvious exception would be the following, where the file MAINDBE.BAK is 24 levels deep in the tree, and DBE can read/write all files in this directory because the length of the longest 8.3 filename and the path will always be less than 68:

```
E:\GM\TMP5\A1\VV\6\7\8\9\0\1\2\3\4\5\6\7\8\9\0\1\2\3\4\MAINDBE.BAK
```

Another example is the file "directories.acrodata" in this folder, which is 9 levels deep in the tree:

```
C:\Documents and Settings\All Users\Application Data\Adobe\Acrobat\
    8.0\Replicate\Security
```

DBE displays the disk ID followed by the path+name for the file as follows,

```
[GROKLK] C:\..\ADOBE\ACROBAT\8.0\REPLIC~1\SECURITY\directories.acrodata
```

although internally the short form for the path is 65 characters long:

```
C:\DOCUME~1\ALLUSE~1\APPLIC~1\ADOBE\ACROBAT\8.0\REPLIC~1\SECURITY
```

and the filename itself is "DIRECT~1.ACR". Altogether, the full path is 78 characters in length; so DBE can read and write the file.

■ LISTS: Level | Selected | Associated

DBE links memory lists of files into three lists: Level, Selected, and Associated lists. The Level (subdirectory) list is organized by Disk, so that each set of Levels corresponds to a directory tree or a portion thereof, belonging to a particular Disk or diskette. Each Level contains zero or more Entries, each of which contains a file name, date, size, attribute byte, and other information related to the status of the entry in one or more of the three memory lists. The Queue is primarily a numbered list which uses the Selected list functions for display, and which must be re-linked or numbered (by pressing QR or QL) after you make changes to it.

The Selected and Associated lists are subsets of the Level list which are created each time you perform a file Select or Associate. Since DBE runs in an 80x86 microprocessor's Real Mode and, thus, has no access to extended memory, the memory available to DBE consists of 640K-bytes minus the amount currently in use by other programs and by the memory image of the DBE program. In most cases, this results in roughly 256K-bytes of memory available for list storage. This limit on the available memory places restrictions on the number of files which can be included in the memory list, although the Selected and Associated lists are simply links, and do not require memory beyond that already allocated by the Level list. To use this limited memory efficiently, DBE allows the disk scan range to be limited in several ways, and also allows dropping of Entries from the list which are not of interest in order to re-use the memory in adding other Entries to the memory list. The disk scans can be limited to several different scan ranges, as follows:

■ DISK SCAN RANGES

The Volume and Tree scans span the entire disk directory tree, but the Tree scan does not add normal files to the list, and thus creates only a skeleton structure containing only Levels (subdirectories). On a large-capacity disk with thousands of files, this scan requires much less time and memory, and this type of scan is used by default on disks over 1.44 Megabytes in size when you use the Ctrl+DriveLetter scan option.

You may turn AutoScan on; and if you do, be aware that DBE will read every unread (or changed or partly-read) directory when it becomes the current one. If you're short on memory, keep AutoScan turned off (or use the \ key to open the SelectDisk view to move between directories; this allows you to skip the ones you don't want to read when AutoScan is on). If AutoScan is on, and you use \ to select an unread directory, DBE won't read it until you press Space (or if you exit the \ option using ^R ; : etc. it will be read as soon as you exit the \ loop) or another key.

The Volume scan will, provided there is enough memory available, add all of the entries in each disk directory on the tree to the Level list. If you try this option on a large hard drive and find that it consumes too much time or memory, you can try using the Tree scan followed by a Current or Subs scan of the particular branches of the tree you intend to work with.

The Root scan is unusual in that it reads only the root directory, but it adds both file entries and Levels to the Level list.

The Update scan re-reads a particular disk directory. This is used with disks which have no serial number, volume label, or DiskID, and which are write protected. Such disks are not 'remembered' by DBE, and are handled differently from disks which can be uniquely identified by these criteria. DBE v4.2 and above no longer attempt to use disk serial numbers provided by DOS or OEM formatting.

The Filespec scan is used when only certain files are to be added to the Level list, and can be used in conjunction with the Current and Subs scan ranges to limit the disk scan to a search for files matching only the file specification. An active Selected list can be used as a list of files to scan for using F4. The F5 and F6 keys perform the Filespec scan, with F6

reading subdirs as well as the current. Use AltF4..AltF6 to minimize the number of subdirs added to the list. These keys cause DBE to prune empty subdirs from the list during the scan if they have no subdirs and no files being scanned for were found in them.

A 'drag-and-drop' of a file icon onto DBE.EXE normally results in a search for the file on the drive containing that file, if that file isn't one of the special files recognized and handled by DBE in a special manner.

■ LIMITING DISK SCAN RANGES

If you want to limit scans by preventing them from adding certain paths to the memory list, use the IGNORE declaration in DBCONFIG.TXT to set the ignore flag (which is not the same as a file entry Ignore flag) for each such directory. "IGNORE C:\WINDOWS", "IGNORE C:\PROGRA~1", and "IGNORE C:\DOCUME~1" will, if those DOS directory name equivalents are unique on your system, prevent DBE from adding file and subdirectory entries belonging to "C:\WINDOWS", "C:\Documents and Settings", and "C:\Program Files" to its memory list. Since the contents of the folder "C:\Program Files" are relatively static, and, under Windows Vista, not used to save temporary user data, DBE doesn't need to bother with that folder location. The contents of C:\WINDOWS or C:\WINNT are closely tied to the operating system kernel and to the permanently installed drivers and system services, so DBE doesn't need to search there for anything. And, since DBE was designed for people who take charge of their storage media and create their own subdirectory structure, it doesn't really need to bother with the "Documents and Settings" subfolders where user data tied to logon sessions and to default Windows application operations is stored based on principles of operation and hierarchy defined by Microsoft and others. It IS a good idea, though, to use DBE to selectively back up particular files in your particular user folders, because most programs have dynamical links which update the contents of files when you perform certain operations on them, and some software systems won't allow you to save your work to arbitrary locations (usually due to the underlying XML or XSLT links which maintain project file updating) even though you may have a hunch that you're going to lose some work if you don't back a file up before making changes to it. (The worst part is when you use an IDE to back up a file and then discover that it is ignoring the original copy of the file and your editing changes are being made to the backup instead of to the original file you wanted to make changes to, and that your 'backup' copy is now part of the project and the original file ISN'T.) If you save the file using DBE, you can always cut-and-paste anything you lose later on back into the file you're working on, and you won't have active files scattered around in places where only your IDE knows where to find them. DBE isn't designed to maintain XML links or GUIDs, and you might as well regard it as a last resort when it comes to maintaining files which are under source control and versioning systems. But you CAN save yourself a lot of headaches if you back up critical work, a file at a time, whether it's under source control or not, so that if something DOES happen you can still have complete file contents to work with in patching up a mistake or recovering from a catastrophe such as a system crash or hard drive failure. You can record special information about the project in the COMMENTS block when you save a .FIG file on exit from DBE, which could easily include a 64-byte GUID string or an href to remind you of where the master file was located for a project.

■ KEYS FOR ACTIVATING DISK SCANS

Several keys (actually, a somewhat confusing variety of them) are used to activate disk scans over one of these ranges. Some allow editing of the file specification before a disk scan and also preserve the status of any entries from the disk director(y/ies) previously added to the memory list. CtrlF4

and ↑5 allow selecting of the Level to be made current in the active window, and if CtrlR or AltR are pressed to exit from the SelectDisk process, a scan will be made to update the current window from the disk. A few other keys simply update the current window from disk. See the DRIVES option (press O\) for ways to add/update disks to/on the list without performing a scan.

The following table gives the keys which perform disk scans, and notes whether they allow editing of the file specification, preservation of previous scan information, and so on.

	Clear	Preserve			↑5 ^F4 \
ScanRange	Reread		Edit	Set Window	Select Level
Current	^R ^F3	;	F4 F5 ↑7 ↑9 RC RF	F7 F8 ↑7 ↑9	; ↑8
Subs	ªR ^F6	:	ªF6 F6 RS		:
Tree	^T	' "	RT		" '
Root			RR		
Disk			RV		
Auto	^A..^M				

Note that the Auto scan, initiated by pressing Ctrl plus a drive letter, allows only drives from A:, B;, C:, etc., on up to drive M: to be scanned using this option. You may use the scans which allow you to edit the filespec in order to select drives above the M: drive.

Note also that the keypad numbers require that Shift be pressed if NumLock is off in order to activate the options associated with them.

DBE assigns a default ScanRange and Scope to each key, for each of the keys in the following table:

SCAN OP	SCOPE		
	Dir	Subs	Volume
Crnt	^F3 ^F4 ; F4 F7 F8 ^R RC RF ↑7 ↑8 ↑9		
Tree	'	"	^T RT ^A..^M
Root	RR		RD
Spec	F5 ^F5 ªF5	ªF4 ªF6 (♦)F4	
Subs		F6 ^F6 ªR RS :	
All			RV ^A..^M

(♦) Selected

In general, the Alt and Ctrl key combinations do not preserve the states of files in the list when their directories are re-read; i.e., their tags and status flags are cleared. In addition, they also disappear from the selected list and the Queue. The ^T key (Scan Tree) does not affect the file states or the selected list.

The text key scan options RC RD RF RR RS RT and R^D do not alter the selected list or the Queue, although they untag the files read. The ; : ' and " options preserve tags and file status flags, and copied files and new files remain distinguished by visual indicators. This also means that in order for you to copy files again in the processing loop, you need to manually reset the status flags or use the R+CDFRST^D reread functions to reset them.

The text key OR option (Options Re-enable) clears the Copied status and lets you choose whether target status affects the qualification for the re-enable of copying for a group of files. If you need to re-enable copy repeatedly for the same files, either save a .FIG file to be read in at startup or a FILESET file to be read in at any time.

(You can set the tags for groups of files by adding them to the selected list with the TAG suboption enabled, which sets the tags to the value you specify as the files are added to the selected list.) If you need to repeatedly re-copy the same files, use the AltX save option to exit from DBE, which will save the tag states. Then, whenever you run DBE you won't need to reset the tags because you can just exit and pop back in again to set the file states to the values you saved.

F4 appears in two positions in the table because it either scans the current directory (you may edit the filespec in this case), or if there are files in the Selected (♦) list it scans the current level and its sublevels in search of files matching the names of those on the Selected list. ªF4 also performs a Selected scan, but also trims empty or unused levels from the list.

DBE v4.40 introduced removal of 'stub' directories from the memory list when performing a disk scan if no files or subdirectories have been found so far during previous scans. The memory recovered is immediately reused instead of allocating memory for new items. The ªF4 and ªF6 keys initiate scans which trim unused levels from the list.

■ IGNORING PATHS TO CONSERVE MEMORY

If you have an IGNORE declaration active, DBE will not search in a folder set to be ignored (and even if you use a direct ; or ' scan it will refuse to scan the folder) if a *.* wildcard search pattern is in use. If you set an IGNOREd level as the current, and then use F5 and a pattern other than *.* DBE will scan it. But the F6 scan won't read it whether wildcards are used or not, and the ªF6 scan, which trims empty levels from the list, will remove it from the list (and other scans such as RS may put it back on the list, although empty).

If you have IGNOREd a folder and want to 'drill down' into its subfolders to find a file anyway, you can use the SelectLevel option. Press \ to open SelectLevel, move the selection bar to the IGNOREd directory closest to the root and containing the folder(s) you want access to, and press ^\ followed by ' to read that folder's immediate subtree. Press \ to open SelectLevel again, select one of that directory's subfolders, and press ^\ and ' again. Keep doing this until you have reached the level you want a file list from, and there, press F5 if you want to set the search spec or ; if you want to use whatever is current. By this method, you can add particular files to the list as you see fit, whether you want to keep ignoring the entire subtree in general or not. Although scanning a folder whose IGNORE status you have defeated with ^\ re-enables the IGNORE flag for each of its immediate subfolders, that scan won't reach subfolders of those whose IGNORE status has been re-enabled and thus won't reset THEIR IGNORE status, because the scan doesn't read them until it has finished reading the immediate subfolders (and re-enabling their IGNOREs); and by the time it gets around to reading their files and subfolders, they're set to be ignored again. If you want to re-enable IGNORE flags, start at the tip of a folder tree and work your way back toward the root, using \ to call SelectLevel, moving up the tree one level, pressing ^\ to re-enable the IGNORE flag, and ' to exit and read the subfolder list, resetting the the flags of subfolders. HOWEVER! Once you've drilled down and added files to the list, you can operate on them using any of DBE's functions just as you would with any other files on the list. And you can copy files to those IGNOREd folders or subfolders, and create subdirectories, whether they're set to be IGNOREd or not.

The purpose behind this seeming complexity is reduction of memory usage if at all possible. Using AltF6 trims the list; using IGNORE directives keeps directories and files off the list if you don't deliberately look for them;

and using F6 doesn't trim the level list you have but also doesn't add items from IGNOREd directories.

Using IGNORE flags can enable you to divide up the work when it comes to managing large hard disks, so that DBE has enough memory to work with for handling portions of the drive limited to about 3000 files. If you save multiple DBCONFIG.TXT files, each with its own set of IGNOREd levels, it can prevent you from running out of memory all of the time when you try to get something done; and all you need to do is to start DBE from a different folder containing a particular DBCONFIG.TXT to make use of its IGNOREs. If you've started DBE this way and find you need to defeat some ignores several times in a row to get what you want done, you can defeat the flags, exit from DBE with ^X to save a DBSYSCON.FIG for that specific reason (under whatever name you choose), and then when you restart DBE the flags will be defeated as you set them because DBE reads the .FIG file at startup rather than the local directory.

■ DISK SCAN FILE SPECIFICATION

F4..F6 set the default file specification to *.*, and F5 and F6 allow you to edit it before scanning for files. This global spec is shown at the top of the screen if you don't have ScrollLock on. The last spec used to scan a particular directory is also shown in the file window if it is empty.

The : and ; keys use the default spec (unless you use them to exit the Select Disk window; in which case they use *.* as do ^R and ªR).

F7..F8, ^R and ªR always use the default spec. This can be changed with the F4, F5, F6, ªF5, ªF6, RC, RF, and RS options.

The ^A..^M specification is governed by the disk size; it either looks for all files on the disk or just reads the tree.

F4 in addition uses the names of files on the selected list, if any, instead of *.* when it scans, and can search an entire drive for matching files.

This is perfect if you have a floppy and want to find out where the files came from on your hard disk; just read the floppy, use the S option to select them, use \ to select the volume label of your C: drive in the Select Disk window and exit using \ or ◄┘ and then press F4. DBE will scan the tree and make file entries for only matching files found. Then you can place the selection bar on any file on the floppy and press N to jump to any others matching its name; or to view them all at once in the same window, use the SG option to select globally. You can also use the ªA associate option to link them into separate chains by file name and then press / to toggle viewing of associated files in the opposite window on and off.

IF CapsLock is on, 'SEARCH OFF' will appear, or 'SEARCH AnyCase' with the active search string below it if a case-insensitive search is on, to the right of the scan status information. The string search is effective only for one scan operation, so you need to either press AltF and edit the search string or press AltI to set or clear the IgnoreCase flag to re-activate SearchMode. When this is on, the search string appears on line one following the 'RScan:' file specification.

■ SELECTION - Selected (♦) List

Each time a Select operation is performed using the S key, the previous Select links are cleared, and at the end of the search for entries to be added to the new Select list, the number of files matching the Select criteria is displayed, and you may then press * to toggle the display of the Selected files on and off within the currently active window. Moving the selection bar or pointer up or down through the list allows you to indicate

which individual files you wish to tag or untag using the tagging keys, and the current disk identifier and directory name where the file resides on disk is shown below the window.

Since the Selected files can be from any disk or directory, viewing the file information in a single window is excellent for comparing file names, dates, and sizes; but, since the files are shown in a single list outside of their natural sequence in the directories in which they reside on one or more disks, it may be mnemonically useful-- so as to keep in mind where everything is at during an extended select-and-tag process --to press Enter to switch between the display of the Select list and the display of the Level to which a Selected file belongs. Navigating from directory to directory following the chain of Selected files is possible by pressing * to display the Selected list, ▼ or ▲ to place the bar on a Selected file, and then Enter to view its 'home' Level. The Selected list can be sorted by name, extn, size, and time just as the normal file lists are.

The Enter key also switches between the normal and Queued file views in the same way as between the normal and Selected list views. Note that if a file on the Selected list is also Queued, pressing Enter goes to the Queue View mode, and the Selected list is dispensed with. (Enter then switches between the Queue List and the normal view.)

The members of the Selected list are indicated by a green ♦ in the position of the period in the file name until they have targets assigned to them. In any case, the ◄] pointer turns bright green when it points to a selected file.

When one of these files is current, you may press Enter to toggle viewing it either in its position in the Selected list or in its position in its home directory.

When the Selected list is in view, the tagging option keys are the same as for a normal window Level display. Single files may be toggled on and off the Selected list by pressing the '-' key.

If you press the _ key the current file becomes the index point for insertion of Selected files into its directory; all files inserted come from that file's directory, and they are inserted in the order in which they appear in the Selected list. This is a quick way of arranging files in a directory list, because you can add them to the Selected list using '-' in the order you want, and use '_' to insert them wherever you want in the directory they came from.

If you wish to add to the Selected list (without clearing existing links) you press ^S instead of S to begin.

The Queue uses the Selected list view mode for display, and you should turn Queue View off when you select files or want to view the Selected List (which is normally cleared before Queue View mode begins).

■ TAGGING ENTRIES / KEY SEQUENCES

Five types of Entry tags are used: COMP, COPY, MOVE, REPL, and DEL. The Ins key toggles the COPY tag on and off; the Delete key toggles the DEL tag; and these two keys pressed in succession will allow cycling through the COPY, MOVE and DEL tags. Thus, if the file is tagged COPY and you press Delete, the MOVE tag will be set. If you press Delete while the MOVE tag is set, the DEL tag will be set. The following examples show how the tag settings change during several different sequences of keypresses:

.... >>Ins>> COPY >>Ins>> ....

```
....  >>Del>>  DEL   >>Del>>  ....
....  >>Ins>>  COPY  >>Del>>  MOVE
....  >>Del>>  DEL   >>Ins>>  MOVE
....  >>Ins>>  COPY  >>Del>>  MOVE  >>Ins>>  COPY
....  >>Ins>>  COPY  >>Del>>  MOVE  >>Del>>  DEL
....  >>Del>>  DEL   >>Ins>>  MOVE  >>Del>>  DEL
....  >>Del>>  DEL   >>Ins>>  MOVE  >>Ins>>  COPY
....  >>Ins>>  COPY  >>Del>>  MOVE  >>Del>>  DEL   >>Del>>  ....
....  >>Del>>  DEL   >>Ins>>  MOVE  >>Ins>>  COPY  >>Ins>>  ....
....  >>Ins>>  COPY  >>Del>>  MOVE  >>Ins>>  COPY  >>Ins>>  ....
....  >>Del>>  DEL   >>Ins>>  MOVE  >>Del>>  DEL   >>Del>>  ....
....  >>Ins>>  COPY  >>Del>>  MOVE  >>Ins>>  COPY  >>Del>>  MOVE
....  >>Del>>  DEL   >>Ins>>  MOVE  >>Del>>  DEL   >>Ins>>  MOVE
```

The last of these two keys pressed becomes a basis for the operation of the Plus key operation. The Plus key sets the tag, and advances the selection bar and pointer. Be aware that pressing Del sets the tag basis to DEL, and that the Plus key will continue to tag files with that state until you press the Ins key again. The Minus key clears any tag and advances the selection bar.

Note that the tag keys work differently when the SelectDisk operation is being carried out. While SelectDisk is active, the Ins, Del, Plus, and Minus keys act on entire directories, and they do not toggle the tags. Ins, Del, and Plus set the COPY, DEL, and MOVE tags, and Minus clears them.

Version 3.0 does not implement the VIEW and COMP tags.

The operation of the Ins, Plus, and Minus keypad keys in the normal file list window depends on the state of the <SLTOI>, <SLTOP>, and <CTOM> flags, which are discussed under the SETTING TARGETS heading.

The COMP tag is toggled with the ᵃC key. With this tag set, the source file is compared to the target file, and a green arrow, ▶, is placed to the right of the source file attribute area if the files are identical. If you tag files with this tag and issue a GO with greater than file scope, the files not found at the target location(s) will be skipped over and will still be COMP tagged; the rest will be verified and green arrows placed to indicate those for which the verification is successful.

Press the V key to verify single files. If you wish verification in addition to the normal copy/move processing, start the GO loop using the ᵃV key and the checks will be performed automatically (although the green arrow is not used to indicate success).

When (♦) appears on the top line, Queue Enable is on, and toggling a tag on or off also toggles the Queue status of the file on or off.

The REPLace tag set using the 'X' key does not affect the normal GO and PROCESS operations. The -><- tags are ignored except by the AltG Replace Mode processing, and only when source links are active for the tagged files. This feature is intended for use in repeated updates of certain files, and isn't of much use if you don't save a FIG file with the links and -><- tags stored in it for the files to be replaced and the source files they were copied to their present locations from.

To use ReplaceMode effectively, you should keep this in mind, and make sure you set the -><- tags as soon as you can get to it after completing the copy which sets up the active source link, and then save the FIG file.

■ LEVELS - SelectDisk OPTION

Selecting a particular level is a keypad-oriented feature; the keys used when the Level select feature is in use are all available on the keypad. Pressing the 5 key (Shift&5 if NumLock is off) begins the SelectDisk option loop. Two

windows are shown; one of them contains DiskID names and Level names, and the other may show the current file display for the current Level. The current Level is indicated by a selection bar in one window, and you may select it as the current Level for the active window by pressing Enter or BackSlash, and you may also cause a disk scan to be activated for it if you press CtrlR or AltR instead. CtrlR reads only the current level, and AltR reads sublevels as well. If you have enabled LongFilenames, pressing '?' or turning Scroll Lock on displays these below the list window.

When you set a Target, the S suboption calls SelectDisk to allow you to select a target Level using this option.

The cursor keys ▲ and ▼ only move the selection bar, whereas the ► and ◄ keys also display the file list for each Level as the bar moves. The Home and End keys move to the first and last Disk/Level. The keypad ▲ (↑8) and ▼ (↑2) keys do not operate in the same manner as the other ▲ and ▼ keys.

The keypad ↑1, ↑2, and ↑3 keys set the number of files displayed per line for the current Level. The ↑5 key toggles SelectDisk on and off. The ↑7 and ↑9 and the F7 and F8 keys initiate a disk scan for the current level in the left and right windows, respectively. The ↑8 key starts a disk scan for the active window, and allows editing of the scan specification, which defaults to the name of the current level, plus wildcards.

Entire Levels are tagged when this option is active. The tags are turned on by the Insert (COPY), Delete (DEL), AltC (COMP), and Plus (MOVE) keys, and turned off by the Minus (UNTAG) key. Note that using this option to tag or untag files will alter any tags of files already set in the directory in question.

Some of the text keys perform functions as well. The ↑4 and ↑6 keys work the same way as the ◄ and ► keys. The < and > keys move half the height of the window up or down. The , and . keys move up and down one level.

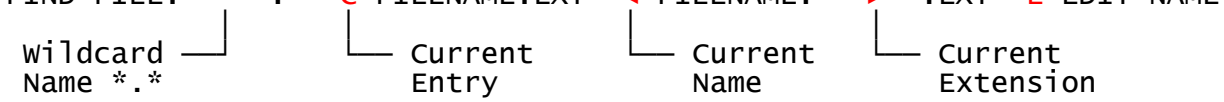
The Tab key cycles between levels which have been defined as targets. This is handy when the processing loop prompts you for a target and you choose the S option to select the level using SelectDisk.

If you type in a few alphabetic characters and then press Space, a search is made for a level name matching what you type in, and you press / to look for the next instance. The complete path isn't used for this; the disk name and backslash characters are ignored, and only the 12-character DOS filename for the directory is scanned for a match.

■ FIND | AGAIN

The F option locates files in memory, and sets the current Level and Entry to the file found, so that the selection bar in the current window appears on that file.

When you press 'F' the following prompt appears, with a box below which will contain up to about 20 previous find items:



To find any file by name, press E and edit the 12-character file name to be found. The search will then commence, and pressing A continues the search for more files by that name. If you keep pressing 'A', you'll come back around to the same file you started from. (If you use *.* as the search spec, it may take a while, because you'll go through all the files in the memory list.) Since * causes DBE to find any file at all, it isn't of much use unless you're in an empty level and want to find any file entry at all

in any level. But if you're using ^◄ and ^► to navigate through a run of empty levels and get tired of it, you can use the Find * option if you don't want to press Space and use the MRU list to return to the last location where a valid file entry existed.

The search box numbers the entries for which you use the E suboption to edit or enter a filename. When selecting the entry by its number, you may need to press Enter (or Space) if selecting one of the first two entries and the list contains more than 9 entries. If you press = or 0 the find option will exit and set the find spec to the name of the current file.

You may also move the pointer using ▲ and ▼ and press Enter to select the search pattern. The item selected is placed at the head of the list, and the others move down. Use Del to delete a list entry. The least recently searched-for item is removed from the list if the total list length goes above about 20 entries and you add another. To edit one of the list box entries, move the pointer to it and press Space.

■ THE TYPE-IN FINDER

Press one of the Shift keys along with Space to start the type-in finder, which allows you to use ▲ and ▼ to restart or continue down the list, or press \ to choose a new directory to look through for files. What you type in is compared to the leading characters in the filename, and if there is a match, the selection bar moves to that file. If you want to return to where you were before starting the finder, press Esc to turn it off. Note that when you're finding, DBE doesn't do anything else, and you have to press Space, Enter, or Esc (or type in more than 12 characters) to resume normal operations. The finder doesn't support wildcard ? and * characters.

■ DISK SCAN & SELECTION WITH FIND

Both disk scan and selection can use a search string as a criterion for qualifying a file entry. If an entry qualifies, the disk scan adds an entry to the memory list for the file, and the selection routine adds the entry to the Selected list (provided other qualifiers are satisfied if you enabled the test for them).

For the string search within files on disk, press AltF in the main file display loop, and for the Selected list search (within disk files for which entries are already on the memory list) press 'F' while in the Select routine choosing the selection qualifiers. In both cases, a prompt is issued at which you can enter the string to be found; and case sensitivity is prompted for when you have entered the search string.

AltI can be pressed to toggle case sensitivity on/off; and in the main loop when both the search string and the file data are to be uppercased before comparison, an upward-pointing arrow will be shown next to the search string on line one of the display.

If CapsLock is ON when a disk scan starts, the scan flag statuses are shown, and the search string will also appear if one is active.

At the main loop, you can press AltI to enable another scan with the same search string without having to re-enter it, toggling the uppercase flag as well. Pressing AltF when the search string is active turns the search off, and another press allows you to edit the string and activates the search.

■ NEXT

Place the selection bar on a filename and press N to find the next file by

that name anywhere on the list in memory. Remember that this does not find files on disk; the corresponding director(y/ies) on the disk must be read first.

During a file Associate, the Enter key normally jumps to the next file linked to the current one, but if the file is also on the Selected list, you need to press N instead of Enter (although these do not necessarily traverse the ring of linked files in the same direction), because Enter always hops to or from the Selected list (or the Queue) if the current file is on that list.

■ JUMP | MARKERS | TARGETS | CYCLES

Several jump options exist. Pressing J and then H sets a marker at a file, and you can then jump to it with JN or JP; or you can press 0 to view the list of nine JH jump points you have set. Pressing 1 through 9 will take you to the corresponding jump point (you can only set 9 different ones with the JUMP HERE option, but you can replace previous ones by pressing = and then the number of the jump point to be replaced).

If you want to jump without changing the current window entry, press AltJ to execute the jump in the opposite window. AltJ T would show the current target (if there is one defined) for the current file, but in the opposite pane.

With DBE Ver. 4.40, the ShiftTab key no longer adds new jump points to a cyclic ring of entries. It returns to the first valid item added to the Entry list at startup.

The neXt and preV jump options (press JX or JV) are for Target directories you have defined using the 'T' option. If you press J and then T you will go to the current target, and if you press JD (D for Destination) the current file's destination directory will be displayed. You don't need to have the destination directory displayed in the other window in order to copy to it, but you do have to select it deliberately. In any event, if you need to pop over to it for a look around, JD takes you to it.

If you happen to be in an empty directory, which may happen often enough if you are jumping to unread target directories, you can press JR for JUMP RETURN to jump to the last valid file which was current, or use ShiftTab to return to the last saved point. If you press Space, a box pops up with a list of recent locations and you can press F1..F9 to select one to jump to. (Space doesn't do this if there are files in the current level; if that is the case, it centers the current file vertically in the window.)

If you don't have v4.2+, the jump points aren't saved when you exit DBE. Newer versions save these if you save a DBSYSCON.FIG file when you exit DBE.

Queued files may be jumped to by holding down the Alt key and pressing the number keys above the keyboard to enter the processing number of the file you want to jump to. (With ScrollLock on and the QE (Queue Enabled) mode on, these numbers will be displayed next to the file tag area.)

■ IMMEDIATE COPY|MOVE|DEL|COMP

Immediate copying, moving, or deletion of the file which is currently selected can be initiated by the C, M, and D keys without having to tag the file first. If a copy target is already set, DBE will go ahead and copy the file to the assigned target; if not, you will be prompted to set a destination for the file. (DBE v4.2+ prompts in any case.) Press the V key to COMPare (verify) the file with a target you select.

Files can be tagged individually, each in turn, using the Ins and Del keys (press both to tag a file to be MOVEd), or groups of files in their entirety

by using the S option to Select files or the ↑5, \, or ^F4 SelectDisk option to access the list of Levels and then tagging entire Levels with the Plus (MOVE), Ins (COPY), Delete (DEL), or Minus (untag) keys.

The destination directories have to be set explicitly for all files to be copied or moved. If you define a global target before tagging any files, then the T target option T suboption will refer to the global target. You can set the range of definition for a target, so that it applies to a file, directory, subdirectories, selected files, etc.; and files which are within the range of definition will have their targets set to the value you have defined.

■ SCOPE

When a disk scan is made, the scope is prompted for if you use the R read option. The effective scope of disk scans is discussed under the DISK SCAN RANGES heading. The scope of list operations, including searching, tagging, Selecting, and Associating of Entries is discussed here.

```
  ■ FILE SELECT RANGE SELECTION ■
  SCOPE:  File  ^Selected  Path  Disk  Subs  Window  Volume  Global
```

The prompt to select the scope of a given operation on the Level list is as above. The meaning of each type of scope is as follows:

File Scope restricts the operation to the currently selected Entry which is indicated by the position of the selection bar.
Selected Scope restricts the operation to Entries which are currently linked into the Selected list.
Path Scope, which could be called Level Scope or Dir Scope, restricts the operation to the file Entries which belong to a particular Level.
Disk Scope restricts the operation to Entries scanned from a unique disk, regardless of the Level(s) in which they exist.
Subs Scope restricts the operation to the Entries belonging to a particular Level (directory) and to its sub-Levels (subdirectories);
Window Scope restricts the operation to Entries belonging to the Level currently being displayed in the active window; this is the Path Scope for the active window.
Volume Scope restricts the operation to Entries scanned by a particular disk drive (only the drive letter is significant).
Global Scope includes all Entries in the memory list.

The C, M, and D options operate exclusively on the currently selected Entry independently of any previously tagged, Selected, or Target-defined Entries. The scope of these options is thus File Scope. The C and M options tag the current Entry and immediately issue a prompt to allow you to select the Target (destination where the actual file is to be copied), and then proceed immediately with the Copy or Move.

The P option (Process) and the G option (Go) both operate implicitly on the set of all currently tagged Entries; however, they act explicitly in radically different ways. The goal of both of these operations is, implicitly, to see that, eventually, all tagged files from all disks scanned have been copied or moved to their defined Targets, and all deletions necessary have been made. Each Copy or Move requires that a Target be defined. Targets may be defined before the Go or Process command is given, or they may be defined individually for each file in turn as it is processed.

The Process option operates on one Entry only and then returns to the main command loop, updating the active window to show the selection bar on the next Entry to be processed when the P key is pressed again. This is equivalent to stepping through the Go loop, one Entry at a time.

If the current file is not enabled for processing, pressing P jumps to the next file eligible for processing.

The Go option prompts you to select the processing range from the eight types of scope discussed above. Once you set the scope over which file operations are to be performed, the Go loop traverses the Level list, processing tagged files which are within the scope you selected until you cancel or suspend processing or all Entries within that scope have been processed. If you have to skip any for some reason, pressing G again restarts the GO loop, and if a file is not set to be Ignored and has not been copied, it will be tried again.

Note that QS toggles Queue Start/Stop modes, and that after you start the Queue, both Go and Process follow these process numbers in order, until you stop the Queue again.

■ SELECTION - Scope and Criteria

Creation of a Selected list involves three primary criteria: The scope over which candidates for Selection are evaluated, the file name specification (which may involve wild-card-characters), and a Selector (which is actually a set of characteristics defining additional criteria which you wish to be satisfied by each Selected file). Given the interpretation of scope as discussed above, it is obvious that Entries outside the Selection scope will not be candidates for Selection. Within that scope, Entries having names which do not match the file name specification will not qualify, either.

The Select options provide five methods of choosing the file name specification. The C option sets the spec to the name of the currently selected Entry. E allows editing of the spec directly, and the ◄ and ► keys parse the current spec, substituting wildcards for name or extension portions of the spec. The * option sets the spec to '*.*' and places <AllSel> in the Selector set, removing <NameSel> from that set; whereas the others remove <AllSel> and add <NameSel>.

The file attributes can also be evaluated according to four criteria. This involves a test to determine the state of one or more of the following attributes: ReadOnly, Hidden, System, and Archive. You use the uppercase R,H,S, or A to enable the attribute test for the particular attribute, and lowercase r,h,s, or a to disable it. <AttrSel> becomes part of the Selector set unless you press N to negate the test, in which case, <NotAttrSel> takes its place. The effect of <NotAttrSel> is to exclude an Entry having any one of the enabled attributes, and the effect of <AttrSel> is to include an Entry having all of the enabled attributes.

The file date (including the time of day) can be compared to a given date if the D option is used to add <DateSel> to the Selector set. While setting the date selection criteria, if ◄ is pressed, <PreSel> is added to the Selector set, and then only files created previous to or on that date will qualify. The ► key adds <PostSel> to the Selector set, qualifying only files created on or subsequent to that date.

The file size can also be used as an additional qualifying factor in a manner analogous to the date qualification. Using this option, <SizeSel>, and, optionally, <SmSel> or <BgSel> can be made part of the Selector set.

The current tag value of the file can also be used as a qualifying factor. This can restrict the selection to those files which are not tagged, to those which have some tag set, or to those which have a particular tag set.

The above criteria determine whether an Entry qualifies during the Selection search. If the Entry qualifies, it may be tagged for a Copy, Move, or Delete operation, or it may be untagged, depending on whether a tag operation is enabled.

The main loop S and U options are used to initiate the Select and Unselect searches. The Unselect search removes all qualifying Entries from the Selected list. The Select search adds Entries to the Selected list. These

searches tag files only if <CPYTAG>, <MOVTAG>, <DELTAG>, or <COMPTAG> have been added to the SetTags selector set. The SET TAG value as well as the tag selection qualifier are shown when you use the S and U options, provided you have used the T suboption to either enable tag setting/clearing or to add a tag qualifier to the qualifying selector set.

Tagging of files in the Selected list can be done individually by displaying the Selected list and using the Ins, Del, Plus, AltC, TextB, TextX, and keypad Minus keys to set a tag for each file. If you enable setting/clearing tags when defining the criteria for selection, all qualifying files will be tagged as such (or cleared) when they are added to or removed from the Selected list. You may turn the tags off if you choose the SET TAGS OFF option.

The S option replaces the Selected list each time you select files. **If you want to keep the files aleady selected on the selected list**, use ^S to start the Select option. The U unselect option acts only on the Selected list, and you can selectively remove files from the list.

The qualifying selector can include files based on their tag state if you press ^O, ^V, ^C, ^M, ^D, T, or U to qualify the search to include only files tagged for COPY, MOVE, or DEL, only untagged files, or only tagged files:

```
SELECTOR: A ALL ^C COPY ^M MOVE ^D DELETE ^V DIFF  T TAGGED U UNTAGGED ^O ■OK■
SET TAGS: O OFF  C       M       D         V VERIFY
```

Note that the top row lists the tag states which can be used as qualifying selection criteria (■OK■ refers to COPIED files, DIFF refers to files which have been compared and differ from the target to which it was compared). The bottom row lists the values which the tag states of the files which qualify for selection can be changed to.

■ DiskID | SERIAL # | VOLUME LABEL

The disk identification criteria (either a disk volume label or serial number or the IDxxxxxx.DSK identifier file) are foremost in importance. If you use a disk which has no such identifier, DBE may not carry out the operations you intend to have carried out, and it may refuse to write files to the disk or to allow you to continue by replacing it with another disk.

If you don't mind DBE writing disk identifiers onto your disks, you can leave them un-write-protected and unlabeled. DBX 3.0 doesn't write identifiers on disks which have serial numbers or volume labels, but with 4.0 you press OS to write a DiskID file to a disk if DBE doesn't do so automatically.

The DiskID files occupy zero bytes of disk space (they are merely a directory entry in the root dir) and are marked with read-only, hidden, and system attributes, which means that for the most part they are ignored. Some people (perhaps most people- who knows?) are too lazy to label their disks. That's fine; that is why DBE writes identifiers when you haven't taken the trouble to. The method used for assigning them is expected to be able to assign unique identifiers to disks until the year 2044. By then, people may have forgotten what disks are.

With DBE versions 4.2 and above, the 6-character identifier is shown next to the volume label when you use the \ option to select a disk or level.

■ GO/PROCESS FILE OPERATIONS

When you have assigned a target to a file and press G or P to start the copy or move, DBE checks the target drive for a disk, and if it is the wrong disk (not the one which the assigned target applies to), you will be offered several options, as described below. If the disk in the drive currently

assigned to the source disk is not the correct disk, the following message appears:

```
SOURCE DISK WAS NOT LOCATED IN DRIVE A:
TARGET DISK IS LOCATED IN DRIVE F:
>>> YOUR OPTIONS ARE:
C CONTINUE WITH NEXT FILE                    B BUFFER DBXI.BMP
Q QUIT                                      ^B BUFFER ALL IN SCOPE
E EXCHANGE SOURCE AND TARGET DRIVES
R RETRY AFTER INSERTING A NEW DISK IN DRIVE
S CHANGE DRIVE ASSIGNED TO SOURCE DISK

DRIVE A:  <SOURCE>  DISK: "FWAWRW" -  SHOULD BE DISK "FUH1S7"
DRIVE F:  <TARGET>  DISK: "FTQ47M" -  OK

 SOURCE: A:\DBXI.BMP
 TARGET: F:\GM\TEST\DBXI.BMP
```

If the target disk is the wrong one, the option

```
 T CHANGE DRIVE ASSIGNED TO TARGET DISK
```

will be offered as well, and the ID for the disk it should be will be shown. You are given the opportunity to press a letter RQCTSE to choose the option, and to try again if you don't press C or Esc. If you have DBE v4.40 12-04-10 or newer, the 'PRESS DRIVE LETTER' prompt has been replaced with a new option which allows you to check all available drives so that you don't have to take a guess at a drive letter which Windows may have reassigned. See the DRIVES option for details.

With v4.1+ you can press B to buffer the current file from the source disk, or ^B to buffer all files to be copied or moved.

When an error occurs copying to a target disk, such as not enough free space, a write-protect, or a disk write error, you may have several options; among them are:

```
NO MORE FREE DISK SPACE IS AVAILABLE ON DRIVE x:
NOT ENOUGH SPACE IS AVAILABLE ON DRIVE x:

YOUR OPTIONS ARE:

F TRY TO FILL OUT THE REMAINING SPACE WITH OTHER FILES
A BEGIN COPYING TO ANOTHER DISK
W WRITE THE REMAINING FILENAMES TO THE OVERFLOW LIST AND QUIT
```

If you press A, the message

```
 INSERT ANOTHER DISK IN DRIVE x:
 THEN PRESS ANY KEY TO CONTINUE.
```

appears. Version 3.0 doesn't write an overflow list (a previous backup program on which it is based did, but the overflow list is built in; not an external file, for DBX).

Other problems, such as a write protect, generate the following message:

```
 >>> YOUR OPTIONS ARE:

 S TRY ANOTHER <Src/Targ> DISK
 N PROCESS NEXT FILE
 Q QUIT / UNTAG ALL TAGGED FILES
 R REMOVE WRITE PROTECT AND RETRY
```

Assuming you were lucky enough to have had the correct disks in the drives, or found the right ones when they were prompted for, the source and target filenames would be displayed at the bottom of the screen before the copy begins. Then, if some error occurs, or if the file already exists on the target, a message similar to the following would appear:

```
THE SOURCE AND TARGET FILES HAVE THE SAME NAME.
YOUR OPTIONS ARE:

R REPLACE EXISTING TARGET      F COPY SOURCE TO NEW FILENAME
A REPLACE ALL IN SCOPE         D CREATE NEW DIRECTORY
S SUSPEND FILE PROCESSING      N COPY NEXT FILE
B BUFFER FILE DBXIC.BMP       ªB BUFFER ALL IN SCOPE
V VIEW FILES

 SOURCE: 04-01-08 11:19:22          4054        ┌┐ PRESS ? TO COMPARE FILES
 TARGET: 04-01-08 11:19:22          4054        └┘

 SOURCE: A:\DBXIC.BMP
 TARGET: A:\DBXIC.BMP
```

If you decide to copy the next file, a message appears:

```
 REMOVE THE SOURCE FILE FROM THE LIST OF FILES TO BE PROCESSED ?  (Y/N)
```

and you can then cause the file to be Ignored during further processing by pressing Y. This Ignore condition for a file can be toggled on/off with the I key in the main loop.

If DBE doesn't automatically compare the source and target and put up a message saying FILES ARE DIFFERENT or FILES ARE THE SAME, you will be offered the option to press the ? key to run a compare before proceeding, and it will put up the message once you do that. (Files which are not the same size will not be compared.) For versions 4.2/4.3, the compare flag may not be reset when a GO starts (or a PROCESS for a single file), so you may want to check manually by pressing '?' when copying single files or starting/restarting GO.

DBE versions prior to v4.40 12-14-10 do not explicitly provide for a file rename. You could, however, MOVE tag the file, setting the target F (for Filename) and then entering the new name. This allows renaming across drives, which is normally impossible. File moving is sometimes done simply to place a file in a new physical location on a disk, and can help to defragment files when there is enough free contiguous space to put the moved file into (although you wouldn't rename the file if you're only trying to de-frag it).

DBE version 3.0 doesn't explicitly provide for directory creation (no built-in MKDir command). But it will create directories under certain conditions as part of a file buffering procedure, and new directories can be created using the DefineTarget procedure.

When a copy is completed, if the target directory is visible in the opposite window, the new or updated files will appear in a different color than the files already present. If you use ; or RC to re-read the disk, it will update only filenames not previously shown or ones deleted from the memory list with the Bksp or ^Bksp options. If you use ^R or ªR to rescan the directory, the file status will be reset for the new/updated files and they

will appear in their normal colors. If you have run out of scan memory, it is still possible for DBE to add new file entries to the list, but it won't be much more than about 200 of them. If new files suddenly stop appearing in the opposite window while copying is proceeding, you've run out of memory. If you press F1, the help summary screen won't show up; and that only happens when you get down to less than 4000 bytes free. Note that the MEM LOW: 0 notice will appear on line 1 of the display while you still have enough memory for about 200 new file entries; but after that shows up you should make sure FIRST to add all of the target directories to the list that you are going to need for <ToDir> targets because without those, you can't define those targets. If worst comes to worst, you can always cut directories you're not working with from the list, and DBE will reuse the memory for the new directories you add to it. But, as noted elsewhere, cutting files from the list won't make memory available for new directory entries on the list; only for file entries.

The <Replace> or <Copied> status indicators are changed to <FileOK> by the RC key sequence, and the ';' key does not change this status. Thus, it can be used to scan levels containing entries with active source links without untagging entries having REPlace tags set. Using RC means that you have to manually retag all <Replace> candidates by pressing 'X' for each one.

Once a file is copied, you may press JC to jump to it (or to the next copied file). This doesn't jump to the new file; it jumps to the source file. If you need to re-copy certain files, the < and > keys won't find them so you need to use the JUMP to COPIED option. The < and > keys only find files tagged and ready for processing. If you want to jump to the target file, place the bar on the source file and press JD for JUMP DEST; or in v4.1+ you press ↵ to hop between source and target files once the copy is complete.

Version 4.3N introduced source links for each file, which are set as soon as a copy is completed, and remain active until the file state is cleared. If a source link is active, pressing '/' to view Associated files will show any source files in bright cyan in the opposite window, and pressing 'X' to turn on the REPLACE tag (which looks like '->><-') shows the tag in bright green when the source link is still active.

Copied files need to be re-tagged (in version 3.0) before they can be copied again, and you need to set the destination again. This presumes you wouldn't need to keep copying a file repeatedly to the same location (if you do want to, you exit DBE with ªX to save the configuration file once the file is tagged and the destination is defined) but would be more likely to copy it somewhere else as well (and v. 3.0 doesn't support multiple copy destinations for single files). The target files, when they appear as new entries in different colors, can't be tagged, except for the REPLACE and BACKUP tags; and the file can be added to the queue, although not selected. This keeps you from accidentally copying, moving, or deleting them until the target directory is re-read or you signify that you want to queue, backup or replace them. If you really do need to do something with one of them, use Bksp to drop it from the list, reread using the ; key, and then tag the new entry when it appears in its normal colors. The rest of the new/update file entries will be unaffected by this if you use ; for the scan. (You could also use 'X' to toggle the '->><-' tag on and off if you don't need the file state reset to the as-read-from-disk condition.)

Enabling copying again in v4.1 can be done using the Options Re-Enable feature (for groups of files), and in v4.2+ also by pressing E for individual files.

If the Go Scope is SubsScope, you will be prompted whether to include the relative directory path of the source file (its subdirectory name relative to the point from which you initiate the Go) on the target disk.

The GoByTarget mode is started by pressing ªG, and this processes the list in the order in which you insert disks into the target drive to copy to them. To save swapping disks in and out of the drive, you should buffer the source files beforehand if they are on disks which cannot be placed in the drive because the target disk is occupying it.

■ DISK STATUS DISPLAY / OPTIONS

You press = to view the status screen, and PgDn to view the Disk list, from there. Space exits the status routine, and ◄ and ► place the selected disk's current directory in the left or right window. The status screen allows you to set the backup and buffer directory names, to turn AutoScan on/off, to turn on or off the preservation of the file time or archive status when copying and the automatic creation of directories on the target disk. You can also set the global Scope and SELECT Scope (although these are usually overridden by keys you press or choices you make along the way). If you decide you need a new directory on a disk, press D to set the DOS default directory; the directory you name will be created, if possible; and if so it will become the current DOS default directory. This is handy if you intend to save a configuration on exit from DBE but don't have anywhere handy at the moment to leave it. IF you exit with CtrlX instead of AltX, you can place the DBSYSCON.FIG file in either the current DOS default directory, the current target directory (which you can set here also), in the same directory with DBE.EXE, or in a location you select using the SelectDisk window.

You can press Plus, Minus, or Ins to set the SLTOP, CTOM, and STOI flags here. Press B to set the backup drive/directory, and you can choose to write the source files to current date/time directories (The <StartTime> is updated every time the date has changed when DBE starts, and the <GoTime> is updated when you initiate a Go the first time after starting DBE) on the target disk.

The summary shows the amount of fixed memory (allocated but not currently in use) available for each type of list element, as well as the total free unused memory available. The number of bytes in files scheduled for deletion or copy and the number of files are also shown. Freeing memory independently is not enabled in any version of DBE so far, and you need to exit from DBE, saving the configuration with AltX, and then start it again, to reclaim the memory as a single block.

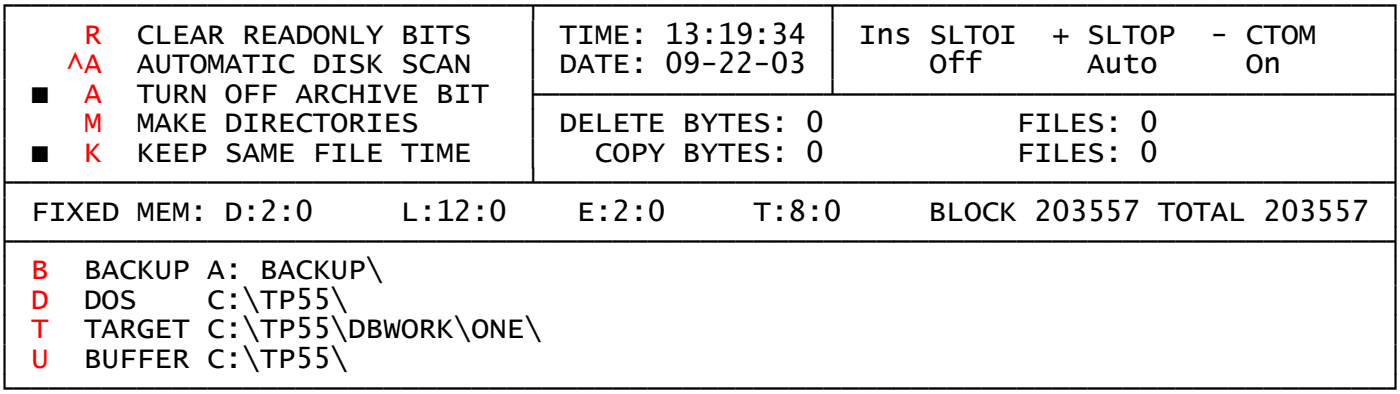
The F option allows you to adjust the fixed memory reserved for special purposes. This memory is only used after DBE runs out of both free memory and the fixed blocks which you have freed up by cutting items from the memory lists (using Bksp or ^Bksp), and in certain situations you will be prompted whether to use up the reserved memory. The reason for this approach is to keep the disk scan functions from entirely using up the available memory during a scan; and it allows you to still be able to define a few targets (which may require adding both new target and level entries at a time when all of the automatically available memory has been taken) provided you have reserved space for them.

The fixed memory is displayed in the form Z:RRR:XXX, where Z is the type of block, RRR is the number of reserved blocks set aside, and XXX is the number of blocks available because of dropping items from the lists. You can only change XXX by cutting items from the list. The RRR values can only be from 0 to 255, and they can only be changed by using the F option or in response to a prompt, issued when memory runs out, to which you answer 'Y' indicating that you wish to take the memory you need from the reserved blocks.

The disk free space, used space, and total, as provided by DOS, are shown. The values which Win98 and WinXP provide do not appear to make much sense at times for some floppies or large disk volumes, so you might ignore them, for most practical purposes. Version 3.0 does not have any method of calculating the correct cluster counts and sizes for disk volumes based on the DOS interrupt service routine INT 21h function 1Ch, which is the standard disk parameter provider for versions of DOS prior to the introduction of the VFAT and FAT32 disk formats. And there is, so far, no guarantee from Microsoft that documented functions will be provided to return the correct values in the future. There also appears to be a disk change line status problem which results in DOS not updating the disk parameter table for floppy drives

(although it will update the system file table and disk directory structures in its internal list) unless some function is called beforehand which makes a system service request that cannot be performed without updating this information (such as a ChDir command which sets the default first to one drive and then to another). I say it appears so; but this is based not on documentation but on observation; and it may not happen with all OEM versions of DOS or on all systems. But I almost expect it when using the new operating systems which more or less leave DOS out of the game.

Status values for the current file in the active window are also shown. The meanings of these values are explained elsewhere in this documentation.



IF the CORRECT used/total/free disk space values are returned by DOS, the total occupied and projected space usage (after copying files) for each disk are shown by a bar consisting of ▒ and ■ characters, and you can tell at a glance where you still have disk space on your disks.

The INT 21h Function 7803h interrupt service routine under Win98 is documented as providing a data block consisting of DWORD integer values for the large disks available under the FAT32 system. It provides the number of free and total sectors and clusters, both for compressed and physical volumes. However, a DOS program running in a window apparently is not returned the correct values when it calls this function, although when running in MS-DOS mode, the values supplied are correct. It beats the incorrect values returned by calling the INT 21h Functions 1Ch or 36h, but you still have to run the program in DOS mode to get the correct values.

So, at this point, I haven't decided whether to use the 21h/7803h code I have written as part of DBE's box of disk routines; and using the DOS or BIOS calls to get the DPB or BPB (parameter blocks) for the disk is still even more of a long shot, when it comes to compatibility issues. I would like to see better support for DOS programs when it comes to disk I/O, because without it, there are a number of built-in features DBE really cannot use reliably that have to do with error-handling (and giving the user alternatives on time, in time, and smoothly) and which I have had to switch off when the program is running on a system which provides it bad disk free space or capacity values. Knowing how much - EXACTLY how much - free space exists on a disk is critical to algorithms like the 'knapsack problem' where you develop ways of determining in what order and into which knapsack items of varying shapes and sizes will fit. The code for DBE's automatic 'space-filling' function which saves the trouble of having to examine the free space on half a dozen disks (or more) to find out where you can fit your

files is practically useless under an operating system that returns bogus free disk space or capacity values! I don't know whether Micro- Soft is trying to tell me not to bother, or what; but I thought it would be a nice feature to have in a backup extender program. And now I get to see what Win98 can do for it, and I find out there's nothing there that makes it any more powerful than it was under plain old DOS. In fact, DOS never lied to it about disk space values on a system that wasn't broken.

To sum this all up, all I can say is that you should try sticking two or three different disks in drive A: or B: or whatever, swapping them around and making DBE read them; noting in particular whether the disk sizes DBE displays are correct for them. And if your operating system doesn't return the correct values to DBE through the INT 21h functions 1Ch or 36h, the correct values won't be shown by DBE, period, no matter what you do. If that is the case, DBE may refuse to copy files onto a disk when you know there's enough room; but as I said, there's nothing anyone can do about it (the knapsack thing has nothing to do with this aspect of it). When the values are bad, the number of free and total clusters are usually the same, and DBE has an internal flag that gets set when free=total for a large disk (which almost HAS to have files on it if it's of any use at all). And DBE ignores the cluster size from then on, and rather than predicting when it will run out of space, it just goes on copying files until DOS tells it there's no more room. You can try it out and see how it works... You'll get a prompt to choose whether to try to fill out the space with other files (but how good that'll be under the circumstances is questionable because DBE has to actually try copying to determine whether one will fit) or to start with another disk, or quit. But it's by no means as appropriate as it would be to determine in advance what course of action to take (an impossibility with bad disk parameters, as I've said).

DBE displays disk free/used space either in terms of clusters or bytes. If clusters are the unit used, a § character (ASCII 21 - NAK) follows numbers displayed at the bottom of the screen. Normally, the free space is in green, the used space in red, and if there's room, the total is shown in yellow. The values INT 2F Function 7803h returns are used when DOS version 7.0 or newer is detected, whether they are correct or not. If the values for free and total clusters are the same it is assumed that either the drive is so large that checking for free space isn't really necessary or that the values returned by DOS aren't worth anything anyway. So it's possible DBE might balk at writing files, but the chances of this happening when the INT 2Fh values are used are minimized. If the free and total space as returned by either INT 2Fh / 7803h or INT 21h / 1Ch appear to be reasonable as byte values and the sum of the base two logarithms of the values returned is low enough that their product can be stored in a signed double-word and still be non-negative, the cluster values are used to calculate the bytes free/used/total; otherwise, you'll see the § symbol following the numbers of free/used/total clusters. Calculation of free space values when the lg2 is below 32 works well enough that DBE will let you know when you're running out of space for files.

There is a reported bug in the file CREATE sections of DOS where large files are concerned. What this amounts to is that, for some reason, if you don't write a single byte to the file first of all to set its current size to other than zero bytes, DOS apparently screws up trying to expand the file size if the number of bytes you want to write is large. So there is some measure of unreliability in DOS itself when it comes to large-item-handling. I have used DBE regularly to copy files of several megabytes in size, without any problems, under Win98. For instance, the swap file is over 98300 kilobytes in size, and DBX didn't seem to mind copying it. But under DOS 3.3 a file that size would span over three hard disks (limited to 32 megabytes each), and although DBE wouldn't have a problem with it, DOS would. (Don't MOVE your swap file; it's a fact that Windows wouldn't like that.)

DBE has been tested enough under Windows 2000 while running in a command prompt window to be able to determine that most of the time the values returned for floppy drives for the sizes is correct; whether it can be considered to be reliable enough for these purposes has not yet been determined.

■ DRIVES - REASSIGNMENT CHECKS

Press 'O' and '\' for the Options DRIVES suboption. This is activated when you choose to change a SOURCE or TARGET drive during file processing when DBE notifies you that a drive does not contain a disk and allows changing the drive letter assigned to it, which happens often enough when Windows is reassigning USB drive letters as you insert and remove drives from the USB slots. If you need to check any drive letters or add a disk to the list, or update one without altering the statuses of the current file entries on that disk in any way (in other words, to change only the drive letter which DBE is associating with that disk), this option is available in newer versions of DBE v4.40 beginning with the 12-04-10 build.

Drives 'A'..'Z' are listed above, and below are listed the DiskID and disk volume label for each disk in the DiskDriveRecord table. Gray entries are those for which no memory list entry exists but the drive, when last checked, contained a disk from which they were read. If an entry is in bright green and bright cyan, a memory list entry does exist in addition to that. Those entries which are dark cyan represent memory list entries which are recently active DDR table entries. The drive letters are shown in yellow after the corresponding disk has been checked for and found in the drive; and otherwise are shown in grey if ACTIVE or blue if UNUSED.

The ACTIVE and UNUSED statuses are only a matter of convenience, to keep the search limited to a few drives when the routine begins or when you use the * CHECK ALL suboption to verify all of the drives you have marked as ACTIVE to make sure the DDR table correctly lists the available disks for those drives. As you exchange USB drives or floppies, these will change; and using * is a way to quickly check the drive letter assignments for them.

If you need to LOCATE the current drive for a memory list entry which has been reassigned, you can press '\' to open the SelectLevel window, navigate to the disk you want to update the drive letter assignment for, and press '\'. This option searches ALL of the drives, whether marked as ACTIVE or not; and if a disk is in a drive it will find it and update both the DDR table and the disk entry drive letter stored in the memory list.

If you haven't scanned a disk and it thus has no memory list entry, LOCATE won't work for it; and you need to press the drive letter (this only adds a DDR table entry, but not a memory list disk entry) or to press Ctrl and the drive letter key (which is similar to the ^A..^Z disk scans but does not start a scan of the disk; it only adds an entry for that disk to the memory list).

Using the arrow keys to move the red triangle pointer above the list of drive letters changes the current DDR table index, and the Enter, Plus, Minus, and End keys initiate operations which are performed based on the value of this index. The Plus/Minus keys set an ACTIVE/UNUSED mark, the End key clears all ACTIVE marks above that index (except for verified DDR table entries), and the Enter key does a drive check but does not add memory list entries.

When this display appears during a file processing operation after you are notified that the wrong disk is in a drive, the name of the disk you need to find a drive letter for will be shown at the bottom right, after the question mark. If you press '?' the correct drive will be searched for, or if you can locate it yourself you can press the drive letter or press '\' to use the SelectLevel window to view the Level list and choose the disk you want to use. (Note that this doesn't change a <ToDir> target; only the drive letter.) Pressing Space to exit will then cause the drive letter in question to be updated to the letter which the red triangle is pointing to when you exit the DRIVES option back to the processing loop. You can see how this works by deliberately choosing the wrong drive. You'll wind up right back at the 'wrong disk' prompt and have a chance to do it over again or correct it. If you press Esc, the processing loop will stop, and you'll be back at the main files display with the selection bar resting on the current entry to be processed.

Note that DBE does not update the DDR table automatically. DBE makes no assumptions about which disks are to be copied from (or to) or in what order, and this depends on the sequence in which you start processing operations as well as their scope. The only record DBE has of what drive is associated with a particular disk is the last disk scan you performed, and if no reassignment of drive letters takes place in Windows, you won't need to change them in DBE's memory list.

■ FILE PROCESSING STATUS INDICATORS

If you want to know the current status of a file, press the ? key or turn on ScrollLock. The state of each of the file process flags will be displayed, as follows:

```
DBCONFIG.CFG TARGET: NoTarg    STATUS: <FileOK>
♦Process  ♦Backup ♦DestDef ♦Found  ♦Written ♦Ignore  ♦Copied  ♦Tried
♦BEnabled ♦SrcRdy ♦DisBuf  ♦FinBuf ♦TargRdy ♦DisTarg ♦ScrCdel ♦BufCdel
```

If the flag is on, the flag name and the adjacent ♦ character will appear in a bright color, which indicates that the condition to which the flag applies is TRUE. The flags are as follows:

BEnabled	File buffering in BufferDir is enabled for the file.
FinBuf	The file has been copied to the buffer.
TargRdy	The target disk is ready in the destination drive.
DisTarg	The destination for the file copy is the target disk.
SrcRdy	The source (or buffer) disk is ready in the source (buffer) drive.
DisBuf	The destination for the file copy is the buffer directory.
BufCdel	The buffered file has been deleted.
SrcCdel	The source file has been deleted after being moved, either to the buffer or to the destination.
Process	COPY/MOVE operations on the file are enabled.
Backup	The file is to be copied to the backup drive.
DestDef	A target directory or disk has been assigned to the file.
Found	The source file was found when processing was initiated.
Written	The file has been written to the destination (buffer or target).
Ignore	The file operation was canceled and the file set to be ignored.
Copied	The file has been copied to the assigned target directory.
Tried	Some operation (even a null one) was begun.

If you have any doubts about whether a file is ready to be copied (has a target defined for it) or why DBE won't do anything with it , you can check the status flags with this option.

For instance, if you copied a file, the Copied and Tried and Found flags would be on; and to copy it again you would need to reset them. You can check to see if they are on by pressing ? and then, if need be, pressing Ins to reset the COPY tag and then E to re-enable copying, which resets the Copied and Tried flags. Once these flags are clear, it can be recopied (this happens by default when the processing routine starts; but the Copied flag must be clear before the processing routine will even consider operating on the file).

If a file was not found when DBE looked for it, the Tried flag would be on, and the Found flag off. If the Ignore flag isn't set, DBE would keep looking for it every time it went through the processing list.

Note that when C, D, or M are pressed to copy, move, or delete a file, the state of these flags are automatically set so that copying, deletion, or moving begins immediately, and you are prompted to set the target. Even if the file was tagged and a target previously set, you still must set a target for it. Thus, instead of a prompt "Copy file to wherever ? (Y/N)", you either choose a destination or press Esc to cancel the operation. This provides a means of overriding in individual cases the targets you may have

already set for groups of files, as well as providing immediate action. Since the <Copied> and <Tried> flags are on and the file status set to <Copied>, the file must be retagged in order to be included in the Go/Process list.

Files must be tagged, at minimum, to be included in the Go/Process list. If any have no target, one will be prompted for. It saves time to set targets for the files in groups if you can, and you can use the selected list to cull out files temporarily while you tag and set targets for them. If for some reason you don't want to set targets, set the opposite window to display the target directory, and then you only need to press O when prompted for a target once copying begins.

The Ignore status of a file can be toggled on or off by pressing 'I'. A red asterisk appears next to the Tag area when a file is being Ignored, and the characters such as . ► · ■ which appear under various conditions between the filename and extension also appear in red. Note that setting/changing targets for other files will remove the Ignore status if the selection bar is resting on the filename.

When an entry for a file is created during a disk scan, none of its status flags are set, except for after the scan if a string search is active and the file contained the search string (in which case the file will have the Selected status after the post-scan selection operation completes). If you want to restore a group of entries to the state in which they were after the disk scan, you can zero their status flags (and deselect them and remove them from the buffer) using the CtrlZ key.

■ DISK PRE-SCAN INDICATORS

If CapsLock is on when you press a key which starts a disk scan, the scan range, type, and specification will be displayed, and you may press Esc to cancel the scan if need be. There are six indicators which show the effect the scan will have on the current list:

```
SCOPE<PATH>=DirScope          ♦ScopeOK ♦EntCS  ♦LevCLR  SEARCH OFF
SCAN<CURRENT>=ScanCrnt        ♦EdFsp   ♦EntRFS ♦DskCLR  ····················
SPEC: C:\CQVIS\*.*
```

The indicators, which relate to the function of the scan key pressed, are as follows:

```
  ScopeOK - The indicated scope applies to the operation
  EdFsp   - Editing of the filespec is normally available
  EntCS   - ClearStatus is in effect during this operation
  EntRFS  - ResetFileStatus is in effect during this operation
  LevCLR  - Levels will be cleared from the list beforehand
  DskCLR  - The Disk record will be cleared from the list beforehand
```

(NOTE: DBE 3.1+ versions do not implement the DskCLR function.)

If EntRFS is in effect but EntCS is not, the DestDef, Found, Copied, and Tried statuses will not be cleared; files will have their statuses set to <FileOK> except for new entries, which will still have <Verify> status; and files will be untagged.

When LevCLR is not in effect, the status changes will only affect files which are found during the scan; i.e., the updated list entry will have its tag and statuses cleared depending on whether EntCS and/or EntRFS are in effect. When LevCLR is in effect, new entries are created for all files because the list is cleared, for each level, before the scan begins.

The DskCLR function no longer (DBE 3.1+) replaces the Disk list entry, nor does it affect the root directory, although scans with both LevCLR and DskCLR active will affect the statuses of entries in subdirectories, completely clearing them because LevCLR is in effect.

■ SETTING TARGETS / KEYS / MODES

Press 't' to set the current target, or to see if one is defined. If you then press B and a drive letter, the target will become the backup drive you chose; this is as good as any for a default.

Note that pressing 't' or 'T' does not make a difference when it comes to the currently selected file, but when 'T' is pressed, after the target is defined for the current file in the same way as if you had pressed 't', you are asked whether to set targets for other files which are UNTAGGED, TAGGED, or BOTH and also those with NO TARGET, WITH TARGETS, or ANY.

The T set targets prompt is as follows:

```
 SOURCE: [F:] F:\GM\DBF.TXT
 TARGET: [F:] F:\GM
 SELECT: Esc CANCEL  NEWDIR  OPPWIN  SELECT  DRIVE  FILENAME  TARGET  BACKUP
```

The target types are associated with these options as follows:

	NEWDIR	OPPWIN	SELECT	DRIVE	FILENAME	BACKUP
ToDir	■	■	■			
ToDrive				■		
ToBkpDrv				■		■
ToFilename					■	

In general, use DRIVE if you just want to put the file on a disk in a drive without having to choose a directory for it. If you use a GO with SubsScope you will be prompted for whether or not to include a relative path in the target file name.

SELECT is used to make sure the file goes to the directory you want it in, on the particular disk it's supposed to go to; relative path may be relevant here if you use SubsScope for the GO. NEWDIR is the same as SELECT except that it creates the directory, if necessary, before setting the target to it. If the version of DBE you are using prompts you for the location of the new directory and then allows you to enter lowercase characters, it supports creating long filenames for target directories.

If you choose to use FILENAME as the target, you will generally be prompted to edit the filename not only when you set the target but also when you are copying, to ensure that setting targets for groups of files has not conflicted with the concept of copying each file to a particular file name. Windows LFNs take precedence over the 8.3 filenames when you have enabled LongFilename copying in DBE, but to change an existing target LFN you may have to choose the 'F' (copy to FILENAME) response in order to edit the filename. DBE will copy from one 8.3 filename to another while the LFNs for the two files differ only in case, and will leave the target LFN as it is. So if you have NoAI.bat and NOAI.BAT (which both exist as NOAI.BAT on the memory list) and copy from one to the other, even though the LongFilename of the second might be NOAI.BAT (it could as well be noai.bat and the same situation would arise) the target LongFilename won't be changed unless you choose a ToName target or edit the filename and set an LFN for it then.

If you want to protect the targets you have set from being changed, use 'I' to set the Ignore status flag. You can still change it by placing the selection bar on the file and using one of the key options that sets the target directly; and this removes the Ignore condition, but the Ignore flag protects the target from being changed when the file just happens to fall within the Scope and Selection criteria of a target-setting operation initiated from some other point.

If the Set Last Target On Ins flag <SLTOI> is set to Auto, any time you press the Ins key to tag a file the currently defined target will be assigned to the file. The same goes for the Set Last Target On Plus flag <SLTOP> and the

Plus key, although the tag depends on the current tag value as discussed earlier.

If <SLTOI> or <SLTOP> are set to Prompt, the T option will be activated when you press Ins or Plus, and you can accept the current target by pressing T to set the current target or you can choose another, which will become the default current target to which the T suboption applies.

You may find it tedious to leave both <SLTOI> and <SLTOP> on Prompt, especially if the files being tagged are all destined for the same target directory. It may work out best to generally set <SLTOI> to Prompt and <SLTOP> to Auto; that way you can choose the target when you press Ins, and have the current target automatically set and the file tagged (as well as advancing the selection bar) when you press Plus.

Note, however, that when selecting files using the S option, no targets are set for any of the files; they are merely tagged (if you choose that option). If you do not use the Auto-tagging for the Ins and Plus keys, and instead set targets for defined scopes before setting the tag values, you will wind up with directories full of files with defined targets but not necessarily with tags. Whether you use the T option or the Auto or Prompt values in overriding the targets set when you do tag the files, doesn't matter; at minimum if you set all the targets beforehand you won't be prompted for them when processing begins.

Clearing targets with the Minus key is enabled if the Clear Targets On Minus flag <CTOM> is set. When this flag is On, not only will the COPY or MOVE tag be cleared but the target link will be removed and the DestDef status flag turned off for the file. If you decide to defer copying temporarily for a file, you can use Minus to clear only the tag and leave the target link intact, if the <CTOM> flag is Off. (You can leave it this way and use Ins to set or clear the targets if <SLTOI> is Auto. Then, when the tag is toggled off by Ins, the target link is cleared as well.) The Minus key doesn't COMPLETELY clear file statuses, and will not change their buffer status. To REALLY clear all of the file states, use the ^Z key.

<SLTOI> and <SLTOP> can be set to Auto, Prompt, or Off when you press one of the Shift keys along with the Ins or Plus keypad key. Pressing a Shift key and the Minus key toggles <CTOM> on and off, displaying the current value for a second or so. If you normally leave <CTOM> in one state but occasionally need to briefly change its state and press Minus to clear an entry's status, you can quickly press Minus three times, with Shift pressed the first and third times, so that the current file is operated on when <CTOM> is in the opposite state.

When you do not answer 'Y' to the relative path prompt, ToBkpDrv and ToDir assign the base drive+path to the value you selected when setting the Backup directory or the target directory, and to the base is appended only the filename unless you have enabled MakeDirs, which appends the complete source path+filename to the ToBkpDrv and ToDrive base name. Only when you have not enabled MakeDirs can you choose to include the relative path, and when you do, only the ToDir target is able to append a selected path to the base drive letter, and the backup directory name is ignored. The MakeDirs feature is associated with targets that do not explicitly link to a particular directory on a particular disk, and when enabled with a disk target it appends the full source path to the base drive+path. You can opt to use either MakeDirs or MakeSubs, or neither; but not both at the same time.

■ BUFFERING REMOVABLE DISKS

The ^B key option allows manual buffering of files (not auto-buffering, which is not available in Ver. 3.0). When you press ^B, you may choose the scope over which buffering takes place; if you press Esc instead, you will be given the opportunity to delete all of the buffered files. When you exit DBE to DOS, if the buffer still contains files placed there by DBE, you will be

prompted to either save the configuration or delete the buffered files, and if you press Esc instead of choosing either of these options, you will have a rather messy- looking set of subdirectories in the buffer directory which DBE will never be interested in removing; so let DBE clean it up if you don't intend to save the configuration (and thus the states of the files which have been buffered) for later use. DBE v4.3N+ attempts to put MOVEd files back where they came from if they're in the buffer (which means they're not yet moved to their final destination) when you exit and don't want to save a FIG file; so cooperate with DBE and have the disks handy to restore these files to when DBE prompts you for them.

Swapping removable disks in a single drive to get a multiple-disk copy done can be tedious, especially if you haven't decided where you're going to put everything beforehand. For this reason, DBE allows for temporarily storing the files to be copied from each of the disks in the buffer directory or in specially-created subdirectories (the names of which you need not be concerned with if your disks all have identifiers on them). If you exit and leave the buffer directory in use, don't change its name; version 3.0 saves the name in a separate configuration file from DBSYSCON.FIG, under the assumption that there really isn't any need for more than one buffer directory in any case.

When buffering is used, DBE automatically tags the original files with the COPY tag (which will be in a bright color) and temporarily turns the color of the original file's name to gray. Buffered files really stand out in the file list; and for good reason.

If you want to copy one floppy to another, you can buffer the whole disk by pressing ^B and then V. You can set the target to another disk by pressing T and then V, and then choosing the target. The files will be copied from the buffer when you press G and then V, and all you need to do is insert the target disk when it is needed. Remember that Volume scope refers to the current window, so make sure the right disk is displayed in the active window before setting the scope to Volume scope.

If you don't want to spend the time buffering the whole disk, wait until DBE prompts you for the target disk, and press ^B. This starts buffering for all of the files in the scope of the current GO operation, and DBE will perform the copying/moving in two cycles, with the first being from the source disk to the buffer and the second being from the buffer to the target disk.

When they are buffered, the files are tagged. If you haven't already set the targets, they won't be set, and you can either select them and set targets for the Selected scope or set them manually if you have more than one destination in mind for them.

If you have twenty disks to exchange files on, read them all, buffering the files you think you're going to need to switch around or copy. If you do this when you first read the disks, you'll only need to insert them once each until copying begins; and you can use = to show the disk status page and compare visually at a glance the free space on them, and you can then set targets and turn tags on and off until you have the destinations set the way you want them. Then, you can insert the disks in any order and copy the files to them (if you haven't planned deletions to make space) without having to keep re-inserting disks into the drive.

DBE 4.1 doesn't use a knapsack algorithm to figure out the logistics of the process. Making an exact calculation of the projected free space after copying to each disk is dependent on correct DPB and BPB information, so if the little blocks on the 'disk map' page look out-of-kilter, it's a sign that Windows is making DBE fly by the seat of its pants instead of handing over the real disk data.

Version 3.0 does not delete the original files which are MOVE tagged when they are buffered, although this is ideally what should be done if they are on removable disks which are to be copied to and the space will be needed for other files yet to be copied to the disk. So the BufCdel and SrcCdel flags are set when deletion is scheduled to take place, but version 3.0 ignores them and you have to delete them manually (the previous version had all file

MOVEs disabled; only the copy was done).

Versions 4.1 and newer do delete files when they are scheduled for a MOVE and you choose to buffer them. When you exit DBE, if the copy from the buffer to the target has not been completed, DBE will prompt you to insert the source disk into the drive so that the file can be restored to its initial location, if you choose to delete the buffered files instead of saving the configuration in order to complete the operation at a later date.

There are three ways to instruct DBE to remove files from the buffer, copying them back to their original location if they had been moved to there. The Bksp and ^Bksp keys, which cut entries from the list, and the ^Z key, which zeroes the statuses and tag/target states of entries, will perform this action. Note that using ^R re-reads the directory FROM SCRATCH; i.e., any files which exist in that directory which have been buffered will be removed from the buffer before the directory is re-read. If you use RC instead, the buffered files will be untagged, but will still be buffered.

■ NAVIGATION

Cursor and paging keys, plus the <◀▲ and >▼▶ keys traverse the file list in a sequential manner, usually stopping at the ends of a directory list. The < and > keys don't do anything when no files are tagged, but when there are some, they jump between them, stopping only at the extreme ends of the list. The ^◀ and ^▶ keys switch directories, and when you run up against an empty directory the , and . keys won't venture into, use these to get into the empty dir (the ,◀▲ and .▼▶ will jump from one empty dir to another easily enough, but once they find one with files in it, they won't go into another empty one... they only go OUT of them, if they can, or TRY to, anyway). To get out of an empty one in a hurry, press Space, and the MRU list of recent locations will pop up. (It will also pop up if you press C, M, D, or some other keys that require a current entry to operate on, whenever there isn't any current entry in the active window.)

If you have the ScrollLock toggle turned on, the current file name will be shown on line 1, the file status display is shown (as if you had pressed '?'), and if a directory is empty you will see a message indicating which keys can be used to navigate out of it. When, in addition, Queue Enable is on, the file Process Numbers will be displayed; and if Queue Start is in effect, the current file's Process Number will precede this information on line 1.

If you want to breeze over empty directories without even noticing them, use the , and . keys to scroll the file list. The bottom two lines of the display show the current directory, and you might keep your eye on it while using the comma and period keys to traverse the file list. Any time you think you just passed up an empty dir when navigating with , and . you can check using the ^◀ and ^▶ keys.

The \ key opens the select disk window, and here the ◀ and ▶ keys move up and down the level list, with the current window file list for each dir shown in the opposite window. The ▲ and ▼ keys just move the bar without showing the file list window opposite. If you want to flesh out the skeleton of the disk tree without adding files to it, use the select disk window and the ' key to select and then read the subtree (without files) for any dirs you want in the list. Keep doing this until you have filled out the particular branch you want to have in the list, and you'll still have lots of memory left for the list of files you're most interested in.

■ SORTING OF FILE ENTRIES

You can manually sort the files by using the Selected list to hold the files you want to move. Select them by pressing '-' for each one in the order you want them to appear wherever you're going to put them; then move

the selection bar to the point where you want them to be placed and press the '_' key. If the Selected list contains files from other directories, these others won't be inserted; they'll still be on the Selected list when you're done with the insertion. However, any files you had selected which were from the current directory will have been removed from the Selected list.

Use the following keys to sort by time, extension, name, size, original ordering, or by Queue Processing number:

aT TIME
aE EXTENSION
aN NAME
aS SIZE
aO SORT OFF
aP PROCESS #

Pressing the same sort key twice in succession inverts the previous sorted order, toggling from ascending to descending; pressing a different sort key sorts in ascending order. Note that when viewing the Queue list, the Process# is changed by the sort. The Selected list AltO sort can be used to invert the 'sort off' ordering of the files because this list is linked independently of the Entry-to-Entry links in the Level list.

■ SAVING SYSTEM CONFIGURATIONS

You may exit from DBE and save the states of all files on the memory list. This includes whether they are tagged for COPY, MOVE, or DELetion; whether they are to be ignored during a subsequent copy operation, whether they were found when DBE looked for them during processing; whether an operation was actually tried; and if they were copied, the new file's state in the target directory is also saved (providing the directory has not been re-read).

The target directory locations are also preserved, and this means that you can tag any files, anywhere on any disk(s), and set the copy destinations for them individually or en masse, and if you then, instead of pressing G to begin copying them (or whatever), exit from DBE using the AltX key, the file DBSYSCON.FIG will be created, and if this is available the next time you run DBE, all you have to do to copy (or whatever) all of the files is to press G and start putting the floppies (or whatever) in the drives when DBE requests them.

If you run low on memory when scanning a large hard disk for files, you may use DBSYSCON.FIG to save everything. You can use a batch file to run DBE with several different DBSYSCON.FIG files (some of which may provide for copying/deleting the configuration files themselves) in turn. Version 3.0 does not support the use of multiple configuration files.

IF you scan a disk in parts, and use Bksp or CtrlBksp to drop files or dirs from the memory list, you can reclaim some memory by saving the configuration and restarting DBE before continuing. If you're low on memory it may be wise to divide up a large task into several stages in this manner.

DBE 4.40 (04-11-11) memory requirements are: 79 bytes per file, 88 bytes per directory, 264 bytes per ToFilename target, 32 bytes per target definition, and 106 bytes per disk. This must all come from real-mode memory in the DOS Transient Program Area.

Version 3.0 of DBX placed the DBSYSCON.FIG file in the directory containing DBX.EXE (if AltX is used to exit the program), and if it was not found there (the EXEdir) at startup, DBX looked for it in the DOS PATH. If it was not found, DBX looked for it in the current DOS default directory (DOSdir).

Version 4.0 allows the choice of locations to which to write the DBSYSCON.FIG file as follows:

```
          NONE   EXEdir   DOSdir   TARGdir
```

If you exit with CtrlX instead of AltX, you may choose in which of these to place the DBSYSCON.FIG file.

If you save a configuration and then copy it into the current DOS directory, DBE will automatically read it in at startup if you have deleted it from the DBE directory, provided you have set the location for it to DOSdir.

If worst comes to worst, you can load the DBSYSCON.FIG file from the command line, or with MS-Windows 'drag-and-drop' of the file icon onto the DBE icon. Note that in this method of loading the configuration, the file does not need to be named DBSYSCON.FIG, but be aware that- chances are -the filename is going to wind up in the active window at startup if the file isn't a valid DBSYSCON.FIG file created by DBE. What happens in this event is that, with DBE v4.2+, the default drive is searched for all instances of the file, DBE places them all on the Selected list, and displays the list at startup.

The thing which might be confusing at some time is if you change your PATH and you have left DBSYSCON.FIG files scattered around. DBE will use one of them if it can find one at startup; so if you're running DBE and the same directory comes up on the screen at startup no matter what directory you start it from, you'll know DBE is finding a .FIG file somewhere; either in the DBE.EXE dir or somewhere in the PATH.

DBE v4.0 and later versions do not write a DBINMEM.TXT database map file. DBR.EXE manages this so that DBE can use the memory for other things. If you want to see how the data in DBSYSCON.FIG is laid out and what the tag values and primary status flags are, run DBR after DBE has created the DBSYSCON.FIG file. DBR will create DBINMEM.TXT and let you page up and down through it or find strings in it. It will also allow you to view the DBSYSCON.FIG file in hexadecimal/dump formats if you run it with /D or /FDBSYSCON.FIG as a parameter.

DBE v4.3N allows the additional choice of selecting a directory to save the configuration file to:

```
      FILE: FLASH.FIG     G GENERATE FILENAME  C COMMENT  R RECORD NAME
 SYSCONFIG: <SELECT>       NONE  EXEdir  DOSdir  TARGdir  -≡-  SELECT
   WRITE E:\GM\FLASH.FIG
```

When the FIG path has been chosen from the SelectLevel list, the type of path used is denoted by <-≡-> instead of the usual 'EXEdir', etc; or it may be denoted by <SELECT> if the file still existed the last time DBE checked for it. In any event, you press 'W' to write the FIG file, and 'L' to load one at startup, so you always have a chance to change the name or path to the file.

If you press 'C' or 'R', you can edit the record name (limited to 58 characters) or the block comment (limited to 16 lines of 78 characters each). See the SoftWindow help (available by pressing AltH while editing text) for keys tied to the built-in editor functions.

■ STARTUP DBCONFIG.CFG/DBSYSCON.FIG

If no DBCONFIG.CFG file exists, the DisplayTotals option will appear, and you may then set the AutoScan, MakeDirs, ClrReadOnly, TurnOffArch, KeepTimes, SLTOI, CTOM, and SLTOP, flags as well as the Backup, DOS, Target, and Buffer directory names (To change these settings later, press = or ↑Ins, ↑Minus, or ↑Plus).

Turning on CapsLock, NumLock, or ScrollLock before starting DBE causes the selections NONE, EXEdir, DOSdir, and TARGdir to appear. If you use CtrlX to exit from DBE to save DBSYSCON.FIG in one of these locations, this is the way you can ensure that the one you saved will be read in at startup. AltX on exit is supposed to place DBSYSCON.FIG in the EXE directory in which DBE resides, and DBE should automatically read it in at startup if you do not

select one of these startup options.

If you place the name of a .FIG file on the command line, DBE will check for a comment block in that file at startup, and will display the comment and give you a chance to press Esc to quit rather than loading that .FIG file.

■ COMMAND-LINE FILE SPECIFICATIONS

If two file specifications are entered on the command line, DBE interprets both of them as names of files to be scanned for and added to the memory list. If these specify different directories, two directories will be displayed in the two panes; otherwise, you will see a merged list there.

DBE responds differently to the following four types of file specifications when only one is given on the command line:

(1) Containing wildcards ?* or multiple spaces
(2) A slash character /
(3) A drive or directory name
(4) A filename containing a . character
    (4a) File is a SYSCONFIG file
    (4b) All other files

Case (2) causes a filespec prompt to appear, and DBE will not load a SYSCONFIG file. If you press Esc, DBE will search for files having the names *.DBE in the current path and add only those to the memory list. If you enter a file specification, DBE will scan for qualifying files.

Case (4) pre-empts all but (2). DBE inspects the file's format and offers to load it into memory if it is a SYSCONFIG file. If it is not, DBE scans the current drive to find all instances of that file, and adds those found to the Selected list.

For case (3), if you specify a directory name without a drive name or a trailing \ character, it must be longer than 3 characters or DBE will treat it as a file extension and search for files with that extension in the current directory.

Case (1) is handled in a unique manner if multiple spaces occur in the filespec (which means that it appears to be 3 or more parameters rather than a single string). Long filenames containing spaces have nothing to do with this feature, and DBE doesn't recognize them at the command line. DBE allows spaces to appear instead of \ characters, except at the end of a directory name not followed by a file specification. Also, if spaces are present, wildcards need not be; and DBE will attempt to construct a fully qualified file specification from it, treating the entire command line as a single string. The documentation for the CQvis program D.EXE contains examples of a similar usage.

■ SPECIAL FILES OR FILENAMES

The filenames with special significance to DBE are:

```
DBXHLP.LIB
DBX.DOC
DBE.DOC
DBCONFIG.TXT
DBCONFIG.CFG
DBSYSCON.FIG
```

The search at startup for the DBCONFIG.CFG file looks for it first in the current DOS directory, then in the directory defined by the 'CQ' environment variable, and then in the directories listed in the 'PATH' environment variable.

The role played by DBSYSCON.FIG can also be played by files having other

names, but these files must be created and updated by DBE and should never be modified by other programs; especially text editors and word processors. The DBCONFIG.CFG file is created and updated by DBE, and should never otherwise be modified.

The DBCONFIG.TXT file is never created or modified by DBE. The user of DBE must create this file, if necessary, using a text editor. Only the IDPATH declaration is supported by DBE v4.3N. The IGNORE declaration is new with DBE v4.4O. Lines of this file beginning with 'IDPATH ' and 'IGNORE ' are followed by fully qualified directory names in DOS format, not ending with a '\' character. IDPATH is the location on a fixed drive where a DiskID is to be written by DBE to be used as an identifier when the root directory of the drive is read-only (as in many networked systems). This enables DBE to recognize a volume during at least one logon session if the drive is a logical drive which is nonexistent when the session ends. If you need to save DBE's memory lists of files, you should run DBE from a USB flash drive and make sure you save your work there just as you would with any other backup software. If you use IGNORE directives for any directories on a disk, they will be duly ignored, for the most part, by DBE, although you may deliberately initiate limited scans for particular files in single directories. See the section on Disk Scan Ranges.

The DBXHLP.LIB file is a binary displayfile library of help information created by the author of DBE, and should never be modified.

Files with arbitrary names but with content which is of special significance to DBE are the FILESET file and the FIG file.

The FIG file, which has the default name of DBSYSCON.FIG, can be renamed with no restrictions other than that the name must not be one of the other names which are of special significance to DBE. The format of any file which DBE is asked to load at startup is checked to see if the file is a FIG file created by DBE, and this depends on the file contents, and not the filename.

The FILESET file, which can be written by DBE, or created using a text editor or the D.EXE AltF LISTFILE feature, is recognized by DBE by the fileset declaration on the first line of the file. After the fileset declaration follows the list of filenames for each of the files in the fileset, and content other than valid filenames is largely ignored by DBE; i.e., it will be ignored if it doesn't resemble a valid filename. The best bet for placing comment lines in a FILESET file is to follow the declaration with lines beginning with a semicolon, percent sign, bracket, or number sign followed by a space if anything other than a valid filename appears on the line. The fileset declaration, which you must place on the first line of the file yourself if you use D.EXE or a text editor to create the list of filenames, is as follows:

```
FILESET [COPY|COMP] [TARGET DirName]
```

The word FILESET is sufficient to identify the file to DBE as containing a list of names of files which are to be searched for on a disk and added to DBE's memory list, optionally tagged for copy or comparing, and optionally assigned a target directory for the copy or comparison. When the selection bar is resting on the name of such a file and OI, O^I, or OF, are pressed, the file is opened and read by DBE; and if the word FILESET begins the file, DBE will treat it as a fileset file and scan the disk, adding the matching files it finds to the memory list (the ^I instead of I causes the files found to be added to the Selected List). The 'F' causes clearing of the level from the list whenever a different pathname appears in the fileset file.

DBE 4.4O builds after 04-09-11 will write long filenames to the FILESET file as a matter of your convenience for uses not involving DBE; but it is still necessary to write the short 8.3 filenames to the FILESET file if you intend to have DBE read the file and add entries to the list.

The fileset file is related more to the relatively static external environment in which DBE operates, and the FIG file more to the dynamically

changing statuses of the file entries internal to DBE's data structures while the program is in use. The FIG file enables a continuity from one DBE session to the next, preserving the context of the processing operations in their varying degrees of completion. The fileset file enables the DBE user to impose on the operation of DBE some of the pre-existing context and relationships between the files, as they stand in the long-term assessment of the user of DBE, and to quickly ready DBE to work with various groups of files on a project-by-project basis, without having to "hunt them down" or sort through large numbers of files to find them. Although the FIG file preserves more status information, and can enable getting started with just a couple of keypresses, there may be a variety of operations called for, on the same group of files, at different times; and starting 'clean', with no need for the processing history or statuses, may be more convenient; but this depends on the context in which DBE is being used at a particular moment. DBE will write the fileset for you; and all it does is to preserve a list of files, whether they are to be tagged, and whether they are to be assigned a target. But many times, that may be all you need.

■ SOURCE FINDING FOR KNOWN TARGETS

Read the drive containing the files you wish to update and then select (use '-' for individual files) the files on it you wish to update and for which you need to find the newer files on the source drive to update them with.

Press '\' and choose the directory on the disk the updates will come from, or choose the parent directory whose subdirectories probably contain them.

Press F4, and then after the disk scan is finished, press 'S' to select, 'S' for Subs scope, and Enter. (Note: If all of the files found are to be copied, you could tag them all by pressing 'T' and then 'C' now, instead of setting targets first. Doing it this way gives you a chance to see them all in the same list, and you can use '-' to drop the ones you don't need, if there are older files in the list that you don't intend to use as replacements.) The Selected list will appear, containing the files matching the names of those on the target disk which you selected.

Press 'T' and then ^S to set targets for the selected files, and press 'S' to SELECT the target directory. Choose the directory on the target. All of the Selected files will now have the that directory as the target.

If all of the selected files are to be copied, press 'S', ^S, 'T', and 'C' to tag them all for copy.

Press 'G' and 'S' for GO and Subs scope, and when the message appears asking you if you want to replace files, etc., you can press 'A' for ALL.

This may seem complicated, but you may find additional copies of files which might need to be considered, in the process.

■ COPYING ONLY FILES WHICH DIFFER

If the source files are in a known location on the drive, use the '\' Select Level option to find the location and press ^C to tag all of the files in that level for compare (COMP). Press '\' to exit the SelectLevel option and set the targets for that path to the drive containing the target (press 'TPD' and the drive letter, or 'TPO' if you have read the target in the opposite window). Press 'GP', and DBE will DIFF tag all of the files which differ. Files not found on the target will still be COMP tagged. Press 'ODP' and Enter. DBE will change the DIFF tags to COPY tags. Check the files which are still COMP tagged to see if any additional files are to be included which were not found and tag them if necessary. If you want to skip doing a second compare of the COMP-tagged files while copying, press the LEFT Shift key with the 'G' key to start the GO the second time (with 'GP').

Preserving incremental backups can quickly be carried out by comparing the

files in the source directory to those in your previous backup, and after changing the DIFF tags to COPY tags set the targets to a new directory and then perform the GO. If you will need to back up the same set of files repeatedly, either save a .FIG file before issuing the GO or use the OW option to create a FILESET file which you can read in later to find and tag the files to be copied (and set the target as well). The FILESET will work better if you can't guarantee that the same disk will be used to copy the files to, because when it is read in, the target path is searched for on a drive which you specify instead of being directly linked to a DiskID for a particular disk as it is for a <ToDir> target in a .FIG file.

■ DRIVE LETTER MIX & MATCH

Swapping multiple disks in and out of multiple drives may be predictable until Windows throws you a loop and reassigns a drive letter on you, and won't let you change it without rebooting (this is a common USB flash drive problem). You may find that DBE will display the following prompt when you attempt to scan a particular directory on a particular disk but the drive letter was reassigned and/or the disk now has been assigned a different drive letter by Windows. In the example situation, drive H: no longer refers to a flash drive currently plugged in; and a key has been pressed to start a scan of a particular directory on that flash drive. If the drive now 'occupies' a different drive letter, you would press ^A..^Z to pick the new drive letter, or press Enter followed by the drive letter. If there is a disk you intend to insert in the drive (here, H: is invalid), insert it and press Space; and DBE will ignore the fact that you were looking for a particular directory on a particular disk and just scan that drive.

```
MSG.BAK       ...A 100510 1628 3933     KMAC.TXT      .... 121509 1445 7692
MSG.TXT       ...A 100610 0012 324      IO2W.TXT      .... 030410 1435 155
DBP.PAS       .... 093010 1909 18033    DBCONFIG.TXT  .... 092310 0025 553
USES.RTF      .... 071610 0749 4915     CNOTE.TXT     .... 092309 1040 199
DBS.BAK       ...A 100510 0527 43339    SOR2.TXT      .... 021510 2329 8430
DBFCOPY.BAT   .... 122809 1358 1913     UPL_LST.TXT   .... 020310 1723 821
USES.TXT      ...A 071610 0750 3849     FIX.TXT       .... 010810 1250 53537
```

```
DISK DRIVE H: IS NOT VALID


YOUR OPTIONS ARE:

 Enter SCAN FOR THIS DIRECTORY (ANY DRIVE)
   \   SELECT ANOTHER LEVEL/DISK AND RETRY SCAN
 Space SCAN ANY DISK IN DRIVE H:
^A..^Z SCAN ANOTHER DRIVE FOR DISPLAYED DISK/DIR
```

```
DISPLAYED DIRECTORY IS ON DISK [FY8X7H] KINGSTON.1B
```

```
MAINDBE·BAK  ■■■■■■■■■■■■■■■■■■■■■■■·  UPL_LST.TXT  .... 020310 1723 821
E:\GM                                  H:\GM
KINGSTON.1   39305216 984678400        KINGSTON.1B  3096576 1020887040
```

DBE attempts to identify the directory you wish to scan with the disk you read it from, and it will let you know the DiskID and DOS volume label of the disk containing the directory being displayed in the current window. If you had two directories with the same name on two different disks and had them mixed up when you started a scan, you might get a message like the above. In this event, you would press '\' to open the SelectLevel window and type in part of the directory name and press '/' to find it on the list, and then ';' to start the scan.

-=-

■ SUMMARY OF KEY FUNCTIONS

The following is a summary of functions associated with the keyboard keys

```
Pg↑↓      Page up/down
^Pg↑↓     Scroll up/down with selected file centered in window
Space     Center selected file in window; or if directory is unread or
            empty, pop up MRU list of recent locations to select from.
            Space also exits the MRU select routine.
Enter     Go to next Associated file, to Selected file's directory, or
            toggle between source and target files if an active link is set
/         Associated files display toggle
,         Prev file in crnt or prev dir (if not empty)
.         Next file in crnt or next dir (if not empty)
^◄        Prev dir on disk
^►        Next dir on disk
<         Prev tagged file
>         Next tagged file
------------------------------------------------------------
J j       Jump to a file (Uppercase J repeats last type of jump)
            P  to previous jump point
            N  to next jump point
            H  to here (the current position); set a jump point
            L  list jump points 1..9 by number
            =  replace a jump point with the current position
            X  jump to next target dir
            V  jump tp prev target dir
            D  jump to current file's target dir
            T  jump to the currently active target
            R  Return to the last entry
            K  jump to the next Backup-tagged file
            C  jump to a tagged file which has been copied successfully
            I  jump to a tagged file to be ignored during further copying
0         Display the jump list
1..9      Jump to jump pts 1..9
Alt+#     Jump to Queued file entry #
AltJ      Same as lowercase j but acts on opposite window

------------------------------------------------------------
ªA        Associate files within scope with others having the same name
+         KEYPAD: Set Entry tag to global TagVal
-         KEYPAD: Clear Entry tag
-         TEXT: Toggle on/off selected list
Bksp      Drop Entry from memory list
^Bksp     Drop Entries within scope from memory list
            Select Scope: File ^Selected Path Subs Window Disk Volume Global
*         Selected files display toggle
_         Insert files from Selected list at current file position
------------------------------------------------------------
= ↑8      DisplayTotals
           Pg↓ Go to Select Disk page
             R ReadOnly bit clearing toggle
            ^A Auto scan toggle
             A Archive bit reset toggle
             M MakeDirs toggle
             K KeepTimes toggle
             G Set default scope
             B Set Backup drive/dir
             D Set default dir name
             T Set default target dir name
             U Set buffer dir name
```

```
            F Set fixed memory reservations (Disks,Levels,Entries,Targets)
            C View FIG comment block, if any
          Ins Set <SLTOI> state (Set Last Target On Ins)
            + Set <SLTOP> state (Set Last Target On Plus)
            - Set <CTOM> state (Clear Target On Minus)
| ↑8      Select disk ( same as '=' PgDn )
            ^A..^Z      Change drive designation for disk
            NumberKeys  Select disk by number
            Pg↑↓ ▼▲     Move selection bar
            ◄ ►         Exit/select disk and window
            Enter       Select disk and exit
```

```
O           Options
              R Re-enable copy for copied files
              D Change DIFF tags to COPY tags
              C Clear status/tags for files
              S Set a DiskID file for a disk
              \ DRIVES check for reassigned drive letters
              L Turn Long Filename creation on and off
              I Input FILESET from the current file (scan disk for files)
            ^I   "      "    and place files found on the Selected List
              F  "      "    but clear levels when encountered
              W Write a FILESET to a text file
aH          Display the DBX.DOC help file
 H          Show mnemonic help screen(s)
aK          Display the DBE.DOC topical summaries file
T t         Define target for file(s) in scope (T uses Tag/Target Qualifiers)
              Select Scope: File ^Selected Path Subs Window Disk Volume Global
              Define target: OppWin NewDir Select Drive Filename Target Backup
^\          Make the directory in the active window the current target
^Enter      List Targets
a=          Display files and targets together
L           List view modes and select one:
              NORMAL  FILE/TARG  TARG  LEV  SEL  DSK  JMP  ASSOC
C         ¶ Copy selected file, defining target if not defined
M         ¶ Move selected file, defining target if not defined
D         ¶ Delete selected file
V         ¶ Verify selected file by comparing with its target
P         ¶ Process one file
K           Rename selected file (press Esc to cancel)
G           GO; process all tagged files within Scope prompted for
aG          GO by TARGET disk order (as you place them in the drive)
             or GO in ReplacementMode
aV          GO; with a VERIFY check for each source file being copied
E           Re-Enable Copy for a Copied file
I           Toggle the Ignore status for a file
F           Find a file by name
↑Space      Start the type-in file finder
aF          Toggle SEARCH mode on/off; if on, edit search string
aI          Toggle IgnoreCase on/off for scans using search string
A           Again; repeat last find
N           Find next instance of current filename

NOTE:  The ¶ mark signifies that IMMEDIATE processing is carried out
```

```
Q           Toggle Queue flag for file and advance selection bar
Alt1..Alt0  Jump to file having Queue number (1..????) keyed in
??          Press ? twice, if QE mode (♦) is on, to view queue numbers
AltQ        Toggles View mode (same as QV)
q           Queue Options
              R Renumber
          0..9 Number entered replaces Process # (requires relink afterward)
              L Re-Link
              C Clear
              A AddTo  (from Tagged or Selected files)
              E Enable (Queue-On-Tag and ↑Q) toggling
              S Start/Stop (Go by Queue Numbers only) toggle
              V View (using Selected list functions) toggle
ScrollLock  When on and QE mode (♦) is on, queue numbers are displayed
```

```
  S         Select files by specification
  U         Unselect files by specification
               Select Scope: File ^Selected Path Subs Window Disk Volume Global
               Get select spec:  Tag Attr * Crnt Edit ▶ EXT ◀ NAME Date Size
             T Tag: SELECTOR:  Any Copy Move Del Diff Tagged Untagged ■OK■
                    SET TAGS:  Off Copy Move Del Comp
             A Attr: RHSArhsaN
             D Date: Set ◀ BEFORE ▶ AFTER + ON
             S Size: Set ◀ SMALLER ▶ BIGGER + EQUAL TO
             F Turn string search on/off; edit string
  ^S        Select without clearing the Selected list beforehand
  ^U        Unselect (currently, the same as U)
  ^Z        Zero statuses for file(s)
 ᵃB         Buffer files before further processing takes place
```

```
  R         Scan disk using selected scan range: RC RT RR RF RS RD RV
               Set scan range:  Current Tree Root Filespec Subs Drive Volume
  "         Tree scan, including subdirs
  '         Tree scan for crnt dir
  :         File/dir scan, including subdirs
  ;         File/dir scan for crnt dir
```

```
  [  F9     Left window
  ] F10     Right window
  ↑4        Reread (left window)
  ↑6        Reread (right window)
  ↑1 F1     One file per line
  ↑2 F2     Two files per line
  ↑3 F3     Three files per line
```

```
  \ ↑5 ^F4  Select disk/directory
             Minus   All tags off
             Plus    MOVE tag all
             Ins     COPY tag all
             AltC    COMP tag all
             Del     DEL tag all
             Home    Top of list
             Text    String to find in Level name ┐
             Space   Locate                       ├─ 'typer-finder' keys
             /       Locate next instance         ┘
             Tab     Locate next defined Target Level
             , .     Move to prev/next Level having file entries
             < >     Up/down half window height
ScrollLock   ?       Show long filename form of directory name
             ᵃH Show help
             ^A..^M Change drive designation for current disk
             Pg↑↓ ↑8 ↑2 ▲▼ Move selection bar
             ◀ ▶ ↑4 ↑6 Move selection bar and show view of selected dir opposite
             ↑1 ↑2 ↑3 Set files per line for dir shown opposite
             Bksp Remove directory (plus subdirs and Entries) from memory list
             F7 F8 ^R ᵃR ^S : ; " ' Select dir and exit to read disk
             ^\ Toggle 'NoSearch' flag on/off for current level
```

```
  ^T        Read tree
  ^R        Read current dir
  ᵃR        Read current and subs
  F7        Read crnt for left panel
  F8        Read crnt for right panel
  F4        Read crnt or scan subs to match selected files
  F5        Prompt for filespec and scan crnt
  F6        Prompt for filespec and scan subs; skip 'NoSearch' directories
 ᵃF4        Scan subs to match selected files, removing stubs from the list
 ᵃF5        Prompt for filespec and scan crnt
 ᵃF6        Prompt for filespec and scan subs, removing stubs from the list
```

```
  Tab       Switch active window
  F9 [      LEFT Window
  F10 ]     RIGHT Window
```

```
 aN        Name sort
 aE        Extension sort
 aS        Size sort
 aT        Time sort
 aO        Sort off
 ^A..^M    Scan disk in drive A: .. M:
 Esc       EXIT to DOS
 aX        EXIT and save memory list to EXEdir (DBSYSCON.FIG)
 ^X        EXIT  "   "   with optional comments and choice of filename/path
```

```
 File View Mode and Selection Keys
 ----- If Not in Viewer ----
 { }       Load current file into left/right window
 `         Load current file into active window

 ----- If Viewing Files ----
 ` X Esc   Exit viewer
 { } [ ]   These switch window or exit viewer if no file is open there
 F1        View text only
 F2        View hexadecimal only
 F3        View both text and hexadecimal
 F4        View dump of raw character data
 F5        Full-screen mode
 F6        Split screen horizontally
 F7        Split screen vertically
 F8        Toggle cyan/grey display color
 [ ]       Switch viewing window
 Tab       Exit viewer to opposite window
 1..9      Get file at jump point for view
 Enter     Get next Associated file for view
 N         Get next instance of file (by name),  from any level, for view
 : "       Get prev/next file in current level only, for view
 < >       Get prev/next tagged file for view
 =         Toggle Synchronize mode on/off (↕ appears if on)
 ; '       Scroll current window up/down
 , .       Page windows, if synchronized, or current if not
 /         Switch between current windows
```

```
 File List Navigation Keys:
 ◄►▲▼      Move selection bar
 ^PgUp     Move bar up, keeping centered if possible
 ^PgDn     Move bar down, keeping centered if possible
 PgUp      Up one page in list
 PgDn      Down one page in list
 Space     Centering off and window reset (if selection bar is not visible)
 Home      First file in dir list
 End       Last file in dir list
 Enter     For Associated files, cycle through the chain for the current file
           For Selected files, toggle between the dir and the selected list
           (Same for Queued files, also.)
 ↑Tab      Pop a jump point from the save stack
```

```
 File List Tagging Keys:
 Ins       Toggle COPY or COPY/MOVE tag (prompt for Target if SLTOI=Prompt)
 Del       Toggle DEL or MOVE/DEL tag
 aC        Toggle COMP (verify/compare) tag
 X         Toggle ->&lt;- (Replacement) tag
 Plus      Set last tag (and last Target if SLTOP is on), advance
 Minus     Turn tag off (and clear Target if CTOM is on), advance
 B         Set COPY tag and <Backup> status flags
↑Ins       Set <SLTOI> mode (Set Last Target On Ins)
↑Plus      Set <SLTOP> mode (Set Last Target On Plus)
↑Minus     Toggle <CTOM> mode (Clear target on Minus)
```

```
  If any of the following toggles are on when DBE starts,
  the location of the DBSYSCON.FIG file will be prompted for.
  Also, if a comment block was added to the FIG file being
  loaded from the command line, it will be displayed.
```

ScrollLock When on, shows the '?' file statuses continuously,
              also displaying Windows' LFNs (Long FileNames);
              and if Queue Enable (or Queue Start mode) are on, this
              shows Queue Process numbers continuously.
CapsLock   When on, causes the disk scan to show the scan flags
NumLock

LeftShift & RightShift

 If DBE stops when it has encountered a long filename or something,
 press and hold the pair of Shift keys temporarily, and DBE should return
 to the processing loop. IF you press Ctrl AND Alt and release them,
 DBE v4.3M will quit pausing when long filenames are encountered.

Ctrl   Generally (A through M) causes a drive to be read on a keypress.
Alt   For number keys above the keyboard, hold Alt while typing in the
        number of the Queued file to be jumped to.

Glenn A. Merritt

404 Birch St.
Lock Haven, PA
January 28, 2011

mailto://GAMerritt@HotMail.com
