BACK$ - Backup Windows Folder

The BACK$ routine can be used to copy all files from one Windows folder to another.

1. Invocation

To backup a Windows folder code:

```
CALL BACK$ USING bk [cp] [ba]
```

where the *bk* parameter is a control block of the following format:

```
01    BK
  02  BKVERS      PIC 9(4) COMP     * CONTROL BLOCK VERSION NUMBER
                                    * MUST BE 1
  02  BKCOPY      PIC 9(4) COMP     * COPYING METHOD
                                    * 0 = INVALID
                                    * 1 = USE COPYQ$ FOR FILE COPY
                                    * 2 = USE COPYX$ FOR FILE COPY
                                    * 3 = USE GXCOP$ FOR FILE COPY
                                    * ALL OTHER VALUES ARE RESERVED
  02  BKMODE      PIC 9(4) COMP     * MODE FLAG, DEPENDS ON BKCOPY
  02  BKPTR1      PIC PTR           * POINTER TO SOURCE FOLDER
  02  BKPTR2      PIC PTR           * POINTER TO DESTINATION FOLDER
  02  BKWILD      PIC PTR           * POINTER TO WILDCARD FILENAME
  02  BKCOND      PIC 9(4) COMP     * CONDITION CODE OF LAST FAILED COPY
  02  BKCRES      PIC 9(9) COMP     * $$CRES FROM LAST FAILED COPY
  02  BKFILES     PIC 9(9) COMP     * NUMBER OF FILES ENCOUNTERED
  02  BKCOPIED    PIC 9(9) COMP     * NUMBER OF FILES COPIED SUCCESSFULLY
```

The optional *cp* parameter is the standard control block for the GXCOP$ routine. The optional *ba* parameter is the standard control block for the GXBAR$ routine.

2. STOP Codes and Exception Conditions

The following STOP codes may be generated by BACK$:

STOP code	Description
1	Unknown version of BK control block (i.e. BKVERS does not contain 1)
2	Unknown copying method (i.e. BKCOPY does not contain 1, 2 or 3)
3	The length of the source folder is more than 256 characters.
4	The length of the destination folder is more than 256 characters.
5	The length of the wildcard filename is more than 256 characters.

Further STOP codes may be generated by COPYQ$, COPYX$ or GXCOP$ (depending on the value of BKCOPY).

The following exception conditions may be returned by BACK$:

EXIT code	**$$COND**	**Description**
1	1	Exception from NEOPN$ when scanning source folder. BKCOND and BKCRES contain the exception condition from NEOPN$.

3. Programming Notes

BACK$ is only available with GSM SP-30, or later.

The source folder name, pointed at by BKPTR1, must be a zero-terminated text-string. The destination folder name, pointed at by BKPTR2, must also be a zero-terminated text-string.

The wildcard filename, pointed at by BKWILD, must be a zero-terminated text-string. This parameter can be to restrict the backup to just a sub-set of the files in the source folder. To copy all the files in the source folder this string should be set to “*.*”.

The actual file-copy method used for the backup is selected by BKCOPY. If BKCOPY is set to 1, the COPYQ$ routine is used for the backup. The BKMODE field is used as the COPYQ$ *mode* parameter (see copyq$.doc for more details).

If BKCOPY is set to 2, the COPYX$ routine is used for the backup. The BKMODE field is used as the COPYX$ *mode* parameter (see copyx$.doc for more details).

If BKCOPY is set to 3, the GXCOP$ routine is used for the backup. This option is only supported on GX. Note that the GXCOP$ “server to server” copy mode is used for the file copy (i.e. BACK$ **cannot** be used to backup from the server to the PC running GX, for example). When GXCOP$ is used for the backup the optional *cp* and *ba* parameters are passed unchanged from BACK$ to GXCOP$ (see gxcop$.doc for more details).

If an error is encountered when copying an individual file, BACK$ does not return an immediate exception but continues with the next file. The total number of files encountered is returned in BKFILES. The total number of files successfully copied is returned in BKCOPIED. Thus, BKCOPIED should be compared with BKFILES to ensure that all the files in the source folder were successfully copied to the destination folder

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

COPYQ$	Copy Windows file using CopyFile()
COPYX$	Copy Windows file
GXCOP$	Copy Windows file with optional progress bar