**FUP**

**(FILE UTILITY PROGRAM)**

**INTRODUCTION**

* FUP (File Utility Program) is a component of the standard software package for the Tandem Non-Stop kernel.
* FUP software is designed to help you manage disk files, non-disk devices (printers, terminals, tape drivers) and processes (running programs) on the Tandem system.
* The File Utility Program is used for **file management**. You can use FUP to create, display and duplicate files, load data into files, alter files characteristics and purge files.
* Enscribe files are created using FUP.
* FUP supports the following types of Enscribe disk files –
* Key Sequenced File
* Entry Sequenced File
* Relative File
* Unstructured File (including text files)
* FUP is used to perform Operations on disk files like –
* Manage disk files and processes.
* Create file, display and duplicate files.
* Load data into files.
* Modify File attributes.
* Purge files.
* Alter File characteristics.
* The File name format is –

**[ $VOL.SUBVOL.FILENAME ]** where,

* **VOL** is volume name (up-to 7 alphanumeric characters)
* **SUBVOL** is sub-volume name (up-to 8 alphanumeric char.)
* **FILENAME** (up-to 8 alphanumeric characters)
* TANDEM supports 4 types of Files-
* Disk Files
* Device Files
* Process Files
* $Receive Files
* **$RECEIVE** :-
* This is a logical file associated with every process.
* Used for inter process communication.
* Used to receive system messages.
* Used to receive params passed by the command.
* Every file is associated with Owner & Security.

(OWNER- is the one who creates it; SECURITY- is defined as “RWEP”)

* FUP has a limitation that the command line cannot exceed 132 characters. To use longer ANSI names, FUP users need to –

1. Use MXCI SHOWDDL or MXCI SHOWLABEL to get the guardian names.
2. Run FUP on one or more of the Guardian names.

* **STARTING A FUP PROCESS** –

Access to FUP is available through TACL, the standard command interface in Guardian environment.

1. Enter FUP, followed by a FUP command, at the TACL prompt.

(For less number of FUP commands)

1. Enter FUP. Then use FUP commands interactively.

(Through this first you enter FUP command prompt and then write commands without writing FUP in the beginning of every command. This saves time.)

* To **TERMINATE** a FUP process after it has started, use “**CTRL-Y**”.
* To **INTERRUPT** a FUP process, press the “**BREAK**” key.
* **Wild-Card Option** :-
* You can use asterisks (\*) or question marks (?) as wild-card characters to help specify files. The asterisk can represent from zero through eight unspecified characters in the position you place it. The question mark represents only one character in the position you place it.
* The only FUP commands that allow you to use wild-card characters to specify a volume name are VOLS, SUBVOLS, INFO, FILENAMES and FILES.
* You cannot use more than 8 characters in any portion of the file name.(volume, sub-volume or file identifier)
* E.g.- To specify all the file IDs that begin with MY, are followed by one character, and end with ILE on the current default volume and sub-volume: “ MY?ILE “
* E.g.- To specify all the files in the sub-volume MYSVOL on the current default volume: “ MYSVOL.\* “
* E.g.- To specify all the File IDs that begin with MY on the current default volume and sub-volume: “ MY\* “.
* Difference between FUP COPY and FUP DUP:-
* FUP COPY**:**
* **FUP COPY err1, err2**

(This command will copy file err1 in err2 only when file err2 is already present/exists in system)

* **FUP COPY err1, err2, count3**

(Using this command we can copy portion of the file/doc into another file)

* FUP DUP:
* **FUP DUP err1, err2**

(This command will duplicate file er1 in err2 only when file err2 was earlier not present/did not exist in the system. This will duplicate entire file as it is.)

* **File Codes** :-
* 1300 – Audit Files.
* 1729 - Pak Files.
* 100 – Object File.
* 101 – Tedit File.
* **File Formats:**

**Format 1 -> less than 2GB**

Maximum partition size = 2 GB – 1 KB

Max Partitioned File size = 4 GB - 4KB

Max Un-partitioned File size = 2 GB – 1 KB

Maximum block size = 4 KB

Maximum record size = [block – 24] (Entry-sequenced and Relative file)

Maximum record size = [block – 34] (Key Sequenced)

**Format 2 -> Greater than 2GB files**

Maximum partition size = 1024 GB

Max Partitioned File size = 1024 GB

Max Un-Partitioned File size = 1024 GB

Maximum block size = 4 KB

Maximum record size = [block – 48] (Entry-sequenced and Relative file)

Maximum record size = [block – 56] (Key Sequenced)

* **AUDITED FILE :-**

In a system with the TMF, any database file can be designated as an audited file. To help maintain database consistency, TMF audits all transactions involving files designated as audited files. Given transaction's changes are either committed or aborted and roll back.

**FUP COMMANDS**

* **FUP**

(This command will bring the FUP prompt. You can write the Commands directly on the FUP prompt.)

* **FUP $\*.\*.\***

(This command will show all the files available in all the volumes and sub-volumes.)

* **FUP info t?st**

(This command will search all the files starting with character “t” whose second character is not known but ends with “st”)

* **FUP info test, detail**

(This command will display you file details e.g. file type, database, format etc.)

* **FUP DUP Test, $dasd.ancm2.\***

(This command will create a duplicate file with the same name i.e. “Test” in the volume and sub-volume “$dasd.ancm2”)

(FUP DUP = Command

Test = File Name

$dasd = Volume

Ancm2 = Sub-volume)

* **FUP DUP Test, $dasd.ancm2.\* , purge**

(FORCE Copy, if the file exists in the volume and sub-volume $dasd.ancm2 then it will delete it first and then will copy the file with the name “Test”)

* **FUP DUP TEST, TESTBACKUP**

(This command will copy/backup the existing test file with another name Test backup)

* **FUP INFO** **<filename>**

(Display basic information about a file)

* **FUP INFO <filename>, STATISTICS**

(Displays the number of record in the file)

* **FUP INFO <filename>, EXTENTS**

(Gives extent allocation of the file)

* **FUP DUP <filename>, <new filename>,SOURCEDATE**

(This command duplicates the file and records, retaining original date and time)

* **FUP LISTOPENS <filename>**

(The processes that are using this file can be listed using this command)

* **FUP COPY <filename>,,a,share**

(Display/Copy the records of the open file)

* 1. ASCII
  2. Binary h- Hex
* **FUP RENAME <filename1>, <filename2>**

(This command is used to rename a file)

* **FUP PURGEDATA <filename>**

(This command Deletes records from a file)

* **FUP PURGE <filename>**

(Deletes a file)

* **FUP <SUBVOLS>**

(Displays sub volumes in a volume)

* **FUP SECURE <filename>,\*\*\*\***

(This command is used to change the security on a file)

* **FUP LOAD <filename1>,<filename2>**

(Loads data into a structured disk file without affecting any associated-alternate key files.

Data in the file being loaded is overwritten. This command applies only to enscribe files)

* **FUP CREATE <filename>**

(This command creates a disk file with the current file-creation attributes defined by the SET command)

* **!**

(Executes an existing command again)

Syntax: **! [ -*num* | *num* | *string* | "*quoted*" ]**

Where,

* –num: executes a command that appears before the current command. For example use !-3 to execute the third command prior to the current one.
* num: It is the number of a command line. For example, use !2 to execute the second command of the current FUP session.
* String: It is the first character or characters of a previous command. For example, use !DUP$ to execute the most recent DUP command that begins with a volume name.
* Quoted: It is a string enclosed in either single or double quotation marks. For example, use !“\MAUI” to execute the most recent command that referenced the system \MAUI.

(If you use the ! command without a number or text string, FUP executes the last command you entered again)

* **?**

(Displays a specific command)

Syntax: **? [ -num | num | string | "quoted" ]**

(To display the last command you entered, use the ? command without a number or a text string)

* **ALLOCATE**

(Allocates file extents for a disk file. This command applies only to Enscribe files)

Syntax: **ALLOCATE fileset-list , num-extents [ , PARTONLY ]**

where,

* Fileset-list: is a list of disk files for which extents are to be allocated. You can specify qualifies-fileset for this fileset-list.
* Num-extents: It is the total number of extents to be allocated to the file. For non-partitioned Disk Process 2 (DP2) files and key-sequenced partitioned files, specify num-extents as a value from 1 through maximum-extents. The default value for maximum-extents is 16.
* Partonly: allocates extents to any primary and secondary partitions of partitioned files in file-set list.
* **ALTER**

(Changes some characteristics of an Enscribe disk file’s label. This command only affects the file label. It does not create or purge files, and does not insert, delete, or move records. It applies only to Enscribe files)

Syntax: **ALTER filename { , alter-option }...**

Where,

* Filename: It is the name of the file that you want to alter. You cannot use wild-card characters in filename or specify qualified-fileset for it.
* Alter-options: Names the file characteristic you want altered. Available options depends on the file type:-

1. For files with alternate-key fields:

* ALTFILE ( key-file-number , filename )
* ALTKEY (key-specifier { , altkey-param }... )
* DELALTFILE key-file-number
* DELALTKEY key-specifier

1. For unstructured files:

* BUFFERSIZE unstructured-buffer-size

Guidelines:

* To receive the current file attributes for any file you want to alter use FUP info command:

-INFO filename , DETAIL

* To alter a file, you must have both read and write access to it.

Examples:

* To assign file code 10 to MYFILE1:

-ALTER MYFILE1, CODE 10

* To delete the alternate key "ab" from MYFILE5:

-ALTER MYFILE5, DELALTKEY "ab", DELALTFILE 0

* To assign the alternate-key file MYFILE4 to MYFILE3 and give it key file number 2:

-ALTER MYFILE3, ALTFILE (2 , MYFILE4)

* **CREATE**

(Creates a disk file with the current file-creation attributes defined by the set command. To override the current file-creation attributes defined by the set command, include create-param in the CREATE command)

Syntax: **CREATE *filename* [ , *create-param* ] ...**

where,

* Filename: It is the name of the file to be created.
* Create-param: overrides the current file-creation attribute setting that corresponds to this creation. The *create-param* used in a CREATE command does not change the file-creation attributes defined by the set command.
* Using the CREATE command, you can create:
* 128 partitions
* Blocklength up to 32 KB
* Recordlength up to 27,648
* Primary keylength up to 2048
* Primary keyoffset up to 27,647
* Alternate keyoffset up to 27,647
* **EXIT**

(Stops the current FUP process and returns to the command interpreter)

* To run the EXIT command, you can **enter E or EXIT**.
* The FUP process terminates when FUP reads the end-of-file (EOF) mark of the input file that you specified in your command to run FUP.
* **GIVE**

(Changes the owner of a file. This command applies only to Enscribe files. Only the current owner of the file (or the super ID, (255,255)) can execute the GIVE command for the file)

Syntax: GIVE fileset-list ,

{groupnum , usernum | groupname.username }

Example:

* To give ownership of all files in the current default subvolume to the user with used ID 8,1:

-**GIVE \*, 8,1**

* To give the files PROG1, PROG2, and LIB in the subvolume $WORK.ORG to the user whose user ID is MANUALS.MARTIN:

-**GIVE($WORK.ORG.PROG\*,$WORK.ORG.LIB), MANUALS.MARTIN**

* **HELP**

(Lists the syntax of the FUP commands)

Syntax: **HELP [ / OUT listfile / ] [ command | ALL [, SYNTAX ] |NEWS ]**

Examples:

* To display the names of all FUP commands, enter HELP ALL(or HELP):

-**HELP ALL**

* To write the syntax for all the FUP commands to the file MYHELP:

-**HELP /OUT MYHELP/ ALL,SYNTAX**