

ALTEFILE

Version 302

Users Manual

17. June 2022



Trademarks or Service Marks

The following are trademarks or service marks of Tandem Computers Incorporated:

Atalla, Challenge/Response, Enform, Expand, Guardian, Guardiango, Inspect, Multilan, NonStop, TACL, Tandem.

All brand names and product names are trademarks or registered trademarks of their respective companies.

The following are trademarks or service marks of *GreenHouse GmbH*:

\$ARROW, \$AS, CRYSTAL, CURIOUS, FTPSERV-E, FUNCTRAC, MPWD, MPWD-L, PASSYNC, SECMAN, SECOM, GSTK, SSTK.

The following are trademarks or service marks of Jelinek EDV:

SECMAN

Copyright

Copyright © 2022 by *GreenHouse GmbH*. All rights reserved. No part of this document may be reproduced in any form, including photo copying or translation to another language, without prior written consent of *GreenHouse GmbH*.

Printed in Germany.

Please Comment

If you have questions or problems concerning the content of this document, please let me know! Send your comments to:

GreenHouse GmbH

Heinrichstraße 12

D-45711 Datteln/Horneburg

Germany

Phone +49 (0)2363 731004

Fax +49 (0)2363 66106

E-Mail: Info@GreenHouse.de

Internet: www.GreenHouse.de



Purpose

This small tool is developed to make the distribution of the GreenHouse products easier. Instead of keeping track where which alternate key entries have to be adjusted, I simply can run this small program - and it does it all.

Normally when a structured file is moved from one disk to another or from one system to another, alternate key files are also moved. What changes is the system name, volume name, or subvolume name (location) or all of these. The file name normally does NOT change.

The 'head file' needs to know where its alternate key files are. Each AK file has an entry in the head file. Changing the location of the AK files requires a change of the entry in the head file. The common way in performing this task is to use the FUP ALTER command. This is no big deal, but it requires that you KNOW about the files to be changed.

ALTFILE performs this task automatically: It allows you to change the AK entries in the head files to a new location.

Command syntax

```
[run] ALTFILE [<template>[,<default>[,STRIPMYNODENAME]]]
```

where

template	is a file name template, describing the head files. Extended wild cards are supported. When missing, the users current default location is used.
default	is the new location ([\system.[$\$$ vol[.subvol]]]) to which the AK entries in the head file have to be changed to. Wildcards are NOT supported! When missing, the head files current default location is used.
STRIPMYNODENAME	causes ALTFILE to delete the node name from the AK entries, when they point to the system, where ALTFILE is running on.

Examples

ALTFILE	Changes all AK entries in all structured files of the current subvol to the current subvol.
ALTFILE *	Does the same as ALTER without any parameter
ALTFILE *.*	Changes all AK entries in all structured files of all subvols, to the files \$vol.subvol.
ALTFILE EXEC*	Changes all AK entries in all structured files of the current subvol where the file name begins with EXEC, to the current subvol.
ALTFILKE , \$GHS1.SECOM	Alters all AK entries in all structured files of the current subvol to \$GHS1.SECOM.*
ALTFILE EXEC*, \$GHS1.SECOM	Alters all AK entries in all structured files of the current subvol where the file name begins with EXEC, to the subvol \$GHS1.SECOM.*
ALTFILKE \$GHS1.SECOM*.EXEC*, \$GHS2.SECOM	alters all AK entries in all structured files matching the template \$GHS1.SECOM*.EXEC*, to the subvol \$GHS2.SECOM

Usage

When you moved a subvol, holding head files as well as corresponding AK files, to a new location, you need to change the AK entries in the head files as well.

With ALTFILE, this task is an easy one. Simply type:

ALTFILE

and all entries in the head files are changed to the head files \$vol.subvol location.

Node Name conventions

Node names of alternate key file entries are deleted, when the keyword STRIPMYNODENAME is given, and when they point to the system where ALTFILE is running on.

Security

ALTFILE is a non PRIV TAL program, using GUARDIAN Cxx and Dxx procedure calls. The security of the file is uncritical, because a successful execution can only be performed when the ALTFILE-user has read/write access to the files in question.

A good GUARDIAN security string is "OOAO", where the owner is SUPER.SUPER, or the equivalent of a SAFEGUARD ACL.

Real live example

1. FUP INFO DETAIL of the file EXECMGMT:

```
$GHS1.ALTFILE.EXECMGMT          3 Apr 1996, 18:17
ENSCRIBE
TYPE R
CODE 18248
EXT ( 2 PAGES, 2 PAGES )
REC 104
BLOCK 4096
ALTKEY ( "FU", FILE 0, KEYOFF 70, KEYLEN 32 )
ALTKEY ( "MA", FILE 1, KEYOFF 0, KEYLEN 32 )
ALTKEY ( "US", FILE 2, KEYOFF 36, KEYLEN 32 )
ALTFILE ( 0, $GHS1.TEST.ALT1 )          <-----
ALTFILE ( 1, $GHS1.TEST.ALT2 )          <-----
ALTFILE ( 2, $GHS1.TEST.ALT3 )          <-----
MAXEXTENTS 16
OWNER 100,5
SECURITY (RWEPP): 0000
DATA MODIF: 3 Apr 1996, 15:35
CREATION DATE: 3 Apr 1996, 15:35
LAST OPEN: 3 Apr 1996, 18:17
EOF: 0 (0.0% USED)
FILE LABEL: 312 (7.6% USED)
EXTENTS ALLOCATED: 0
```

2. ALTFILE command

```
$GHS1 ALTFILE 281> altfile          <----- (default)
ALTFILE (100) - T9999D30_00 - (03Apr96) System \SEQUOIA
Copyright (c) GreenHouse Software & Consulting 1996
\SEQUOIA.$GHS1.ALTFILE.EXECMGMT modified
$GHS1 ALTFILE 282>
```



3. Result shown by FUP INFO:

```
$GHS1.ALTFILE.EXECMGMT                                3 Apr 1996, 18:26
ENSCRIBE
TYPE R
CODE 18248
EXT ( 2 PAGES, 2 PAGES )
REC 104
BLOCK 4096
ALTKEY ( "FU", FILE 0, KEYOFF 70, KEYLEN 32 )
ALTKEY ( "MA", FILE 1, KEYOFF 0, KEYLEN 32 )
ALTKEY ( "US", FILE 2, KEYOFF 36, KEYLEN 32 )
ALTFILE ( 0, $GHS1.ALTFILE.ALT1 ) <-----
ALTFILE ( 1, $GHS1.ALTFILE.ALT2 ) <-----
ALTFILE ( 2, $GHS1.ALTFILE.ALT3 ) <-----
MAXEXTENTS 16
OWNER 100,5
SECURITY (RWEF): OOOO
DATA MODIF: 3 Apr 1996, 15:35
CREATION DATE: 3 Apr 1996, 15:35
LAST OPEN: 3 Apr 1996, 18:24
EOF: 0 (0.0% USED)
FILE LABEL: 312 (7.6% USED)
EXTENTS ALLOCATED: 0
```