

# Delete Orphaned ACLs

## DelOACL

Version 310

31. July 2009

  
**greenHouse**  
Software & Consulting

Karl-Heinz Weber  
Heinrichstrasse 12  
D-45711 Datteln/Horneburg

**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 Software & Consulting*:

\$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 © 2009 by *GreenHouse Software & Consulting*. 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 Software & Consulting*.

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 Software & Consulting*

Karl-Heinz Weber

Heinrichstraße 12

D-45711 Datteln/Horneburg

Germany

Phone +49 (0)2363 72566

Fax +49 (0)2363 66106

Mobile +49 (0)172 23 18248

E-Mail: [Info@GreenHouse.de](mailto:Info@GreenHouse.de)

Internet: [www.GreenHouse.de](http://www.GreenHouse.de)

PGP fingerprint: 3A32 D90A D125 5418

1150 2484 6629 2DD2



# DelOACL

---

## Index

1.	Warning.....	3
2.	Introduction .....	3
2.1	Creating orphaned ACLs with SAFECOM .....	3
2.2	Creating orphaned ACLs by the system .....	3
3.	Product features .....	3
4.	List mode .....	4
5.	Clean-up mode .....	5
5.1	Clean-up ALL orphaned ACLs .....	5
5.2	Selective clean-up of orphaned ACLs .....	5
5.2.1	Prepare exception list.....	5
5.2.2	Edit exception list.....	6
5.2.3	Perform selective deletion .....	6
6.	Installation .....	7
7.	Run time security .....	7
8.	Support .....	7



# DelOACL

---

## 1. Warning

DelOACL uses PRIV code to physically access SAFEGUARD files.  
The SAFEGUARD internal security settings are NOT obeyed by DelOACL!  
Any user, allowed to execute DelOACL, can delete ALL orphaned ACLs!

## 2. Introduction

What is an Access Control List (ACL)?

An ACL is an access control list, created and controlled by SAFEGUARD. It defines access rights on an object.

What is an orphaned ACL?

An orphaned ACL is related to a nonexistent object.

How can it happen, that an ACL gets orphaned?

There are two ways in creating this type of ACL:

- Using SAFECOM
- By the system

### 2.1 Creating orphaned ACLs with SAFECOM

ACLs can be created with SAFECOM for Subvols, Devices and Subdevices, Processes and Subprocesses as well as Terminals even in case there is no such an object.

e.g. the command SAFECOM ADD TERMINAL \$HALLO is accepted by the system, even in case there is no such terminal (or device etc.). The same is true for adding an ACL for a subvol, e.g. \$GHS2.ABS: The existence of the volume is checked; the presence of the subvol is not checked.

### 2.2 Creating orphaned ACLs by the system

Only one ACL-type is deleted, when the related object is deleted: The ACL of a disk file. This is true as long as the ACL is not defined as persistent.

The ACLs of all other objects stay alive, even in case the objects are deleted, or no longer available.

This is true for:

- Disk file pattern
- Disk files with the persistent attribute set to ON
- Subvols
- Processes and Subprocesses
- Devices and Subdevices
- Terminals

SAFEGUARD does not have a function to display, and optionally delete orphaned ACLs.

To close this functionality hole, DelOACL was designed, and implemented by GreenHouse.

## 3. Product features

- DelOACL is able to display, and optionally delete, orphaned ACLs.
- All deleted ACLs are saved in an EDIT type file in command mode: This allows an easy re-implementation of deleted ACLs.
- The DELETE mode supports the definition of ACLs which have to be kept.
- No installation required.
- Delivered in code 100,700 and 800.
- DelOACL is supported FreeWare from GreenHouse.

# DelOACL

---

## 4. List mode

DelOACL can list all ACLs, where the object no longer exists.

The command syntax is:

```
RUN DELOACL[/OUT <listfile>/] [LISTONLY] [!]
```

where

<listfile>	defines the file, to which the output is sent. This entry is optional. When missing, the home terminal is used. When present, and the file does not exist, it is created as an EDIT type file. When present, and the file does exist, it is used.
LISTONLY	optional keyword
!	Exclamation mark. When present causes DelOACL to erase a possible existing OUT file.

A typical output to the terminal looks like this:

```
$GHS1 DELOACL 122> deloacl listonly
DelOACL(202) - T7172H06 - (06Jul2009)   System \GINKGO, running NSK H06.12
Copyright (c) GreenHouse Software & Consulting 2009
DiskFile  $SYSTEM.LISTLIB.LIB (persistent)
SubVol    $GHS2.HALLO
SubVol    $GHS2.SEI606
SubVol    $GHS1.HORST
DiskFile  $GHS1.MYSFGSPI.LOGON (persistent)
DiskFile  $GHS1.OLDSECOM.ABC (persistent)
DiskFile  $GHS1.SECOM600.ABC (persistent)
DiskFile  $GHS1.SECOMO.ABC (persistent)
Device    $GHI
Device    $HALLO
Process   $ABC
Process   $ABCD
Process   $ABCDE
Process   $CMON
Process   $HALLO
Process   $ZTC00
SubDevice $TAPE0.#ASDF
SubProcess $ABC.#DEF
SubProcess $ABC.#GHS
SubProcess $ABCD.#A1
SubProcess $HALLO.#HALLO
SubProcess $ICH.#DU
SubProcess $ICH.#ICH
Terminal  $A.#B
Terminal  $A345678.#A234567
Terminal  $HALLO
Terminal  $TERMA
Pattern   $GHS3.*X*.*Y*Y.
$GHS1 DELOACL 123>
```

The output can be directed to any OUT type file.

# DelOACL

---

## 5. Clean-up mode

The main intention of DelOACL is to delete orphaned ACLs from the SAFEGUARD system. Two modes are available:

1. Cleaning-up ALL orphaned ACLs
2. Keeping specific orphaned ACLs intact depending on a list, while cleaning up the others.

All deleted ACLs are saved in an EDIT type file.

The file name can be assigned by the ASSIGN ACLFILE <file> statement. In case the assign is missing, the file is named: ACLFILE.

The contents of the ACL file is cleaned-up by DelOACL to allow its immediate usage.

### 5.1 Clean-up ALL orphaned ACLs

To clean-up ALL orphaned ACLs, execute DelOACL with this command:

```
DELOACL [/OUT <listfile>/] DELETE [!]
```

where

**<listfile>** defines the file, to which the output is sent.  
This entry is optional.  
When missing, the home terminal is used.  
When present, and the file does not exist, it is created as an EDIT type file.  
When present and the file does exist, it is used.

**DELETE** required keyword

**!** Exclamation mark.  
When present causes DelOACL to erase a possible existing OUT file.

All deleted ACLs are saved in a file. This file can be assigned before DELOACL is started with this TACL ASSIGN command:

```
ASSIGN ACLFILE <aclfile>
```

When this assign is missing, a file named ACLFILE is created/used in the user's current location.

### 5.2 Selective clean-up of orphaned ACLs

It does make sense to exclude some ACLs from being deleted, such as persistent ACLs on disk files, or subvols.

To perform this task, follow these few steps:

#### 5.2.1 Prepare exception list

To prepare the exception list, execute DELOACL with this command:

```
DELOACL /OUT <listfile>/ [LISTONLY] [!]
```

This creates/uses a file and writes all orphaned ACLs into it.

# DelOACL

---

## 5.2.2 Edit exception list

Edit this file, just created by DELOACL.

Delete all entries, where the ACL has to be deleted.

**All entries, mentioned in the list, are NOT touched by DelOACL.**

Individual entries can be added as well.

An entry does have this structure:

Objecttype	File Name
------------	-----------

where

**Objecttype** is one of these:

- PATTERN
- DISKFILE
- SUBVOL
- VOLUME
- PROCESS
- SUBPROCESS
- DEVICE
- SUBDEVICE
- TERMINAL

The **File Name** structure depends on the object type:

- PATTERN        \$vol.subvol.file, where each name level may contain template characters
- DISKFILE       \$vol.subvol.file
- SUBVOL         \$ vol.subvol
- VOLUME         \$vol
- PROCESS        \$process
- SUBPROCESS    \$process.#subprocess
- DEVICE         \$device
- SUBDEVICE     \$device.#subdevice
- TERMINAL       \$terminal or \$terminal.#line

- Lines, beginning with an unknown object type, are skipped.
- The object type and file name have to be separated by at least one blank.
- Entries are NOT case sensitive.
- File names, except PATTERN names, may contain wildcards, such as: \$GHS1.SECOM\*.\*. This addresses all ACLs, related to disk files, matching the given template.
- The use of PATTERN does not support of wildcards.  
e.g. the pattern \$A\*.B\*.C\* addresses exactly this entry, not entries, matching the pattern, such as \$AA.BB.C\*

## 5.2.3 Perform selective deletion

To finally perform the selective deletion, execute DelOACL with this command:

```
DELOACL/IN <listfile>[,OUT <outfile>]/DELETE [!]
```



# DelOACL

---

All deleted ACLs are saved in a file. This file can be assigned before DelOACL is started with this TACL command:

```
ASSIGN ACLFILE <aclfile>
```

When this assign is missing, a file named ACLFILE is created/used in the user's current location.

The exclamation mark directs DelOACL to erase the OUT-file as well as the ACLFILE.

## 6. Installation

Select the correct file code for your system:

- 100 = K-, S- and Itanium systems
- 700 = K- und S-Systems
- 800 = Itanium systems

and upload the program file in binary mode into a location of your choice.

- Secure it to "OOAO"
- Set the owner to SUPER.SUPER

DelOACL uses PRIV code.

To make it work, the user has to be SUPER.SUPER.

To allow non SUPER.SUPER users the use of DelOACL, SUPER.SUPER has to license the object file:

```
FUP LICENSE DELOACL
```

**Doing so allows every user with execution access rights on DelOACL to delete orphaned ACLs!**

## 7. Run time security

DelOACL uses PRIV code to get physical access to some SAFEGUARD files. It always uses SUPER.SUPER capability to access the files, and to perform SAFEGUARD operations.

**The access rights of the DelOACL user in SAFEGUARD are NOT obeyed!**

## 8. Support

DelOACL is supported FreeWare.

In case you stumble into problems, or do have questions, remarks, or requests for enhancement, please contact us at: [Info@GreenHouse.de](mailto:Info@GreenHouse.de).

Please report any error and/or request for enhancements to:

*GreenHouse Software & Consulting*

Karl-Heinz Weber

Heinrichstraße 12

D-45711 Datteln/Horneburg

Germany

Phone +49 (0)2363 72566

Fax +49 (0)2363 66106

Mobile +49 (0)172 23 18248

E-Mail: [Carl.Weber@GreenHouse.de](mailto:Carl.Weber@GreenHouse.de)

Home page: <http://www.GreenHouse.de>

PGP fingerprint: 3A 32 D9 0A D1 25 54 18  
11 50 24 84 66 29 2D D2

