



CONSULTING

**IRRHFSU
HFS Unload Utility**

CHIRUG – November 2018



RSH Consulting – Robyn E. Gilchrist



RSH Consulting, Inc. is an IT security professional services firm established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. RSH's services include RACF security reviews and audits, initial implementation of new controls, enhancement and remediation of existing controls, and training.

- www.rshconsulting.com
- 617-969-9050

Robyn E. Gilchrist is a Senior RACF and CA ACF2 Consultant. She assists clients with conducting penetration and vulnerability tests to evaluate z/OS controls and with enhancing access controls. As a systems programmer and network engineer, Ms. Gilchrist has installed, configured, and maintained the z/OS Communications Server and WebSphere Application Server (WAS) for z/OS in Network Deployment (ND) mode with associated ACF2 and RACF controls. She has converted CPF-connected ACF2 databases to RRSF-connected RACF databases.

- 617-977-9090
- R.Gilchrist@rshconsulting.com
- www.linkedin.com/in/robyn-e-gilchrist/

RACF and z/OS are Trademarks of the International Business Machines Corporation

IRRHFSU



- Free utility provided by IBM/GitHub in both source code and executable form
 - Developed by Bruce Wells of IBM
- Extracts File Security Packets (FSPs) and status information (e.g., last used date) for files and directories in the z/OS Unix Hierarchical File System (HFS)
 - Optionally extracts security information on mounted file systems
- Generates a sequential text file with extracted information
- Format of IRRHFSU output records is very similar to IRRDBU00
- IRRHFSU output is suitable for browsing and processing with tools such as REXX, DFSORT, and ICETOOL
- IBM provides SQL for loading output into DB2

RACF and z/OS are Trademarks of the International Business Machines Corporation

File System Security



- zFS Dataset Access
- File System Mount
- File and Directory Security

zFS Dataset Access



- Control whether users can access files and directories contained in a zFS dataset

- General Resource
 - Class - FSACCESS (Note: RACLIST-REQUIRED)
 - Resource - *zfs-dsname* (e.g., OMVS.APPL.PAYROLL.ZFS)
 - Access levels - UPDATE - access zFS file

- Supersedes all other file security permissions and authorities (e.g., Superuser)

- RACF System-AUDITOR and ROAUDIT grant UPDATE access

- Access is permitted if there is no protecting profile

- Root File System (/) is excluded from FSACCESS check

zFS Dataset Access



```
RLIST FSACCESS ZFS.APP.V2R01 AUTH
```

```
CLASS      NAME
```

```
-----
```

```
FSACCESS   ZFS.APP.**
```

LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING
00	PSOFTGRP	NONE	UPDATE	NO

```
INSTALLATION DATA
```

```
-----
```

```
APPLICATION ZFS FILE SYSTEM
```

USER	ACCESS	ACCESS COUNT
*	UPDATE	

```
RLIST FSACCESS ZFS.PSOFT.V6 AUTH
```

```
CLASS      NAME
```

```
-----
```

```
FSACCESS   ZFS.PSOFT.**
```

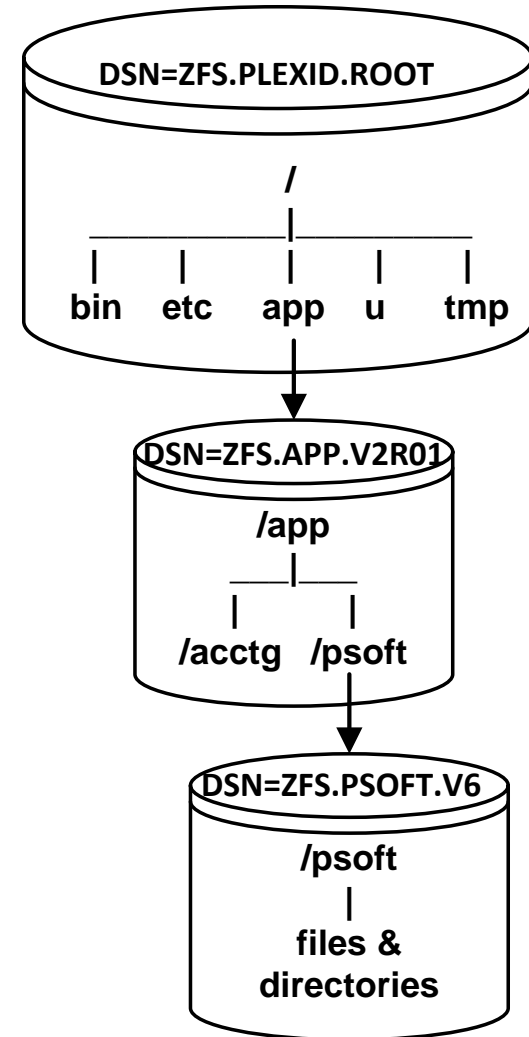
LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING
00	PSOFTGRP	NONE	NONE	NO

```
INSTALLATION DATA
```

```
-----
```

```
PEOPLESOFT ZFS FILE SYSTEM
```

USER	ACCESS	ACCESS COUNT
PSOFTSVR	UPDATE	
TECHSPTP	UPDATE	
PSOFTDEV	UPDATE	



File System Mount - Security Options



- MOUNT options determine whether users can write to a file system and whether setuid and security will be in effect

- Options
 - `MODE(RDWR | READ)` - Allow READWRITE or READONLY access
 - `SETUID | NOSETUID` - Honor setuid, setgid, APF, program control
 - `SECURITY | NOSECURITY` - Perform security checking

File System Mount - Security Options



```
====> df -v
```

```
/VERYSYB/usr/lpp/cobol (IGY420.HFS)          3576/4320      4294967279 Avail  
able
```

```
HFS, Read Only, Device:17, ACLS=Y
```

```
File System Owner : S0W1          Automove=Y      Client=N
```

```
Filetag : T=off  codeset=0
```

```
/S0W1          (OMVS.S0W1.SYSTEM.ZFS)      2564/2880      4294967283 Available
```

```
ZFS, Read/Write, Device:2, ACLS=Y
```

```
File System Owner : S0W1          Automove=N      Client=N
```

```
Filetag : T=off  codeset=0
```

```
Aggregate Name : OMVS.S0W1.SYSTEM.ZFS
```

```
/VERYSYB/usr/lpp/cicsts/cicsts42/samples (DFH420.SAMPLES.ZFS)      11408/17280  
Available
```

```
ZFS, Read/Write, Device:61, ACLS=Y, No SUID, Exported, No Security
```

```
File System Owner : S0W1          Automove=N      Client=N
```

```
Filetag : T=off  codeset=0
```

```
Aggregate Name : DFH420.SAMPLES.ZFS
```


Directory and File Security



- Each directory and file has its own File Security Packet (FSP)
 - Stored in directory's or file's parent directory in the file system
 - Created and deleted along with the directory or file

- FSP contains
 - Owner UID and Group GID
 - Permission bits - Owner/User, Group, and Other
 - Extended Attributes (e.g., APF)
 - Access Control List (ACL)
 - Audit bits

- Access checking is based on user's UNIX identity as established in the User Security Packet (USP)

File Security Packet



File Security Packet - Base Access Control List (ACL) entries

Owner uid	Group gid	Extended Attributes			setuid	setgid	sticky bit	Permissions						Auditing			Access Control List
		APF	Program Control	Run Share				Owner		Group		Other		Owner		Auditor	
								Read	Write	Execute	Read	Write	Execute	Read	Write	Execute	
chown	chgrp	extattr			chmod						chaudit			setfacl			

Permissions

- r read
- w write
- x execute (dir = search)
- T sticky bit
- t sticky bit + execute
- S set uid / gid
- s set uid / gid + execute

(file - sticky - load program from MVS)
 (dir - sticky - only Owner or UID 0 can delete)

Audit

- f failures
- s successes
- a all

All

- null

Extended Attributes (only applies to programs)

- a APF authorized
- p enable program control
- s run shared address space
- l load from shared library region

List File and Directory FSP - ls Command



```
$ ls -alEW rshtest
```

```
drwxr----x  fff--- --s-  4 RSH      SYS1      8192 Oct 15 10:31 .
drwxrwxrwt  fff--- --s-  3 BPXROOT  SYS1     24576 Oct 29  2011 ..
drwxr-xr-x+  fff--- --s-  2 RSH      SYS1      8192 Oct 29  2011 rshdirx
drwxrwxrwx  fff-s- --s-  2 BPXROOT  SYS1      8192 Oct 29  2011 rshtest2
-rwxr-xr-x+  fff--- --s-  1 RSH      SYS1    127910 Oct 15 10:31 sampfile
-rwxr-xrwx  fff--- --s-  1 6179050  SYS1    539070 Oct 29  2011 mastfile
-rwxr-xr-x+  fff--- --s-  1 RSH      9698211  12897 Oct 29  2011 testfile
```

"+" indicates presences of extended ACL

Extended Access Control List (ACL)



- Extension to base (original) file and directory permissions
- Activated by SETR CLASSACT(FSSEC)
- Maximum number of entries - 1024
- Supports inheritance of access controls - default ACLs

Extended ACL - getfacl



```
$ getfacl sampfile
#file:  sampfile
#owner: RSH
#group: SYS1
user::rwx
group::r-x
other::r-x
user:RLW:r-x
group:LEVEL1:--x
```

"-a" is the default

```
$ getfacl -adf rshdirx
#file:  rshdirx/
#owner: RSH
#group: SYS1
user::rwx
group::r-x
other::r-x
user:RLW:rwx
user:$OEDFLU:--x
group:LEVEL1:r-x
fdefault:group:LEVEL1:--x
fdefault:user:RLW:r-x
default:group:LEVEL1:--x
default:user:RLW:r-x
```

IRRHFSU Records



0900 HFS File Basic Data record

- One record per file or directory
- Contains File Owner, Group, Permissions, Attributes, Status, Auditing, File System DSNAME, Links

0901 HFS File Access record

- Each record is associated with a 0900 file or directory record
- One record per Extended ACL entry
- Contains User or Group and associated Permissions

0902 HFS File Default Access record

- Each record is associated with a 0900 directory record
- One record per Extended ACL File Default entry
- Contains User or Group and associated Permissions

0903 HFS Directory Default Access record

- Each record is associated with a 0900 directory record
- One record per Extended ACL Directory Default entry
- Contains User or Group and associated Permissions

0904 Mounted File System record (appear first in unload output)

- One record per mounted file system
- Contains DSNAME, Dataset Type (e.g., zFS), Mount Point, Mount Attributes - Mode, Security, SETUID

IRRHFSU Records



- Basic format - 0900-0903 records (position - contents):

1 - 4	Record Type (e.g., 0900)
6 - 1028	File/Directory (full path - /dir1/dir2/filex)
1030 - 1039	Inode (file serial number)
1041 - 4096	Record-specific fields (values or YES NO)

*** With DFSORT/ICETOOL, add 4 to starting position for variable blocked records

- Symbolic links show unresolved variable names (e.g., &SYSNAME/etc)
- Output records provide a RACF USERID and Group ID associated with each UID and GID
 - First RACF ID found is the one displayed
 - ❖ In z/OS 2.3, PARMLIB(BPXPRMxx) SUPERUSER user will be displayed for UID(0) USERID
 - No indication other RACF IDs may share same UID or GID
 - To find other RACF IDs, cross-reference with IRRDBU00 output
 - If UID or GID is not defined to RACF, the associated RACF ID field is blank
 - ❖ **Warning - if the default group of the user assigned the UID does not have a GID, the USERID field will misleadingly be blank**

Obtaining IRRHFSU



IBM RACF Downloads webpage now lives on GitHub

- <https://github.com/IBM/IBM-Z-zOS/tree/master/zOS-RACF/Downloads>

Branch: master | IBM-Z-zOS / zOS-RACF / Downloads /

Mark-Nelson-IBM Update readme.md | Latest commit 8ba9ebd on Aug 22

readme.md | Update readme.md | 2 months ago

readme.md

The RACF development team has numerous tools that can assist you in managing your RACF environment. These tools include:

1. **BPXCHECK**: A REXX exec which uses the RACF IRRXUTIL REXX interface to report the status of various UNIX- related settings in RACF.
2. **CDT2DYN**: A REXX exec which examine the contents of the current classes in the RACF static class descriptor table and creates the commands to put those installation-defined classes into the ynamc CDT.
3. **CUTPWHIS**: A utility which trims the orphaned passwords created by decreasing the SETROPTS password history value.
4. **DB2PRM**: A REXX exec which converts your RACF data set names table (ICHRDSNT) or RACF range table (ICHRNG) into a RACF PARMLIB member.
5. **DBSYNC**: REXX exec to find differences between two RACF databases and create commands to synchronize them. b
6. **ICHDEX01**: A sample RACF ICHDEX01 which ensures that the default for the encryption of passwords is not MASKED.
7. **IRRHFSU**: A utility which unloads your z/OS UNIX System Services hierarchical file system data (HFZ, TFS, or z/FS) in a manner which is complimentary to the RACF Data Base Unload Utility (IRRDBU00). d
8. **IRRXUTIL**: RACF IRRXUTIL Sample prgrams.

Obtaining IRRHFSU



FTP directory /eserver/z

ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/irrhfsu/

☆ Google Maps Interweb Registries IBM stuff Admin Research LinkedIn Protocols Z IBM Destination z - M

FTP directory /eserver/zseries/zos/racf/irrhfsu/ at public.dhe.ibm.com

[Up to higher level directory](#)

07/16/2013 12:00AM	127,308	HFSUnloadReadMe.pdf
12/17/2012 12:00AM	8,181	RACHFSLD.txt
12/17/2012 12:00AM	19,764	RACHFSTB.txt
02/20/2001 12:00AM	23,128	irrhfsu.c
08/14/2013 12:00AM	118,784	irrhfsu.o
08/14/2013 12:00AM	49,019	irrhfsu.txt

Obtaining IRRHFSU



- Items provided ...
 - HFSUnloadReadMe.pdf Documentation Manual
 - irrhfsu.txt C source code (text)
 - irrhfsu.o Executable object module (binary)
 - RACHFSLD.txt Sample DB2 Load Statements
 - RACHFSTB.txt Sample DB2 Table Statements

Installing IRRHFSU



■ To install as a Unix program

- PC - CMD window - upload file

```
cd \win-directory-where-irrhfsu.o-resides
FTP your-mainframe-ipaddress-or-dnsname
cd /your-home-directory
mkdir hfsu-subdirectory
cd hfsu-subdirectory
bin
put irrhfsu.o
quit
```

- z/OS - OMVS command - set 'r-x' permission to allow OWNER to execute

```
cd /your-home-directory
chmod 700 hfsu-subdirectory
cd hfsu-subdirectory
chmod 500 irrhfsu.o
```

- Optionally rename irrhfsu.o

```
mv irrhfsu.o hfsu
```

■ To install as an MVS program

- Allocate a PDSE library dataset

```
//jobname JOB job-card-parameters
//          EXEC PGM=IEFBR14
//HFSU      DD DSN=pdse.library,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(TRK,(10,5,3)),
//          UNIT=SYSDA,
//          DSNTYPE=LIBRARY,
//          DCB=(RECFM=U,
//          BLKSIZE=32760)
```

- Install as a Unix program and then copy to MVS PDSE library

```
cp irrhfsu.o "'/'pdse.library(IRRHFSU)'"
```

Executing IRRHFSU - Command Syntax



```
irrhfsu.o [-c] [-m] [-M] [-f output-file] path1 [path2 ... ]
```

path File or Directory (e.g., /u)

- includes all subdirectories

-c Clean up orphaned ACL entries

-m Create 0904 records along with other 0900-series records

- Creates records for all mounted file systems, not just that of target directory

-M Create only 0904 records (ignores any specified *path*)

-f Name of output Unix file or dataset (*//dsname*)

- Opens in append mode

Executing IRRHFSU - OMVS Shell



- Enter `irrhfsu.o` with options at the `===>` command line prompt

- Send output to Unix file (two options):

```
irrhfsu.o -f unix-output-filename path1 [path2 ... ]
```

```
irrhfsu.o path1 [path2 ... ] > unix-output-filename
```

- Using redirect ">" overwrites existing file's contents

- Send output to z/OS dataset

```
irrhfsu.o -f //OUTPUT.FILE.NAME path1 [path2 ... ]
```

- Send output to display

```
irrhfsu.o path1 [path2 ... ]
```

```
irrhfsu.o -M
```

- Example: `irrhfsu.o -m -f //RSH.HFS.ALL /`

Executing IRRHFSU - BPXBATCH Batch



```
//RSHHFSU JOB (0),'ROBERT HANSEL',NOTIFY=&SYSUID, ...
//STEP0010 EXEC PGM=BPXBATCH,
// PARM='PGM /u/RSH/hfsu/irrhfsu.o -m -f //RSH.HFSU.SYSA /'
//STDERR DD PATH='/u/RSH/hfsuerr',
// PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
// PATHMODE=SIRWXU
```

```
//RSHHFSU JOB (0),'ROBERT HANSEL',NOTIFY=&SYSUID, ...
//STEP0010 EXEC PGM=BPXBATCH
//STDPARM DD * (Note: Use NUM OFF)
PGM /u/RSH/hfsu/irrhfsu.o -m
-f //RSH.HFSU.SYSA /
//STDERR DD PATH='/u/RSH/hfsuerr',
// PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
// PATHMODE=SIRWXU
```

Note: Unix file and directory names and program options are case-sensitive

Executing IRRHFSU - MVS Program Batch



```
//RSHHFSU JOB (0),'ROBERT HANSEL',NOTIFY=&SYSUID, ...
//HFSUNLD EXEC PGM=IRRHFSU,PARM=( '/-m /' ) (Start parm with /)
//STEPLIB DD DSN=RSH.IRRHFSU.LOAD,DISP=SHR
//SYSPRINT DD DSN=RSH.HFSU.SYSA, (Alternative to -f)
//          DISP=( ,CATLG,DELETE),UNIT=TAPE,
//          DCB=(RECFM=VB,LRECL=4096,BLKSIZE=0)
```

- Note: (1) Unix file and directory names and program options are case-sensitive
(2) The / at the beginning of the PARM indicates to LE that the LE parameters have ended

Required Authority to Execute



- To unload a directory
 - FSACCESS *zfs-dataset* - UPDATE (except if System-AUDITOR or ROAUDIT)
... plus either of the following ...
 - READ and SEARCH (r-x) to the parent directory and all subdirectories, and SEARCH (--x) to all directories in the path
 - UID 0
 - FACILITY BPX.SUPERUSER - READ - execute 'su' command
 - RACF System-AUDITOR or ROAUDIT authority
 - UNIXPRIV SUPERUSER.FILESYS - READ
 - UNIXPRIV SUPERUSER.FILESYS.DIRSRCH - READ

- To create or write to the output file
 - Dataset - pre-allocated - UPDATE
 - Dataset - to be created - ALTER
 - FSACCESS *zfs-dataset* - UPDATE (except if System-AUDITOR or ROAUDIT)
... plus either of the following ...
 - Unix file already created - WRITE (-w-) to the file and SEARCH (--x) to all directories in the path
 - Unix file to be created - WRITE and SEARCH (-wx) to parent directory of the file and SEARCH (--x) to all directories in the path

Execution Tips



- Do NOT execute with a RACF ID having UAUDIT
 - Could generate an excessive number of SMF records causing an SMF buffer overflow

- Do NOT execute on a system where (a) UNIXMAP is inactive, (b) AIM Stage 2 or 3 has not been implemented, and (c) VLF is not caching UID and GID mappings
 - UID and GID lookups may cause performance problems
 - To improve performance, irrhfsu locally caches last 10 UID and GID mappings

- If sending output to a dataset ...
 - IRRHFSU creates dataset with RECFM=VB, LRECL=4096, and SPACE=(6144, (8,24))
 - Recommend pre-allocating dataset with substantial space

- Output is not sorted - sort 0900-0903 records by directory/file and record type
SORT FIELDS=(10,1023,CH,A,5,4,CH,A)

- Last access date of a directory is updated when unloaded by IRRHFSU

Execution Tips



- Analyzing Unix security for the entire z/OS environment requires unloading the file system on every individual z/OS system image (a.k.a., LPAR)
 - If the file system is shared by multiple images (BPXPRMxx parameter SYSPLEX=YES), and these images also share a RACF database, only need to unload the file system on one of the sharing systems
- Contents of unmounted file system datasets will not appear in the unload
- Contents of automounted file system datasets (e.g., user file systems) will not appear in the unload if not currently mounted
 - Can cause automounted file systems to be mounted temporarily by listing their contents with an ls command prior to executing IRRHFSU (e.g., ls -al /u/userid)
 - Can unload automounted file systems by specifying directory as IRRHFSU input
- RSH has found it helpful to ...
 - Convert YES/NO fields into Unix permission and attribute display symbols (e.g., r-x)
 - Add tabs between fields to facilitate importation into Excel

IRRHFSU Cleanup Option



- Execution option to remove obsolete entries from Extended ACLs (UID or GID with no matching RACF ID)

`irrhfsu -c ...`

- Unloads and then automatically deletes obsolete ACL entries
 - Assumes no matching USERID or Group ID means UID or GID is not assigned
- Authority to execute
 - FSACCESS *zfs-dataset* - UPDATE (except if System-AUDITOR or ROAUDIT)
... plus either of the following ...
 - UID 0
 - FACILITY BPX.SUPERUSER - READ - execute 'su' command
 - UNIXPRIV SUPERUSER.FILESYS.CHANGEPERMS - READ - plus any of the following
 - ❖ READ and SEARCH (r-x) to all directories to be cleaned and SEARCH (--x) to all directories in the path
 - ❖ RACF System-AUDITOR or ROAUDIT
 - ❖ UNIXPRIV SUPERUSER.FILESYS - READ
 - ❖ UNIXPRIV SUPERUSER.FILESYS.DIRSRCH - READ
 - Access to the output file and directories

Execution Errors



- If problems are encountered, browse file hfsuerr for error messages (specified in STDERR DD)
- See IRRHFSU documentation for most error messages
- If output dataset not pre-allocated with sufficient space, may get a B37 abend or fprint() error
- IRRHFSU will fail if it hits a corrupted directory

Sample ICETOOL Reports



- The following slides provide sample Batch Job JCL statements, TOOLIN DD ICETOOL commands, and xxxxCNTL DD DFSORT commands to generate reports from IRRHFSU output

- Sample ICETOOL reports include ...
 - Find OWNER and GROUP with no matching RACF ID
 - Find files with SETUID or SETGID
 - Find files and directories where OTHER has WRITE (-w-) permission

- The jobs have not been tested with SYNC SORT or any other sort programs and may require modifications to successfully run them with other products

Sample ICETOOL Report - Orphaned IDs



Find OWNER and GROUP with no matching RACF ID

```
//          JOB    ...
//S010URPT EXEC PGM=ICETOOL
//SYSOUT   DD    SYSOUT=*
//TOOLMSG  DD    SYSOUT=*
//DFSMSG   DD    SYSOUT=*
//HFSUFILE DD    DISP=SHR,DSN=irrhfsu.output.file
//SELDATA  DD    DSN=&&TEMP,DISP=(NEW,PASS,DELETE),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=VB,LRECL=4096)
//HFSUDID  DD    DSN=hfsudid.report.file,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE)
//TOOLIN   DD    *
           - see next slide -
//UDIDCNL  DD    *
           - see slide after next -
```

Sample ICETOOL Report - Orphaned IDs



Find OWNER and GROUP with no matching RACF ID

```
//TOOLIN DD *
SORT FROM(HFSUFILE) TO(SELDATA) USING(UDID)
DISPLAY FROM(SELDATA) LIST(HFSUDID) -
    PAGE TITLE('HFS OBJECTS WITH UNDEFINED UID OR GID') -
    DATE TIME -
    LINES(999) -
    BLANK -
    ON(1036,10,CH) HEADER('OWNER-UID') -
    ON(1046,8,CH)  HEADER('USERID') -
    ON(1054,10,CH) HEADER('GROUP-GID') -
    ON(1064,8,CH)  HEADER('GROUP') -
    ON(1112,3,CH)  HEADER('ACL') -
    ON(1115,3,CH)  HEADER('DDACL') -
    ON(1118,3,CH)  HEADER('FDACL') -
    ON(1072,10,CH) HEADER('CREATED') -
    ON(1082,10,CH) HEADER('LAST-MOD') -
    ON(1092,10,CH) HEADER('STATUS-CHGD') -
    ON(1102,10,CH) HEADER('LAST-ACC') -
    ON(1028,8,CH)  HEADER('FILETYPE') -
    ON(5,1023,CH)  HEADER('PATH+FILE/DIRECTORY-NAME')
```

Sample ICETOOL Report - Orphaned IDs



Find OWNER and GROUP with no matching RACF ID

```
//UDIDCNTL DD *
OPTION VLSHRT,VLSCMP,DYNALLOC=(3390,4)
INCLUDE COND=(5,4,CH,EQ,C'0900',AND,
              (1065,8,CH,EQ,C'      ',OR,
              1085,8,CH,EQ,C'      '))
INREC FIELDS=(1,4,10,1023,1045,8,1054,10,1065,8,1074,10,1085,8,
              1273,10,1333,10,1313,10,1293,10,1369,3,1374,3,1379,3)
SORT FIELDS=(5,1023,CH,A)
```


Sample ICETOOL Report - SETUID or SETGID



Find files with SETUID or SETGID

```
//          JOB    ...
//S010URPT EXEC PGM=ICETOOL
//SYSOUT   DD    SYSOUT=*
//TOOLMSG  DD    SYSOUT=*
//DFSMSG   DD    SYSOUT=*
//HFSUFILE DD    DISP=SHR,DSN=irrhfsu.output.file
//SELDATA  DD    DSN=&&TEMP,DISP=(NEW,PASS,DELETE),UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=VB,LRECL=4096)
//HFSUSET  DD    DSN=hfsuset.report.file,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE)
//TOOLIN   DD    *
//          - see next slide -
//USETCNTL DD    *
//          - see slide after next -
```

Sample ICETOOL Report - SETUID or SETGID



Find files with SETUID or SETGID

```
//TOOLIN DD *
SORT FROM(HFSUFILE) TO(SELDATA) USING(USET)
DISPLAY FROM(SELDATA) LIST(HFSUSET) -
    PAGE TITLE('HFS FILES WITH SETUID OR SETGID') -
    DATE TIME -
    LINES(999) -
    BLANK -
    ON(1036,8,CH)    HEADER('USERID') -
    ON(1044,10,CH)   HEADER('OWNER-UID') -
    ON(1054,3,CH)    HEADER('SETUID') -
    ON(1057,8,CH)    HEADER('GROUP') -
    ON(1065,10,CH)   HEADER('GROUP-GID') -
    ON(1075,3,CH)    HEADER('SETGID') -
    ON(1078,10,CH)   HEADER('CREATED') -
    ON(1088,10,CH)   HEADER('LAST-MOD') -
    ON(1098,10,CH)   HEADER('STATUS-CHGD') -
    ON(1108,10,CH)   HEADER('LAST-ACC') -
    ON(1028,8,CH)    HEADER('FILETYPE') -
    ON(5,1023,CH)    HEADER('PATH+FILE/DIRECTORY-NAME')
```

Sample ICETOOL Report - SETUID or SETGID



Find files with SETUID or SETGID

```
//USETCNTL DD *
OPTION VLSHRT,VLSCMP,DYNALLOC=(3390,4)
INCLUDE COND=(5,4,CH,EQ,C'0900',AND,
              (1094,3,CH,EQ,C'YES',OR,
              1099,3,CH,EQ,C'YES'),AND,
              1045,8,CH,EQ,C'FILE  ')
INREC FIELDS=(1,4,10,1023,1045,8,1065,8,1054,10,1094,3,1085,8,
              1074,10,1099,3,1273,10,1333,10,1313,10,1293,10)
SORT FIELDS=(5,1023,CH,A)
```

Sample ICETOOL Report - OTHER WRITE Access



Find files and directories where OTHER has WRITE (-w-) permission

```
//          JOB    ...
//S010URPT EXEC PGM=ICETOOL
//SYSOUT   DD    SYSOUT=*
//TOOLMSG  DD    SYSOUT=*
//DFSMSG   DD    SYSOUT=*
//HFSUFILE DD    DISP=SHR,DSN=RSH.HFSU.OUT
//SELDATA  DD    DSN=&&TEMP,DISP=(NEW,PASS,DELETE),UNIT=SYSDA,
//          SPACE=(CYL,(20,5),RLSE),DCB=(RECFM=VB,LRECL=4096)
//HFSUOTW  DD    DSN=RSH.HFSUOTW.REPORT.FILE,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(20,5),RLSE)
//TOOLIN   DD    *
- see next slide -
//UOTWCNTL DD    *
- see slide after next -
```

Sample ICETOOL Report - OTHER WRITE Access



Find files and directories where OTHER has WRITE (-w-) permission

```
//TOOLIN DD *
SORT FROM(HFSUFILE) TO(SELDATA) USING(UOTW)
DISPLAY FROM(SELDATA) LIST(HFSUOTW) -
    PAGE TITLE('HFS OBJECTS WITH OTHER WRITE PERMISSION') -
    DATE TIME -
    LINES(999) -
    BLANK -
    ON(1036,10,CH) HEADER('OWNER-UID') -
    ON(1046,8,CH)  HEADER('USERID') -
    ON(1054,10,CH) HEADER('GROUP-GID') -
    ON(1064,8,CH)  HEADER('GROUP') -
    ON(1112,3,CH)  HEADER('ACL') -
    ON(1115,3,CH)  HEADER('DDACL') -
    ON(1118,3,CH)  HEADER('FDACL') -
    ON(1072,10,CH) HEADER('CREATED') -
    ON(1082,10,CH) HEADER('LAST-MOD') -
    ON(1092,10,CH) HEADER('STATUS-CHGD') -
    ON(1102,10,CH) HEADER('LAST-ACC') -
    ON(1028,8,CH)  HEADER('FILETYPE') -
    ON(5,1023,CH)  HEADER('PATH+FILE/DIRECTORY-NAME')
```

Sample ICETOOL Report - OTHER WRITE Access



Find files and directories where OTHER has WRITE (-w-) permission

```
//UOTWCNTL DD *  
OPTION VLSHRT,VLSCMP,DYNALLOC=(3390,4)  
INCLUDE COND=(5,4,CH,EQ,C'0900',AND,  
              1144,3,CH,EQ,C'YES',AND,  
              (1045,8,CH,EQ,C'FILE',OR,  
              1045,8,CH,EQ,C'DIR'))  
INREC FIELDS=(1,4,10,1023,1045,8,1054,10,1065,8,1074,10,1085,8,  
              1273,10,1333,10,1313,10,1293,10,1369,3,1374,3,1379,3)  
SORT FIELDS=(5,1023,CH,A)
```