

ECFS – ECMWF User Archive

Computer user training course 2017

Carsten Maass

User Support

C.Maass@ecmwf.int



Content

- Introduction
- The ECFS client: a Unix-like interface
- ECFS User commands
- ECFS in scripts
- Recommendations
- Future plans

Introduction

- Since 1983, ECMWF has operated a large-scale Data Handling System (DHS), in which all users can store and retrieve data
- The Data Handling System consists of three main components:
 - IBM's High Performance Storage System (HPSS), used as the underlying archiving system in which data is kept
 - **MARS** - **M**eteorological **A**rchival and **R**etrieval **S**ystem
 - **GRIB and BUFR data**
 - About 166 Petabytes in 18 million files (~9.2 GB/file)
 - ~ 193 Terabytes added daily
 - **ECFS** – **E**CMWF **F**ile **S**torage system
 - **Any kind of data**
 - About 44.0 Petabytes of data in 260 million files (~169 MB/file)
 - ~ 40 Terabytes added daily

The ECFS client: a Unix-like interface (1/2)

- Files are mapped to a Unix-compatible directory tree
- Either absolute and relative pathnames can be used
- Concept of current ECFS working directories, analogous to the Unix current working directory
- Wildcard characters are supported for (rightmost) ECFS file element of a path but not for directories,
e.g. you cannot use `els ec:directory*/filename.out`
- The ECFS file size limit is 32 GB. Be aware that certain Unix systems (not at ECMWF) or software packages cannot handle files over 2 GB in size

The ECFS client: a Unix-like interface (2/2)

- But **this is not a UNIX file system**:
 - Files are migrated off to tape(s) behind the scenes
 - There are **overheads** when files are transferred to/from ECFS, unless file is on disk cache (small and recent data)
- ECFS commands:
 - **els, erm, ermdir, emkdir, ecd, epwd, echmod, echgrp, ecp, emv** (and **emove**), **ecat, etest, etouch, eumask** and
 - **ecfsdir, ecfs_status**
- Environment is set up for Korn-shell, Bash and C-shell users

Documentation & availability at ECMWF

- ECFS commands are available on all ECMWF platforms (ecgate and HPCF systems) except `ecfs_status` command for monitoring ECFS usage (available on ecgate only)
- Documentation is available at <https://software.ecmwf.int/wiki/display/UDOC/ECFS>
- ECFS man page:
`man ecfs`
- In addition there are man pages for each specific command e.g.:
`man els`

ECFS domains

- ECFS files are currently stored in two domains:

ec: and **ectmp:**

- **ec:** permanent domain where files are stored indefinitely.
This is the default domain.
- **ectmp:** temporary domain where files are stored for 90 days, after which they are automatically deleted. **Once a file has been automatically deleted it CANNOT be recovered.**

NB: Co-Operating states may ONLY use domain ectmp:

- The domain names **ec:** and **ectmp:** should be used with all ECFS commands to explicitly indicate which domain to use
- Note that, as an alternative, the **ectmp:** domain can be referenced by **ec:/TMP**, thus the following are equivalent:
 - **ec:/TMP/uid/newdir**
 - **ectmp:/uid/newdir**

User commands: Exploring the ECFS file system

- List ECFS files described by target:

`els [-l] [-1] [-a] [-d] [-R] <target>`

← Target should be prefixed by an ECFS domain either ec: or ectmp:

↑ To list subdirectories recursively.

els can time out for very large ECFS directory trees. (see ecfs_audit file)

- Change the current ECFS working directory for the specified ECFS domain:

`ecd <target>`

← Sets the value of the current working directory of the specified domain/directory

NB: Defaults to login name of user if target omitted

- Print name of the ECFS current working directory for the specified ECFS domain:

`epwd ec:`

or `epwd ectmp:`

← Display the current ECFS working directory for the relevant domain

Practical 1: Exploring the ECFS file system

- Try the following commands on ecgate:

```
epwd ec:
```

```
epwd ectmp:
```

- Use els to list all the files contained in both domains:

```
els -la ec:
```

```
els -la ectmp:
```

- Change the working directories and use els to list their contents:

```
ecd ec:/trx
```

```
epwd ec:
```

```
els -la ec:
```

```
ecd ec:
```

User commands: Transferring files between ECFS and client storage

Overwrite existing file unconditionally

Create a backup copy in Disaster Recovery System (DRS)
Use sparingly, for files impossible/expensive to recreate.

Do not overwrite if file exists.
Not an error (DEFAULT)

File's timestamp, group and permission will be kept

Either target or source should be prefixed by an ECFS domain (ec: or ectmp:)

ecp/emv [-e|n|o|u] [-b] [-p] <source> <target>

ecp/emv [-e|n|o|u] [-b] [-p] -F sourcelist <target>

Do not overwrite.
Treat as an error if attempted (Return 1)

Overwrite only if target is older than source. (Time standards differ on local workstations and servers).

Alternative syntax

NB: emv is similar to ecp but <source> files are removed after being transferred

Example: Transferring files between ECFS and client storage

```
> ecp $SCRATCH/my_file ectmp:Backup/Mar/ecfs_scratch_file
```

Note that ecp will automatically create missing directories in the target path.

```
> emv ectmp:ecfs_scratch_file $SCRATCH/my_file
```

Client storage
ECFS

Transferring files between ECFS and client: `ecp --parents`

With **--parents** the source pathname structure with its directory tree, is replicated into the target

```
> ecp subdir1/subdir2/foo ec:tgtdir/
```

The target pathname is "`ec:tgtdir/foo`"

Client storage
ECFS

```
> ecp --parents subdir1/subdir2/foo ec:tgtdir/
```

The target pathname is "`ec:tgtdir/subdir1/subdir2/foo`"

With version
2.2.3

Potentially very beneficial if fetching many files scattered across different directories and pathname structure needs to be preserved

Practical 2: ecp and emv

- Work in your \$SCRATCH

```
cd $SCRATCH
```

- Create a copy of the practicals directory in your \$SCRATCH

```
tar -xvf /perm/ecrain/trx/ecfs_practicals.tar
```

- Copy the files \$SCRATCH/ecfs_practicals/data/file*.out in ectmp:
- Move the file ectmp:file1.out in your \$SCRATCH

Client storage
ECFS

User commands: File deletion

`erm [-i] [-r] <target>`

 ↑ ↑
interactive recursive

Target should be prefixed by an ECFS domain either `ec:` or `ectmp:`
No client files are affected.

```
> erm ec:ecfs_scratch_file
```

`erm` will not ask for confirmation, unless `-i` is specified

```
> erm ec:test*
```

Files are removed from ECFS with a soft-delete: files will still be kept for currently 30 days during which it will be possible, on request, to un-delete any file that was deleted by mistake. **After that period any removal will become permanent.**

Please contact us if you have to remove large directory trees

Backup support

- **No automatic backup copy is made of ECFS data.** Specify the “-b” option on the ECFS commands (ecp, emv, ecfsdir) to request a backup copy to be made:

```
ecp -b myfile ec:essential_data
```

```
emv -b myfile ec:essential_data
```

```
ecfsdir -b $SCRATCH/results ec:essential_directory
```

- The existence of a backup copy will be indicated by a **b** as the first character of the line listing:

```
b-r--r----- 1 uid group 512 Nov 19 2003 essential_data
```

```
-rw-rw----- 1 uid group 512 Nov 19 2003 non_essential_data
```

- NOTE: Irrespective of the existence of backup copies: any ECFS files removed (deleted) by a user can only be recovered for a limited period of 30 days

User commands: creation and removal of directories

Creates all the non-existing parent directories first

Specifies the octal file permission mode to be used for new directories. If not present, the ECFS umask (027 by default) is applied

- Make directory:

`mkdir [-p] [-m octal_mode] <target>`

- Remove a specified empty directory:

`rmdir [-i] <target>`

interactive (request confirmation for each item)

```
> mkdir -p ectmp:DIR1/DIR2/DIR3
```

```
> rmdir ectmp:DIR1/DIR2/DIR3
```

Delete empty directories only

User commands: changing permissions

`chmod [-R] octal_mode <target>`

```
> chmod 640 ec:myecdir
```

- Change the current ECFS eumask: `eumask [<umask>]`

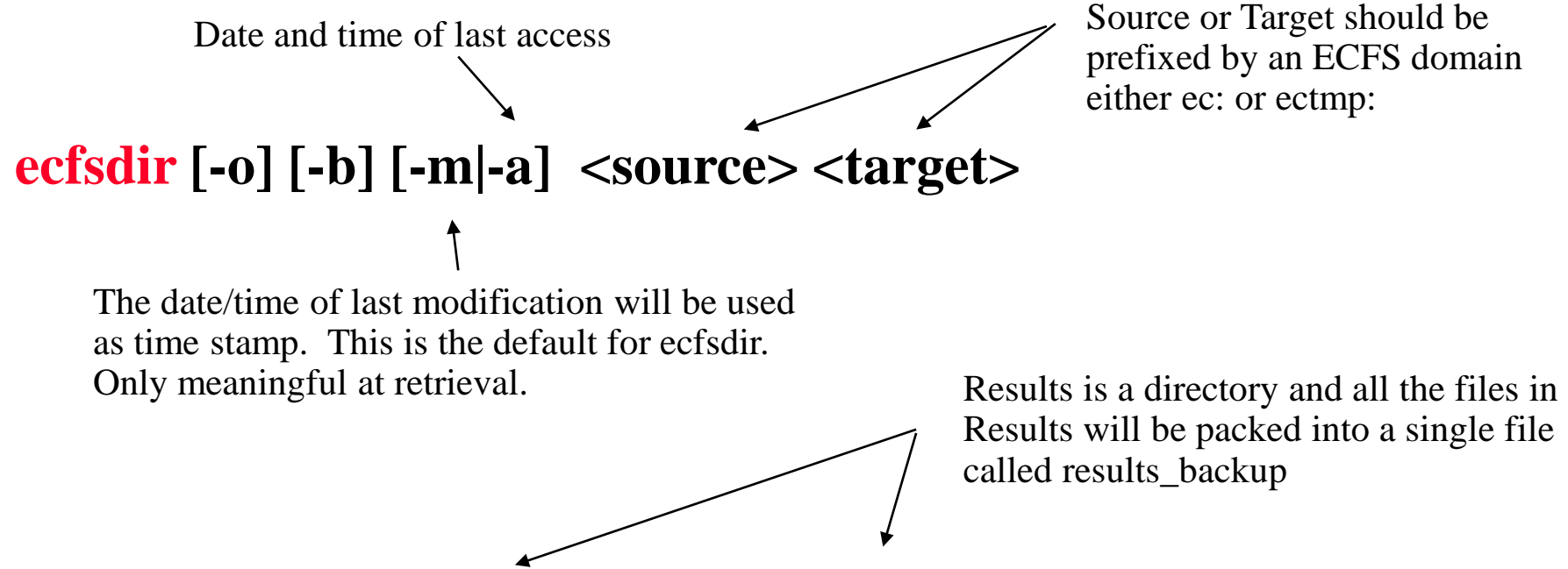
```
> eumask 022
```

← Only numerical values can be used as ECFS umasks. The default ECFS umask is set to 027.

- Change group of file(s): `chgrp group <target>`

```
> chgrp mysecgrp ec:/uid/*
```

User commands: save/retrieve a complete Unix directory as one ECFS file



```
> ecfsdir $SCRATCH/Results ectmp:Model/results_backup
```

Results directory saved

NB: ecfsdir uses cpio to “compact” the files

User commands: save or retrieve a complete Unix directory as one ECFS file

```
> cat $HOME/ECFS/data_1717.06Mar2011
```

```
Contents of the directory saved:
```

```
=====
```

```
./DIR1/DIR2/file1
```

```
./DIR1/DIR2/DIR3/file2
```

```
.
```

```
.
```

```
.
```

```
./DIRn/.../DIRm/filep
```

```
.
```

```
Name of the directory saved:
```

```
/scratch/ectrain/trx/Results
```

```
Ecfs backup in :
```

```
/trx/mp:Model/results_backup
```

```
Date : Fri Feb 3 12:19:04 GMT 2006
```

```
From : ecgate
```

This file is stored in \$HOME/ECFS to give you the list of files/directories saved. However, you can delete this file or move it (it is not needed when retrieving from ECFS).

Practical 3: ecfsdir

- Use ecfsdir to copy the content of the directory `$SCRATCH/ecfs_practicals/data` in `ec:mydata`

```
> ecfsdir $SCRATCH/ecfs_practicals/data ec:mydata
```



Faster than the equivalent ecp

- Check the content of your `$HOME/ECFS` (search for a file named `data_*20Feb2017`).

```
> cat $HOME/ECFS/data_* 20Feb2017
```

- Then retrieve `ec:mydata` in your `$SCRATCH/ecfs_practicals/mydata`

```
> ecfsdir ec:mydata $SCRATCH/ecfs_practicals/mydata
```

```
> cd $SCRATCH/ecfs_practicals/mydata
```

Client storage
ECFS

User commands: renaming/moving files within the same ECFS domain

Source and target should be prefixed by the same ECFS domain (ec: or ectmp:)

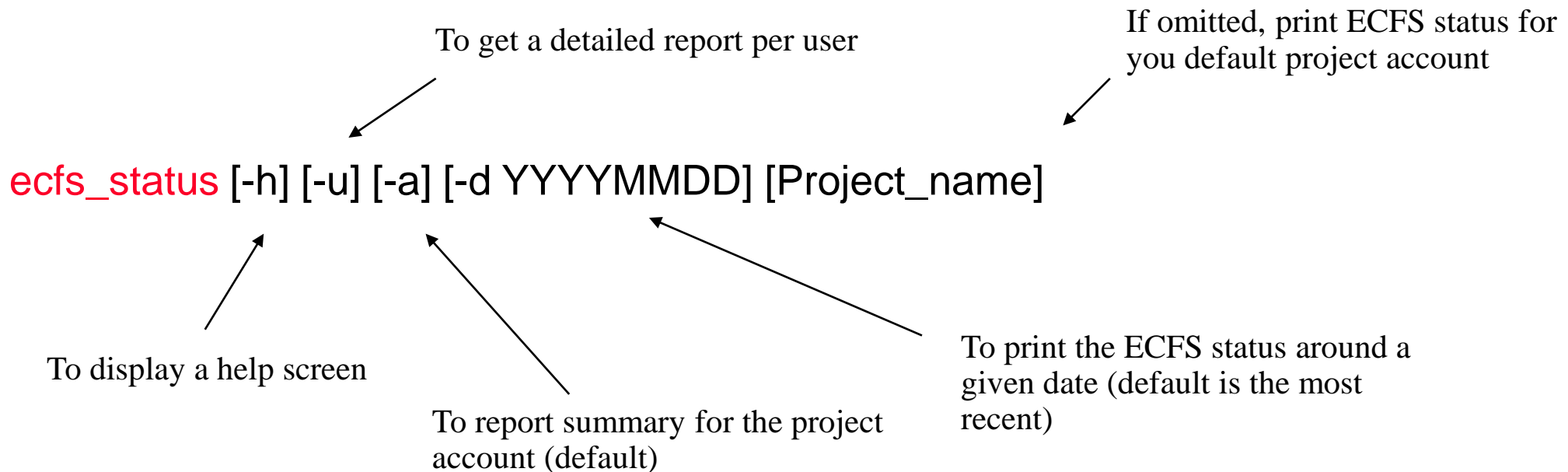
emove [-o|t|n|e] <source> <target>

```
> emove ectmp:ecfs_file ectmp:DIR1/ecfs_fileFeb06
```

- DIR1 must exist!
- Not possible to move data between **ec:** and **ectmp:** domains

User commands: usage monitoring

The `ecfs_status` command to be run on ecgate to get the most recent usage by project account



To get an overview on their ECFS usage, users can also refer to the audit files `ec:ecfs_audit` and/or `ectmp:ecfs_audit.tmp` which are created once per month and contain a complete list of a user's files in each ECFS domain

Examples: usage monitoring

Client storage
ECFS

- Running `ecfs_status` on `ecgate`:

```
> ecfs_status  
ECFS status on 20110210 for my_acct  
Account my_acct  Total: 64259322 MB - 1057024 files  Transfer previous month:  
                                                                3486719 MB - 23833 files  
Total: 64259322 MB - 1057024 files  Transfer previous month: 3486719 MB - 23833 files
```

- To read `ecfs_audit` or `ecfs_audit.tmp`, you need first to copy them locally (these two files don't exist for new accounts; they will be created after the first month)

```
> ecp ec:ecfs_audit $SCRATCH/ecfs_audit  
> cat $SCRATCH/ecfs_audit  
-- uid gid      size(bytes) creation  last_access path  today= 2011-02-14  
* trx ectrain   1945665 2005-12-16 2005-12-16 /trx/test1  
* trx ectrain   1305088 2005-12-16 2005-12-16 /trx/test2  
...  
Total files =20    megabytes = 116.864808082581  
total directories = 2 total files not accessed since 20040708 = 0
```

User commands: **copy file between a domain and STDIN/STDOUT**

ecat -s size [-e][-n][-o][-b] - domain:filename

ecat domain:filename -

```
> ecat ec:ecfs_audit -  
> -- uid gid      size(bytes) creation  last_access path  today= 2011-02-14  
* trx  ectrain    1945665 2005-12-16 2005-12-16 /trx/test1  
* trx  ectrain    1305088 2005-12-16 2005-12-16 /trx/test2  
...  
Total files =20    megabytes = 116.864808082581  
total directories = 2 total files not accessed since 20040708 = 0
```

ecat is not as resilient as other ECFS commands!

Client storage
ECFS

User commands: Check file attributes

etest [-{option}] <target>

Target should be prefixed by an ECFS domain either ec: or ectmp:

Options are a subset of Unix test.

```
> els -l ec:testfile  
-rw-r----- 1 trx   ectrain   29 Mar 4 09:36 testfile  
> etest -w ec:testfile && echo writable  
writable
```

ECFS within scripts

- Check existence of local copy before getting file from ECFS:

```
#!/bin/ksh
if [ ! -r $SCRATCH/file2.out ]; then
    ecp ec:file2.out $SCRATCH/.
fi
```

- Loop over ECFS directories to change mode

```
ECFSdir=ec:/$USER/TESTDIR-1
ECFSprefix=`dirname $ECFSdir`; dirs=`basename $ECFSdir`
while [ -n "$dirs" ]; do
    newdirs=""
    for dir in $dirs; do
        for name in `els -1 ${ECFSprefix}/${dir} | tr -d '/'`; do
            echmod 755 ${ECFSprefix}/${dir}/${name}
            newdirs="$newdirs ${dir}/${name}"
        done
    done
    dirs=$newdirs
done
```

ECFS in HPC batch jobs

Use ECMWF PBSpro custom directive to avoid that your job runs during ECFS system

```
#PBS -l EC_ecfs=1
```

Where # is the number of parallel ecfs requests that will be started by this job

Recommendations

- **Do not copy in/out the same files frequently.** Use temporary local disk space such as `$SCRATCH` to keep a local copy of these files (by default `ecp` will not overwrite a file if it exists; do not use the `-o` option in that case)
- **Create fewer large files rather than many small files** otherwise it can adversely affect performance of the entire system
 - Group together what belongs together using `ecfsdir` or `tar` and **only then** store into ECFS
 - Find reasonable balance
- Use file lists with `ecp/emv` as much as possible
- Think about who needs to read your files (permissions, `eumask`)
- Use `ectmp`: if files do NOT need to be kept for long periods
- Delete files which you do not need in `ec`:
- Never use ECFS commands in parallel jobs on HPCF

Future plans

- Implement retrieve of multiple files in optimal order by default
- Introduce concept of lifetime/expiry
- Enhance accounting
- Introduce quotas