

Use of the ECMWF computing systems

Dave MacLeod, University of Oxford

23 March 2016

This document can be found at gitHub: <https://github.com/doelcam/ECMWF-verification-suite.git>

Contents

Contents	2
1 Basics	3
1.1 Access	3
1.2 Copying data to a remote location	4
2 Data storage: MARS and ECFS	5
2.1 MARS	5
2.2 ECFS	6
3 SMS, CDP & XCDP	7
3.1 Gettting started with CDP	7
3.2 Starting a suite	8
3.3 Restarting a suite	9

Chapter 1

Basics

There is a lot of information available online.

User support are very helpful and friendly so if you are having problems, don't hesitate to contact them! For the UK this is Paul Dando (paul.dando at ecmwf.int); if he is unavailable the general user support email is advisory at ecmwf.int.

1.1 Access

You'll need to be registered as a user and have a security token to log in - more details can be found on the ecmwf website. You can connect to ECMWF via ssh, from the command-line:

```
ssh -X <uid>@ecaccess.ecmwf.int
```

where `<uid>` is your unique user ID. Once you have connected you will need to generate and input a temporary 8-digit code, by entering your pin into your security token. You are then able to choose where to log into, e.g. "cca" (the high performance computer) or "ecgate" (general purpose computing). An overview of computing systems is available.

1.2 Copying data to a remote location

There are several ways to copy data to a remote location from ECMWF. For example you can use scp (secure copy) whilst logged onto ecgate:

```
scp files remoteusername@remotelocation
```

If you are copying a lot of files, you may want to create a convenient environment variable. To do this add the following line to `~/.user_kshrc` on the ECMWF machine:

```
export rem=remoteusername@remotelocation
```

where remoteusername is username on the remote server. Then to copy, you just need to enter the following command on the ECMWF machine:

```
scp filelist $rem:directory
```

There are other ways to copy data available (e.g. via ftp). Note that if you have a large number of files, it will be more efficient to create an archive first and copy that (e.g. `tar -cfv newarchivename.tar files`, use `man tar` for details).

Chapter 2

Data storage: MARS and ECFS

At ECMWF there are two separate archives: MARS (Meteorological Archival and Retrieval System) and ECFS (ECMWF File Storage system). Introduction here. Your scratch directory is a good place to put data downloaded from MARS or ECFS, before copying to Oxford. Scratch space is located at `/home/ms/gb/juidj` and is pointed to by the environment variable `$SCRATCH` (e.g. to change to the scratch directory just do `cd $SCRATCH`). The scratch area has a large quota (unlike `$HOME`) but is not backed up (unlike `$HOME`).

2.1 MARS

MARS has been designed to handle large volumes of meteorological data. It contains observations, analyses, forecasts etc. both from the ECMWF operational forecast system and from various research experiments, mainly held in WMO standard formats: GRIB (for fields) and BUFR (for observations).

Data can be retrieved from MARS via requests with a (specially designed language), run in interactive or batch mode. A good starting manual for writing MARS requests can be found on the website.

2.2 ECFS

ECFS is the Centre's archive/retrieval system for user files. Some documentation is available [here](#).

ECFS can be accessed whilst logged into other systems, (e.g. `ecgate`). It is controlled by unix-type commands, but with the prefix `'e'`. For instance:

- `epwd`: print working ECFS directory
- `els`: list contents of current ECFS directory
- `ecd directory`: change to directory on ECFS
- `ecp filesfrom target`: copy `filesfrom` to `target`. This can be used to copy files between systems, e.g. between ECFS and `ecgate`.

...and so on. Note that not all unix commands are replicated for ECFS, and options are sometimes not the same. Check a command with `man command` to find out more. It is often more efficient to create a tar archive and copy this, if you have a large amount of files to copy.

Chapter 3

SMS, CDP & XCDP

SMS is an application that enables users to run a large number of programs which may have dependencies on one another, and in time, in a controlled environment with reasonable tolerance of both hardware and software failures, combined with good restart capabilities. Users talk to SMS using either the command and display program (CDP), or its X-windows equivalent XCDP.

3.1 Getting started with CDP

To start with CDP you first need to do some setup.

1. Log on to ecgate with your User ID.
2. Start a new SMS server. You can do this by running the command:

```
sms_start -r
```

This will start an SMS server with the default SMS_PROG which will be unique to your user. Note down this number! You'll need it in the following steps. The log file will be written to:

```
/home/ms/gb/<uid>/sms_server/ecgate.<prog_num>.sms.log
```

Again, this is the default location. The directory can be changed fairly easily if this is not what you want.

3. Log into the server via cdp.

```
cdp
CDP> setenv SMS\_PROG <prog_num>
CDP> login ecgate <uid> 1
CDP> play suite.def
CDP> begin suite
CDP> exit
```

4. Start XCDP by typing xcdp from the command line
5. In XCDP under the "Edit" menu item, select "Preferences". Then select the "Servers" tab and add the new server:

```
Name: MySuite      (you can use whatever you want here)
Host:  ecgate
Number: <prog_num>
```

Then click on "Add".

6. Still in XCDP, select "Servers" from the menu bar and you should now see "MySuite" as the bottom item. If you select this it will appear in the main XCDP window and you should see the suite.
7. Set an alias to make things easier in CDP in future. You can do this by adding the following line to your \$HOME/.cdprc file:

```
alias ecgate set SMS_PROG >prog_num}902774 \; login ecgb UID 1
```

3.2 Starting a suite

A submission script is available on the gitHub page: subSuite.sh. This will automatically log in to CDP, play and begin a suite and then log out:

```
bash subSuite.sh suiteName.def
```


This is based on using the 'ecgate' alias set above.

Below are listed instructions for manually starting the suite (the subSuite.sh script does all this):

1. cd to the directory where your .def definition file is located
2. Open CDP from the command line by typing cdp.
3. Log on to the ecgate server with your shortcut (instructions for creating this are above)
4. Play the suite by entering play suiteName.def
5. Begin the suite by entering begin suiteName

Note that the suite name suiteName is defined as SUITE_NAME within the .def file, this can be identical to suiteName, however make sure you are using the value of the variable SUITE_NAME with the begin command, not just the name of the .def file.

You can then check the progress by typing status -f from within CDP. Better yet, open XCDP in a new window from the command line (xcdp &) and look from there.

Use the middle click of a mouse to expand the job trees. N.B. If you are using a mac and don't have a mouse you may have to set up a middle click shortcut to do this (xcdp is quite useless without middle click!). There is a useful tool which can help to do this - BetterTouchTool. It used to be free but recently moved to a paid model (though only a few pounds), however it is a useful application.

3.3 Restarting a suite

To restart a suite already in progress (if you have made changes in your .def file for example) you will need to make sure the running tasks in the previous suite have either completed or cancelled. You can do this via XCDP - right click on the task and set complete/aborted. After this check your jobs on the ecgate batch queue: from the command line

```
squeue -u username
```

If necessary kill any running tasks from the old suite

```
skill jobID
```

(you can see the jobID from the squeue command). Once the old tasks are gone from the queue you can restart the suite. If you don't do this, then you will create zombies clogging up the system - these can delay submission of new job. Best practice is to keep the queue as clean as possible!

Then to restart the suite, go into CDP and then enter the command:

```
play -r/suiteName.def suiteName
```

Alternatively a bash script is available resubSuite.sh which will resubmit a suite, from the command line run:

```
bash resubSuite.sh suiteName.def
```

(you can see the jobID from the squeue command). Once the old tasks are gone from the queue you can restart the suite. If you don't do this, then you will create zombies clogging up the system - these can delay submission of new job. Best practice is to keep the queue as clean as possible!