WIM Description.

T.D. Williams
timothy.williams@nersc.no

Nansen Environmental and Remote Sensing Center
Thormøhlensgate 47
5006 Bergen
Norway

March 25, 2015

Contents

Chapter 1

Introduction

Text.

Chapter 2

Mathematical and physical background

Text.

2.1 Scattering model

Text.

2.1.1 Two-dimensional model: a single floe

Text. ?

Chapter 3

Cron and Crontab

In this chapter, after a brief introduction on *Cron*, the user will learn how to check and modify a *Crontab* file.

3.1 Introduction

Cron is a software utility time-based for **Unix-like** operating systems. *Crontab* is a text file that guides *Cron* providing the time and nature of its operations; it is possible for every user to have a personal *Crontab*. Every **minute** *Cron* checks the uploaded *Cronfile* comparing it to the **system** time. These tasks will run regardless of whether the user is actually logged into the system.

3.2 Crontab editing

The layout of the *Crontab* file (fig. 3.1) provides 5 inputs for time and 1 for the command:

```
# * * * * *  command to execute
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └──── day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is Sunday, the same as 0)
# │ │ │ └────── month (1 - 12)
# │ │ └──────── day of month (1 - 31)
# │ └────────── hour (0 - 23)
# └──────────── min (0 - 59)
```

Figure 3.1: *Crontab* file's layout with explanation

```
# This is an example of a Crontab file
* * * * * /home/user/prog1.sh    # Every minute
15 * * * * /home/user/prog2.sh   # Every hour at min 15
# Comments don't interfere with Cron
12 18 * * * /home/user/prog3.sh # Every day at 18:12
*/5 * * 1 * /home/user/prog4.sh # Every 5 minutes of January
```

NOTE - *\ will reiterate that command

To modify the *Crontab* a previously written file can be uploaded (*i.e. mycrontab*) or it can be manually modified (this will change the current user's).

```
$ crontab mycrontab      # Uploading a personal Crontab
$ crontab -e             # Manually modifying the Crontab
```

3.3 Crontab on Hexagon

Cron is working on **Hexagon** as well but since is a very busy and heavily stressed system there are a few precautions that should be remembered before setting up a *Cron* job.

First of all you should choose a *node* in which set up your *Crontab*. A *node* is a connection, redistribution or communication point for a network. For more information please visit (`http://docs.hpc.uib.no/wiki/Main_Page`).
Another precaution is to use unique *Crontab* file's names so that different set-ups will not be uploaded by mistake.

Here is shown a complete procedure for a *Crontab* setup in **Hexagon**:

```
$ ssh hexagon            # Personal login on Hexagon

# Now a Cron-job is appended to a unique tab
$ echo "* * * * * /home/user/prog.sh" >> prog_cron

$ ssh login1             # Login node1 (change number to change node)
$ crontab prog_cron      # Uploading unique Crontab
```

Now every minute *Cron* will execute *prog.sh* from *node1*.

Remember that the outputs of *Cron* are shown in a virtual display hence are lost if not saved. It is possible to set up an automatic e-mail service that will save and send to a specified address the outputs by adding the following line in the *Crontab* file:

```
$ MAILTO=user@domain.org
```

Chapter 4

Model's Inputs Updating Procedure

In this chapter the user will be instructed on how to control the scripts used to download and storage the input files that will be used as initial conditions for the WIM model.
There are four products that are updated and/or stored daily:

- TOPAZ - Weekly Restart Files
- WAMNSEA - Daily Waves
- ECMWFR - Daily Weather

4.1 TOPAZ

4.1.1 Introduction

Towards an **O**perational **P**rediction system for the North **A**tlantic European coastal **Z**ones, simply known as TOPAZ is a coupled ocean-sea ice data assimilation system for the North Atlantic Ocean and Arctic. It is the only operational, large-scale ocean data assimilation system that uses the ensemble Kalman filter. This means that TOPAZ features a time-evolving, state-dependent estimate of the state error covariance.

In the regional Barents and Kara Sea forecast system (**BS1**), the TOPAZ (**TP4**) is used as an outer model in the nested system. Locally the TP4 model runs once a week for an 11 days period, with 9 days forecast, and produce initial and boundary conditions for the regional model BS1.

4.1.2 Data Gathering - last modification March 25, 2015

The initial product consist of 3 different files that will provide initial conditions:

- **TP4restart *YYYY_ddd_hh*_mem001.a** - [**binary data**] initial conditions of the system.
- **TP4restart *YYYY_ddd_hh*_mem001.b** - [**text**] information and instructions for the first file.

- **TP4restart *YYYY_ddd_hh*ICE.uf** - [**data**] informations about sea ice (localization, height, etc..).

NOTE - file names could change depending on the service

These files are weekly uploaded to the main server of the forecast system, HEXAGON, the supercomputer service at the University of Bergen. Are then temporary stored in a working directory that will be cleared every two weeks hence the need of a script that will daily check and if needed archive and transfer the new data.
The logic behind the script is to find all the files with **TP4restart** in their file-name and check their presence in a list of the stored files. If the file is on the list it has already been archived and the script will move on otherwise will automatically add the new files to the list and to the archive (using a *tar* compression protocol). Every operation will write a report on a daily log that will be sent to a selected user (see chapter [3]).