

WIM Description.

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

Nansen Environmental and Remote Sensing Center
Thormøhlensgate 47
5006 Bergen
Norway

March 25, 2015

Contents

1	Introduction	2
2	Mathematical and physical background	3
2.1	Scattering model	3
2.1.1	Two-dimensional model: a single floe	3
3	Cron and Crontab	4
3.1	Introduction	4
3.2	Crontab editing	4
3.3	5
4	Model's Inputs Updating Procedure	6
4.1	TOPAZ	6
4.1.1	Introduction	6
4.1.2	Data Gathering - last modification March 25, 2015	6

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*. 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:

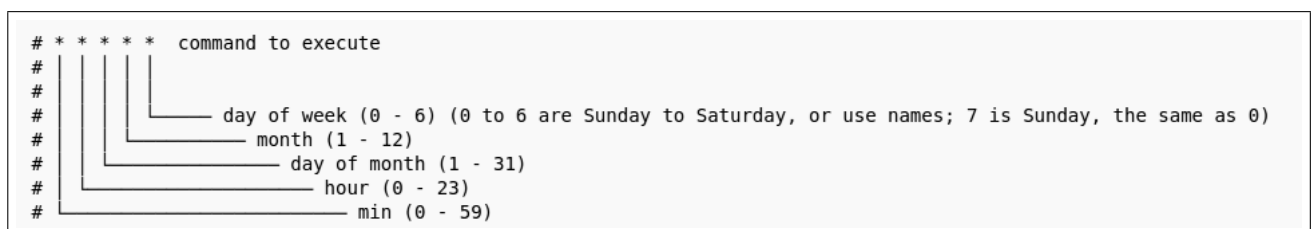


Figure 3.1: *Crontab* file's layout with explanation

```
# This is an example of a Crontab file
* * * * * /home/user/prog.sh      # Every minute
15 * * * * /home/user/prog.sh    # Every hour at min 15
12 18 * * * /home/user/prog.sh   # Every day at 18:12
*/5 * * 1 * /home/user/prog.sh   # Every 5 minutes of January
```

NOTE - `*\` will reiterate that command

To modify the *Crontab* it's possible to upload a file previously made (*i.e.* *mycrontab*) or

to manually modify it (this will change the current user's *Crontab*).

<pre>\$ crontab mycrontab <i>#loading a personal crontab</i> \$ crontab -e <i>#manually modifying the crontab</i></pre>
--

3.3

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_hhICE.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]).