NILU: OR 56/2002

NILU: OR 56/2002
REFERENCE: O-100064
DATE: NOVEMBER 2002
ISBN: 82-425-1404-6

Implementation of extraction routine for ECMWF data through the CalVal web-pages

Final development report

Aasmund Fahre Vik, Terje Krognes, Sam Erik Walker, Trygve Bårde, Roland Paltiel, Kjersti Karlsen Tørnkvist, Bjørn Gloslie, Rita Larsen and The Nguyen Thanh

Preface

This report describes a new tool that has been implemented by NILU on the ESA CalVal web-pages. The new technology is based on an old data-extraction routine called met-mars, and the new development makes it possible to perform similar extractions through a web-interface. The data output is furthermore changed from ASCII to the binary HDF format.

Contents

Summary

The extraction tool consists of mainly two parts. The first is a UNIX shell-script, hdf-mars, that extracts ECMWF analysed data, performs calculations and prints out an hdf-file according to the ESA CalVal metadata guidelines. The second part is the web interface that lets the user specify parameters, starts the shell script and displays its output.

Implementation of extraction routine for ECMWF data through the CalVal web-pages
Final development report

1 Description of the extraction tool

The extraction tool consists of mainly two parts. The first is a UNIX shell-script, hdf-mars, that extracts ECMWF analysed data, performs calculations and prints out an hdf-file according to the ESA CalVal metadata guidelines. The second part is the web interface that lets the user specify parameters, starts the shell script and displays its output.

Source code of the program hdf-mars is displayed in appendix A, and the script can be found under /nadir/bin/ at the server zardoz.nilu.no. It is based on a shell-script, met-mars_v2, that is used to extract and calculate PV- (Potential Vorticity), temperature- humidity- and wind-fields from ECMWF analysed data. The script accepts a number of input parameters to specify date and time, geolocation of extracted fields, output parameter etc. The new version of the program, hdf-mars, takes more or less the same arguments, but the output is an hdf-file instead of NASA-AMES (strict ASCII format). Another difference is that a filename should not be specified for hdf-mars since the script creates a filename that is according to the ESA CalVal metadata guidelines. The main task of the shell-script is to start up subroutines that perform the actual extractions and calculations of data. The current development includes a total rewrite of 5 such subroutines or subprograms. Each of these exists in 4 different versions (adapted to various versions of ECMWF input data), giving a total of 20 programs. These are named hy2th_hdf, hy2p_hdf or generally hy2xx_hdf and are located und /nadir/bin/hdfmarsbin/.

Hdf-mars works as follows: All input parameters are validated and checked for consistency with existing data. Only a specific set of parameters are allowed. A program, sp2ll_v2, is called to read ECMWF data and interpolate these from a spherical harmonic form to a regular latitude longitude one. Depending on the requested output parameter, one or several hy2xx_hdf programs are called to calculate data and interpolate these from hybrid levels to either pressure or theta surfaces. The output from these programs is two ASCII-files, one data- and one meta-data file. These are used as input for the Asc2hdf program previously developed at NILU under contract by ESA, and the final hdf-file is created and written to the user home-directory together with a log-file. The logfile is printed to the screen at the end of the script. The output may also be written to a user-specified directory (the last and only optional argument), and this feature is used when the script is accessed through a web-interface.

2 Description of the web-interface

This part consist of two files, t106_in.php and t106_ex.php, and can be found at http://nadir.nilu.no/ecmwf/. The first one has boxes and drop-down menus that help the users to specify correct input parameters for the hdf-mars script. A submit-button starts the program through a remote-shell (rsh) since the script and the web-server is running on different servers. All output from the remote shell is displayed in the second web-page (happens automatically), and a link is provided to the extracted hdf-file. Any error-messages or incorrect input will therefore be available on the t106_ex.php-page.

Appendix A

Source code for the extraction script hdf-mars

This file is a recoded version of the shell-script met-mars_v2. Both versions can be found at zardoz.nilu.no under /nadir/bin/. Both scripts are open for all users at the server, but the properties of ECMWF are protected by a strict control of access to the analysed data both programs need for extractions. The source code of the programs sp2ll_v2 is owned by ECMWF, as is also the old versions of hy2xx. According to contract agreements, the source of the new hy2xx_hdf Fortran programs is not shown here.

```
#!/bin/sh
#
#
# This script extracts and interpolates Mars t106 data from Mars t106 binary
# data files and interpolate onto pressure or theta surfaces.
# The script can also produce PV data from mars t106 data availible.
#
# This script make calls to several programs, includning sp2ll_v2, hy2p,
# hy2th etc.
#
#
# Name: hdf-mars
#
# Written in Bourne Shell by:
#
# Original by Ole W Saastad, NILU.
#
# Version 1.2  05/09-1995
# Version 1.21 06/09/1995 Tackles both 19 level and 31 level data.
# Versjon 1.22 06/02/1996 New datastructure for input data.
# Version 1.23 27/02/1996 Problems with calculation of dir. 2 and 02.
# Version 1.3  19/06/1996 All t106 data in the same data structure ! CD jukebox.
# Version 1.31 24/02/1997 Changed upper year limit to 1997.
# Version 1.32 07/01/1998 Changed upper year limit to 1998.
#
#
# Version 2.00 02/04/1999 BRB - Beta: use Version 2 programs.
#              06/11/1999 BRB - Beta: Y2K & include 60-level model also.
#              08/12/1999 BRB - Final version.
# Version 2.01 03/01/2002 TRB - Implemented the variabel MAXYEAR, so
#                               the script have not to be updated every
#                               new year.
#
# Current version is called hdf-mars; a more or less clone of met-mars_v2
# It should not have an argument for output file, since this is generated
# automagically by the asc2hdf program. Final output of this script is a
# hdf file and it is put in the users homedirectory.
#
# hdf-mars was coded by Aasmund Fahre Vik, July 2002
#
# ***************************************************************************
# *                                                                         *
# *         Declaration of at trap to remove temporary files.               *
# *                                                                         *
# *                                                                         *
# ***************************************************************************


trap "/bin/rm -rf $TMP/*.$$; exit 1" 1 2 3 9 15


# ***************************************************************************
# *                                                                         *
# *         The usage function                                              *
# *                                                                         *
# *                                                                         *
# ***************************************************************************

usage()
echo '
usage :

hdf-mars yy mm dd hh west east north south resol surface level variable
```

```
or

hdf-mars help

Note that "outfile" is not an argument - The output is written to your home dir.
It you use a 13th argument (e.g. extra) your output will be written to
    /nadir/tmp/hdfmars/extra/
'


# ************************************************************************
# *                                                                      *
# *         The help function                                            *
# *                                                                      *
# *                                                                      *
# ************************************************************************

help()
  echo '
Welcome to Version 1.00 6/8 2002 of the hdf-mars program

This script can extract meteorological data from Mars T106 data origin from
ECMWF, and interpolate these model level data onto pressure and theta surfaces.
The script uses a program called sp2ll_v2 to extract met. data like ground
pressure, temperature, zonal and meridional wind. These meteorological data are
on 31 (19 Prior to 1991) model levels, hybrid levels.
Some programs hy2* are called to interpolate from hybrid levels to
either pressure or theta surfaces.

Not all combinations of levels and variables are allowed.
For some sophisticated uses the manual run of sp2ll_v2 and subsequent hy2*
programs are suggested.

Output is a hdf-file, and it is written to your homedirectory. Filename is
generated automatically - all in accordance with the esa calval metadata
definitions

Possible resolutions are in the range from high resolution 1.125 to low
resolution at 5.0 degrees. (At present only 1.125 can be used)

An example of extraction of PV onto a 475 K theta surface :

hdf-mars 00 07 22 6 -180 180 90 30 1.125 th 475 PV



 '


# ************************************************************************
# *                                                                      *
# *         Find out if we are on zardoz or not ?                        *
# *                                                                      *
# *                                                                      *
# ************************************************************************

setpath()
{

        yy=$1
        mm=$2

#
# Make the month with two digits. Only the month is used to calc. the dir.
#
          m=`echo $mm | awk '{printf("%.2d\n",$1)}'`

          export MARSPATH SCRATCH TMP BIN

# Find the correct path for data. Is it daily or archieve data ?


# There are two possible storage paths:
# Depending on if the script is started from internal network or outside.
```

```
          if [ $yy -lt 88 ]; then
           ute=/nadir/t106/20$yy\/$m
           inne=/extern/nadir/t106/20$yy\/$m
#
           for DIR in $ute $inne
           do
             [ -d $DIR ] && MARSPATH=${MARSPATH:-$DIR}
           done
#
        elif [ $yy -ge 88 -a $yy -lt 100 ]; then
           ute=/nadir/t106/19$yy\/$m
           inne=/extern/nadir/t106/19$yy\/$m
#
           for DIR in $ute $inne
           do
             [ -d $DIR ] && MARSPATH=${MARSPATH:-$DIR}
           done
#
        fi

# Set the other directories.

          for DIR in /nadir/tmp/scratch /extern/nadir/tmp/scratch
              do
             [ -d $DIR ] && SCRATCH=${SCRATCH:-$DIR}
              done

          for DIR in /nadir/tmp /extern/nadir/tmp
              do
             [ -d $DIR ] && TMP=${TMP:-$DIR}
              done
          for DIR in /nadir/bin/hdfmarsbin /extern/nadir/bin/hdfmarsbin
              do
             [ -d $DIR ] && BIN=${BIN:-$DIR}
              done
          for DIR in /nadir/bin /extern/nadir/bin
              do
             [ -d $DIR ] && S_BIN=${S_BIN:-$DIR}
              done


}


# ***************************************************************************
# *                                                                         *
# *         Determine the correct date and time  (dato tids gruppe)         *
# *         and check if input data exists.                                 *
# *                                                                         *
# *                                                                         *
# ***************************************************************************

dtg()
{
  MAXYEAR=`date +%y`
  yy=$1
  mm=$2
  dd=$3
  hh=$4

# data from 1988 on!
  if [ $yy -gt $MAXYEAR -a $yy -lt 88 -o $yy -gt 99 ]; then
     if [ $yy -gt 56 -a $yy -lt 88 ]; then
        echo 'Year is incorrect: ' 19$yy
          exit 1
     elif [ $yy -gt 00 -a $yy -lt 57 ]; then
        echo 'Year is incorrect: ' 20$yy
        exit 1
     fi
  fi
  if [ $mm -lt 1 -o $mm -gt 12 ]; then
          echo 'Month is incorrect ' $mm
          exit 1
  fi
  if [ $dd -lt 1 -o $dd -gt 31 ]; then
          echo 'Date is incorrect' $dd
          exit 1
```

```
  fi
  if [ $hh -ne 0 -a $hh -ne 6 -a $hh -ne 12 -a $hh -ne 18 ]; then
          echo 'Hour must be either 0, 6, 12 or 18 !  :' $hh
          exit 1
  fi

  y=$yy
  m=`echo $mm | awk '{printf("%.2d\n",$1)}'`
  d=`echo $dd | awk '{printf("%.2d\n",$1)}'`
  h=`echo $hh | awk '{printf("%.2d\n",$1)}'`

  METFILE=$MARSPATH\/nilut106.$y$m$d$h


  if [ -f $METFILE ]; then
          export METFILE
#
# Allocate programs
# For the 19 level model data the LNSP is at level 0, but for newer 31/50/60
level.
# LNSP is located at level 1.
          if [ $yy$m -gt 8810 -a $yy$m -lt 9110 ]; then
            HY2TH=hy2th_hdf_19
            HY2PV=hy2pv_hdf_19
            HY2P=hy2p_hdf_19
            HY2Z_TH=hy2z_th_hdf_19
            HY2Z_P=hy2z_p_hdf_19
            LV=0
          elif [ $yy$m$d -gt 990308 -a $yy$m$d -lt 991012 ]; then
          HY2TH=hy2th_hdf_50
          HY2PV=hy2pv_hdf_50
          HY2P=hy2p_hdf_50
          HY2Z_TH=hy2z_th_hdf_50
          HY2Z_P=hy2z_p_hdf_50
          LV=1
          elif [ $yy$m -lt 5701 -o $yy$m$d -gt 991011 ]; then
          HY2TH=hy2th_hdf_60
          HY2PV=hy2pv_hdf_60
          HY2P=hy2p_hdf_60
          HY2Z_TH=hy2z_th_hdf_60
          HY2Z_P=hy2z_p_hdf_60
          LV=1
          else
            HY2TH=hy2th_hdf_31
            HY2PV=hy2pv_hdf_31
            HY2P=hy2p_hdf_31
            HY2Z_TH=hy2z_th_hdf_31
            HY2Z_P=hy2z_p_hdf_31
            LV=1
          fi
#
  else
          echo ' '
          echo 'Input file does not exist for the date : '$yy $mm $dd $hh
        echo ' '
          echo 'Check your date and time input. Daily data are availible from '
        echo 'January 8th 1995 00 UTC and onwards '
          echo ' '
          echo 'Archived data exist from November 11 th 1988,'
        echo 'and subsequent winters.'
          echo ' '
          exit 1
   fi
}


# ************************************************************************
# *                                                                      *
# *         Read and check the area supplied by the user.                *
# *                                                                      *
# *                                                                      *
# ************************************************************************

area()
{
  west=$1
  east=$2
```

```
  north=$3
  south=$4
  resol=$5

  if [ $west -lt -180 -o $west -gt 180 ]; then
          echo ' West longitude must be in the interval from -180 to 180 '
          exit 1
  fi
  if [ $east -lt -180 -o $east -gt 180 ]; then
          echo ' East longitude must be in the interval from -180 to 180 '
          exit 1
  fi
  if [ $west -gt $east ]; then
          echo ' East must be greather then west. East : '$east 'west : '$west
          exit 1
  fi
  if [ $north -lt -90 -o $north -gt 90 ]; then
          echo ' North lattitude must be in the interval from -90 to 90 '
          exit 1
  fi
  if [ $south -lt -90 -o $south -gt 90 ]; then
          echo ' South lattitude must be in the interval from -90 to 90 '
          exit 1
  fi
  if [ $north -lt $south ]; then
          echo ' North must be greater than south. North : ' $north ' south : '
$south
          exit 1
  fi
  if [ $resol -lt 1.125 -o $resol -gt 1.125 ]; then
          echo 'Resolution must be in the interval 1.125 to 1.125 '
          echo 'At present only 1.125 degrees work.'
          exit 1
  fi
}


# ************************************************************************
# *                                                                      *
# *         Determine whether pressure surface of theta surface.         *
# *                                                                      *
# *                                                                      *
# ************************************************************************

P_or_TH()
{
  SURFACE=$1
  LEVEL=$2

  if [ "$SURFACE" != "P" -a "$SURFACE" != "TH" -a "$SURFACE" != "p" -a "$SURFACE"
!= "th" ]; then
          echo ' Level must be either P for pressure surface or TH for theta
surface '
          exit 1
  fi

  if [ "$SURFACE" = "P" -o "$SURFACE" = "p" ]; then
          if `echo '.1 0.1 .14 0.14 .2 0.2 .3 0.3 .5 0.5 .7 0.7 1. 1.4 2. 3. 5. 7.
10. 14. 20. 30. 50. 70. 90. 100. 140. 200. 300. 500. 700. 850. 1000. 1013.25 -1' |
fgrep -s -v -e $LEVEL` ; then
            echo 'Incorrect pressure surface ' $LEVEL
            echo ' '
            echo 'Valid levels are : '
            echo '   0.1    0.14   0.2   0.3    0.5   0.7'
          echo '   1.     1.4    2.    3.     5.    7. '
          echo '  10.    14.    20.   30.    50.   70.   90.'
          echo ' 100.   140.   200.  300.   500.  700.  850.'
          echo '1000.  1013.25'
          echo 'Levels less than 10. are for files after March 8, 1999'
          echo '-1 is for all levels'
            exit 1
          fi
  fi

  if [ "$SURFACE" = "TH" -o "$SURFACE" = "th" ]; then
```

```
    if `echo '3000, 2750, 2500, 2250, 2000, 1750, 1500, 1250, 1000, 975, 950, 925,
900, 875, 850, 825, 800, 775, 750, 725, 700, 695, 690, 685, 680, 675, 670, 665,
660, 655, 650, 645, 640, 635, 630, 625, 620, 615, 610, 605, 600, 595, 590, 585,
580, 575, 570, 565, 560, 555, 550, 545, 540, 535, 530, 525, 520, 515, 510, 505,
500, 495, 490, 485, 480, 475, 470, 465, 460, 455, 450, 445, 440, 435, 430, 425,
420, 415, 410, 405, 400, 395, 390, 385, 380, 375, 370, 365, 360, 355, 350, 345,
340, 335, 330, 325, 320, 315, 310, 305, 300, 295, 290, 285, 280, 275, 270, 265,
260, 255, 250, 245, 240, 235, 230, 225, 220, 215, 210, 205, 200, -1' | fgrep -s -v
-e  $LEVEL` ; then
          echo 'Incorrect Theta surface ' $LEVEL
          echo ' '
        echo 'Valid levels are : '
          echo '3000, 2750, 2500, 2250, 2000, 1750, 1500, 1250, 1000, 975, 950,
925, 900, 875, 850, 825, 800, 775, 750, 725, 700, 675, 650, 625, 600, 575, 550,
525, 500, 475, 450, 425, 400, 375, 350, 325, 300, 275, 250, 225, 200'
        echo ' '
        echo '3000[k] to 700[k] is for files after March 8, 1999'
        echo '-1 is for all levels'
          exit 1
    fi
  fi
}


# ***************************************************************************
# *                                                                         *
# *         Read and check that the variable is a valid one.                *
# *                                                                         *
# *                                                                         *
# ***************************************************************************

variable_daily()
{
  par=$1

  par=`echo $par | tr 'tuvwpvz' 'TUVWPVZ' `

  if [ $par != T -a $par != U -a $par != V -a $par != W -a $par != Z -a $par != PV
]; then
          echo 'Variablbe must be T,U,V,W,Z or PV  :' $par
          exit 1
  fi
}


variable_arch()
{
  par=$1

  par=`echo $par | tr 'qtuvwpvz' 'QTUVWPVZ' `

  if [ $par != Q -a  $par != T -a $par != U -a $par != V -a $par != W -a $par != Z
-a $par != PV ]; then
          echo 'Variablbe must be Q,T,U,V,W,Z or PV  :' $par
          exit 1
  fi
}

# ***************************************************************************
# *                                                                         *
# *     Execution  of the script starts here.                               *
# *                                                                         *
# *                                                                         *
# ***************************************************************************

  if [ $# -eq 1 ]; then
          if [ $1 = "help" -o $1 = "HELP" ]; then
            help
            exit 1
          else
            usage
            exit 1
          fi
  else
          if [ $# -eq 13 ]; then
            movefile=1
          else
```

```
            if [ $# -eq 12 ]; then
              movefile=0
            else
              usage
              exit 1
            fi
          fi
  fi

  setpath $1 $2
  dtg $1 $2 $3 $4
  area $5 $6 $7 $8 $9

  shift 9
# Shift to get more aguments, $10 will now be $1.

  if [ $movefile -eq 1 ]; then
    outdir=$4
  fi

  if [ $yy -ge 95 ]; then
          variable_daily $3
  else
          variable_arch $3
  fi
  P_or_TH $1 $2

#  filename $4
#  the user may not choose the output filename. It is rather created
#  by asc2hdf according to the esa calval metadata guidelines

# PRESSURE SURFACE :

# Extraction and interpolation onto pressure surfaces are requested.

  if [ $SURFACE = P -o $SURFACE = p ]; then
          $S_BIN/sp2ll_v2 $yy $mm $dd $hh LNSP $LV $resol $west $east $north
$south $TMP/LNSP.$$
          case $par
          in
          Q | T | U | V | W)
          $S_BIN/sp2ll_v2 $yy $mm $dd $hh $par -1 $resol $west $east $north $south
$TMP/$par.$$
            $BIN/$HY2P $LEVEL $TMP/LNSP.$$ $TMP/$par.$$ $TMP/RESULT.$$
          ;;
          Z)
          $S_BIN/sp2ll_v2 $yy $mm $dd $hh T -1 $resol $west $east $north $south
$TMP/T.$$
            $BIN/$HY2Z_P $LEVEL $TMP/LNSP.$$ $TMP/T.$$ $TMP/RESULT.$$
          ;;
          *)
            echo ' met-mars_v2: '
            echo ' Unable to process the requested data. '
          echo ' Potential Vorticity can only be requested on theta surfaces. '
            exit 1
          ;;
          esac
  fi


# THETA SURFACE :

# Extraction and interpolation onto theta surfaces are requested.

  if [ $SURFACE = TH -o $SURFACE = th ]; then
          $S_BIN/sp2ll_v2 $yy $mm $dd $hh LNSP $LV $resol $west $east $north
$south $TMP/LNSP.$$
          $S_BIN/sp2ll_v2 $yy $mm $dd $hh T   -1 $resol $west $east $north $south
$TMP/T.$$
          case $par
          in
          Q | U | V | W)
            $S_BIN/sp2ll_v2 $yy $mm $dd $hh $par -1 $resol $west $east $north
$south $TMP/$par.$$
            $BIN/$HY2TH $LEVEL $TMP/LNSP.$$ $TMP/T.$$ $TMP/$par.$$ $TMP/RESULT.$$
          ;;
          T)
```

```
            $BIN/$HY2TH $LEVEL $TMP/LNSP.$$ $TMP/T.$$ $TMP/RESULT.$$
          ;;
          PV)
            $S_BIN/sp2ll_v2 $yy $mm $dd $hh U -1 $resol $west $east $north $south
$TMP/U.$$
            $S_BIN/sp2ll_v2 $yy $mm $dd $hh V -1 $resol $west $east $north $south
$TMP/V.$$
            $BIN/$HY2PV $LEVEL $TMP/LNSP.$$ $TMP/T.$$ $TMP/U.$$ $TMP/V.$$
$TMP/RESULT.$$
          ;;
          Z)
            $S_BIN/sp2ll_v2 $yy $mm $dd $hh T -1 $resol $west $east $north $south
$TMP/T.$$
            $BIN/$HY2Z_TH $LEVEL $TMP/LNSP.$$ $TMP/T.$$ $TMP/RESULT.$$
          ;;
          *)
             echo ' met-mars_v2: '
             echo ' Unable to process the requested data. '
             echo ' Refer to help for additional information '
             exit 1
          ;;
          esac
  fi


# The extracted and interpolated data are placed in the users home-dir

    asc2hdfpath=/viper2/nadir/bin
    datafile=$TMP/RESULT.$$
    metadatafile=$TMP/RESULT.$$.meta
    tablefile=/nadir/esa/.scriptdata/table.dat
    logfile=$TMP/logfile.$$

#   When the script is run via a webpage, 13th argument should be given. Then
#   a directory will be created under /nadir/tmp and the output will be
#   written here.

    if [ $movefile -eq 1 ]; then
      mkdir /viper2/nadir/tmp/hdfmars/$outdir
      chmod 755 /viper2/nadir/tmp/hdfmars/$outdir
      cd /viper2/nadir/tmp/hdfmars/$outdir
    else
      cd
    fi

    $asc2hdfpath/asc2hdf $metadatafile $datafile $tablefile $logfile
    cat $logfile
    /bin/rm -rf $logfile

    if [ $movefile -eq 1 ]; then
      chgrp ecmwf4 *
      chmod 750 *
    fi
#  fi

# Remove temporary files:

  /bin/rm -rf $TMP/*.$$
  /bin/rm -rf $TMP/*.$$.meta
  exit 0
```

Appendix B

Abbreviations

NILU	Norsk institutt for luftforskning, Norwegian Institute for Air Research
ECMWF	European Centre for Medium-Ranged Weather Forecasts
CalVal	Calibration and Validation of ENVISAT
ENVISAT	ENVIronmental SATellite
Hdf	Hierarchical data format

Norwegian Institute for Air Research (NILU)
P.O. Box 100, N-2027 Kjeller, Norway

REPORT SERIES	REPORT NO. OR 56/2002	ISBN 82-425-1404-6	
SCIENTIFIC REPORT		ISSN 0807-7207	
DATE:	SIGN.	NO. OF PAGES	PRICE
		16	NOK 150.-

TITLE

Implementation of extraction routine for ECMWF data through the CalVal web-pages

Final development report

PROJECT LEADER

Aasmund Fahre Vik

NILU PROJECT NO.

O-100064

AUTHOR(S)

Aasmund Fahre Vik, Terje Krognes, Sam Erik Walker, Trygve Bårde, Roland Paltiel, Kjersti Karlsen Tørnkvist, Bjørn Gloslie, Rita Larsen and The Nguyen Thanh

CLASSIFICATION *

A

CONTRACT REF.

REPORT PREPARED FOR
European Space Agency / ESRIN
Contracts service
Via Galileo Galilei
P.O. Box 64
IT-00044 FRASCATI (ROME), ITALY

ABSTRACT

The extraction tool consists of mainly two parts. The first is a UNIX shell-script, hdf-mars, that extracts ECMWF analysed data, performs calculations and prints out an hdf-file according to the ESA CalVal metadata guidelines. The second part is the web interface that lets the user specify parameters, starts the shell script and displays its output.

NORWEGIAN TITLE

Implementering av en ekstraksjonsrutine av ECMWF data via CalVal webportalen.

KEYWORDS		
Calval	ECMWF	hdf

ABSTRACT (in Norwegian)

Verktøyet for ekstrahering består i hovedsak av to deler. Den første er et UNIX program, hdf-mars, som leser ECMWF analsyerte data, utfører beregninger på disse og skriver ut en hdf-fil som er i henhold til ESA CalVal metadata standarden. Den andre delen er et web-grensesnitt som lar brukeren spesifisere ulike parametre, starter hdf-mars og viser resultatene fra denne.

** Classification*
A Unclassified (can be ordered from NILU)
B Restricted distribution
C Classified (not to be distributed)