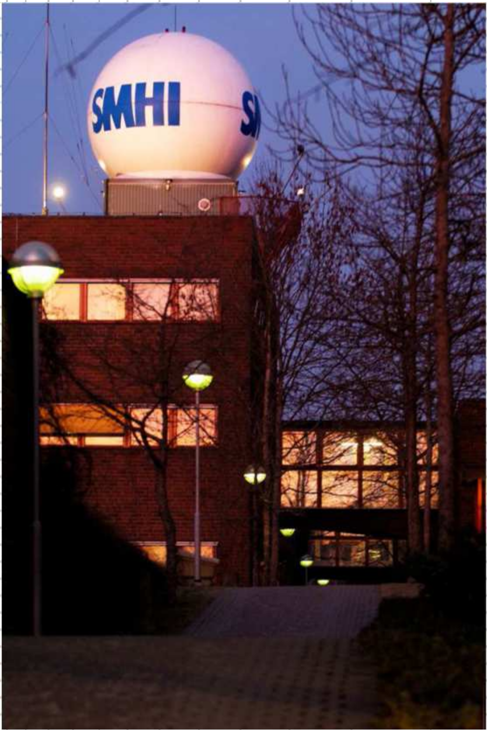
Radar observation treatment

Hirlam-B Training Week on HARMONIE system
Norrköping, Wednesday, 17 September, 2014

Martin Ridal

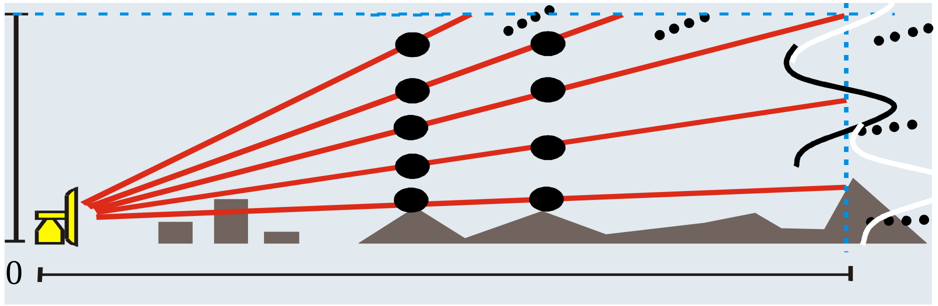


Radar assimilation

- Radar data
  - Volume scans from each radar
  - Very large data amounts

- Reflectivities
  - Difficult to do direct assimilation (complicated relation between control variables and reflectivity, including microphysics)
  - 1D + 3DVar
  - Assimilation of a humidity pseudo observation
  - Assimilation of “no humidity” to dry the model

- Radial velocities
  - Easier and more straight forward
  - Dealiasing is needed



HARMONIE radar assimilation: Reflectivity

- Combination of 1D Bayesian and 3DVAR schemes
  - Developed by MeteoFrance (Caumont et. al, 2007)

- 1D humidity profiles are retrieved utilizing reflectivity measurements
  - Comparing radar reflectivity with simulated reflectivity to obtain weights for interpolation
  - Pseudo-observations of relative humidity using the weights model humidity to observation point
  - If there is no precipitation in the model, but in observations, humidity is set to 100%
  - If there is precipitation in the model but not in observations, surrounding dry pixels are used

- Pseudo-observations assimilated using 3DVAR

- Areas of no precipitation
  - It is also important to assimilate pixels with no reflectivity to dry the model
  - “nodata” or “undetected” ?!

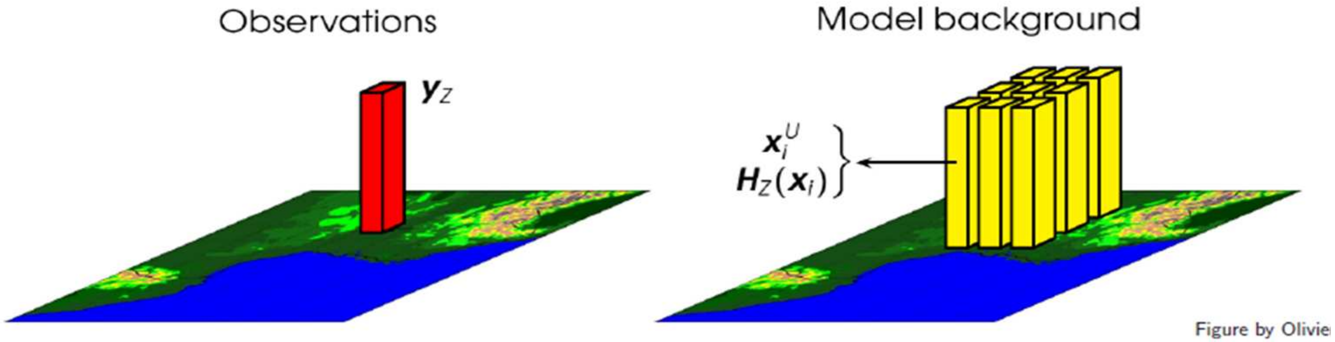
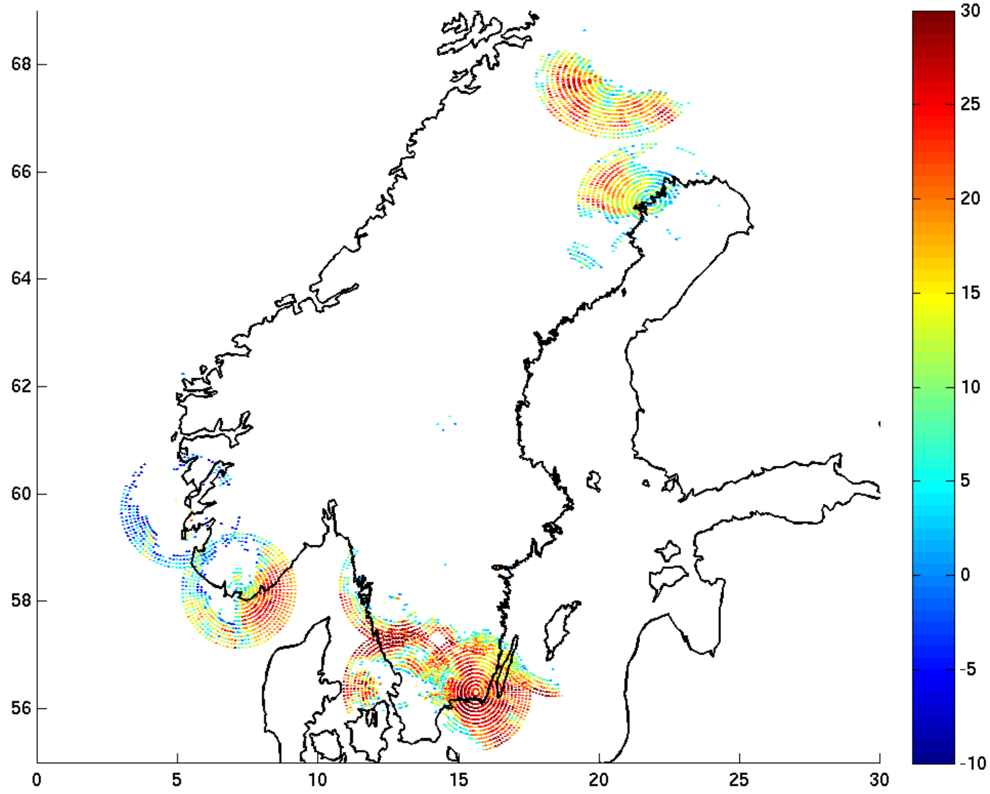
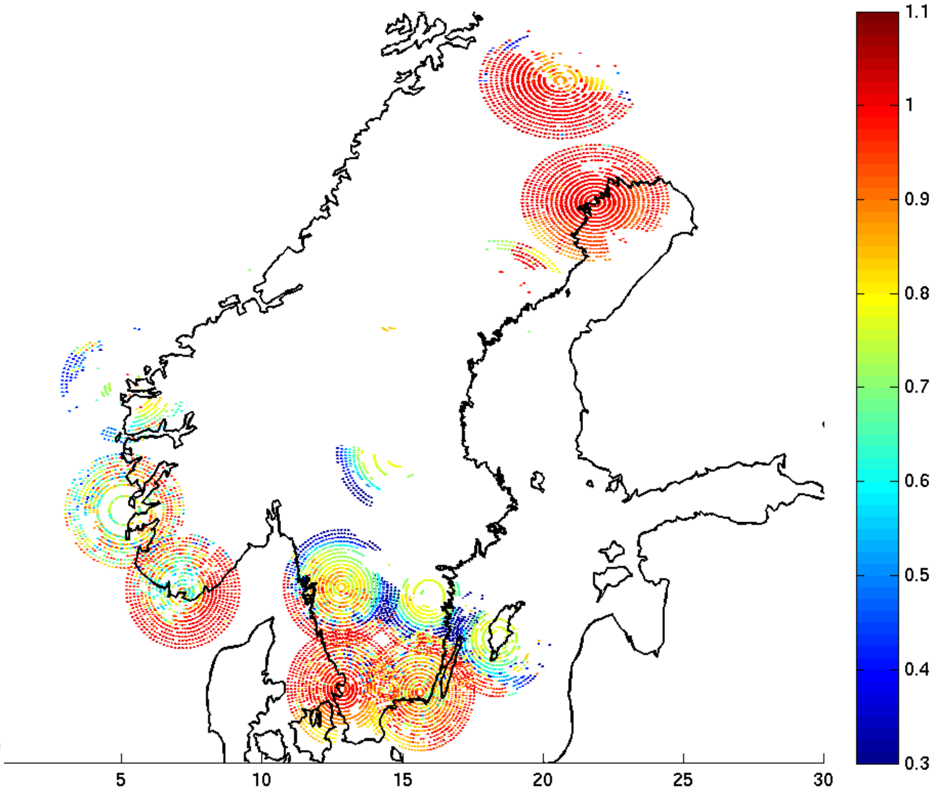


Figure by Olivie

Observed reflectivity



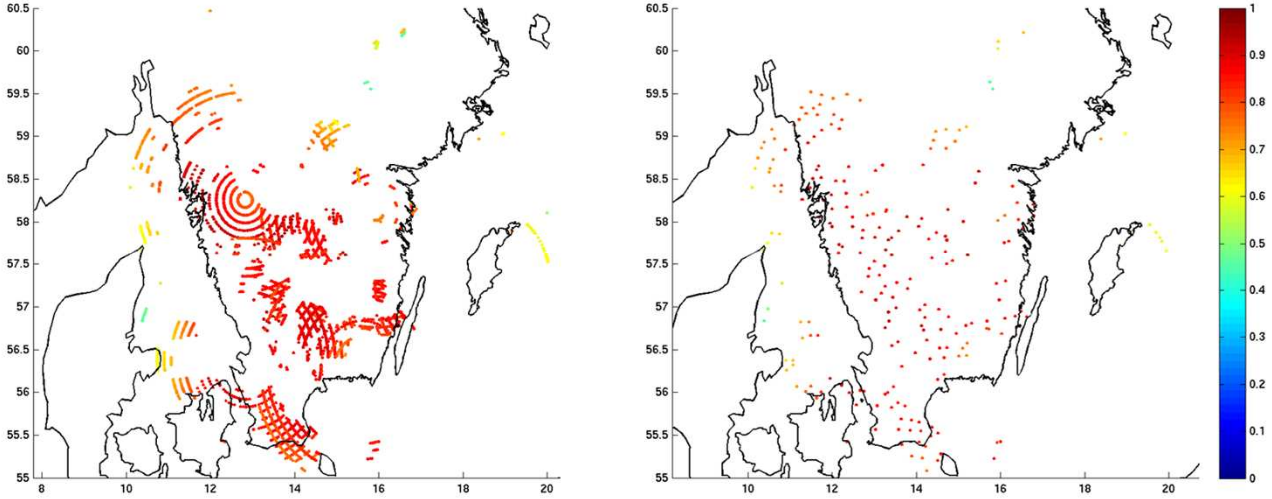
Pseudo observation of humidity



Very large data amount

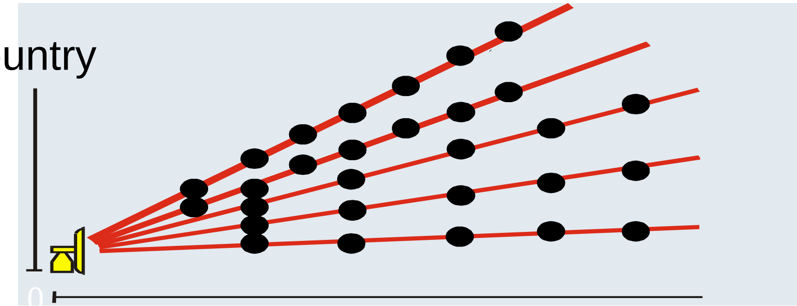
- Thinning
  - – Pre-thinning to be able to read the data
  - – Thinning in BATOR to be able to handle it in the screening
  - – Final thinning in the screening with mindist = 15 km

- Super observations
  - – Calculate spatial averages of the raw data
  - – Needs to be made carefully, e.g. use precipitating or non-precipitating pixels

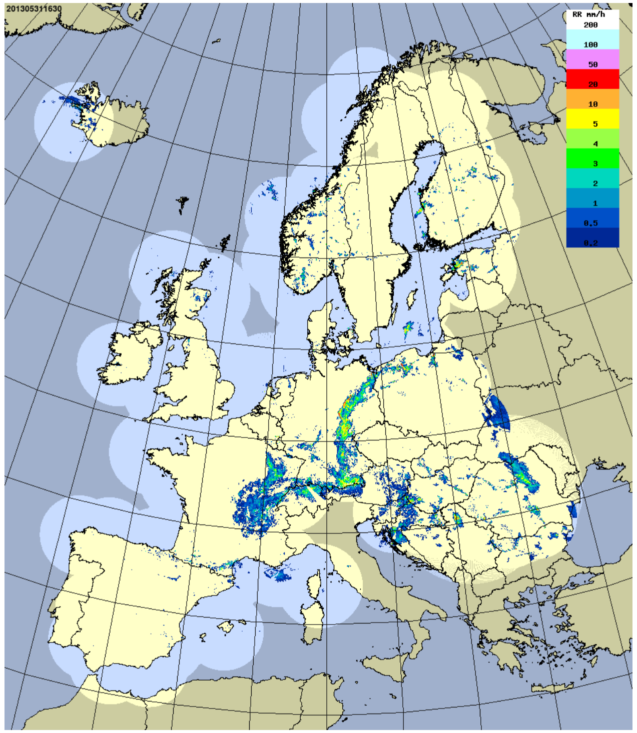


HARMONIE assimilation: Challenges

- Different data formats from different data suppliers
  - HDF5, BUFR, internal formats…
  - Many countries are aiming for the OPERA Information Data Model (ODIM) in HDF5 or BUFR file format
- Different grid types
  - Most countries use polar coordinates (azimuth angle and range)
  - Different volume sizes
- Different scan strategies
  - Different for different elevations
  - Different for reflectivity and radial velocity
- Different quality of the data
  - Different levels of quality control in each country



ODIM HDF5



- OPERA
  - Eumetnet programme to collect radar data
  - Produces 2D radar composites
  - Little distribution of volume data and low demand for winds
  - NWP usage put new, high demands on the data

- Advantages of ODIM_H5
  - Well defined
  - Well documented
  - Most countries deliver data in ODIM_H5 to OPERA
  - The output format from the BALTRAD toolbox (QC)

- Local differences (errors)
  - Wrong naming, unit errors, missing parameters…
  - Not detected since there has been no such advanced usage earlier
  - The use of “undetect” and “nodata”

- Delivery of volumes or scans
  - The model expects volumes
  - Can cause differences when the scans are put together into a volume

HARMONIE assimilation: Where are we?

- Input data format
  - A conversion tool is available (local format → MF-BUFR)
  - A reader for ODIM_H5 is available
  - We can handle (most?) local differences regarding volume sizes, scan strategies…
  - Data reduction is needed
  - Reads directly into BATOR

- Assimilation experiments
  - Impact experiments performed at SMHI, MET-Norway, KNMI
  - Technical experiments performed at DMI including data from around 40 radars from several countries (Denmark, Sweden, Norway, Finland, Estonia, Germany, The Netherlands, Belgium, France and Poland). Impact studies is underway.
  - Work started to include radar data in HARMONIE 4D-Var
  - MetCoOp: Passive mode over the summer, will be included actively in the test runs of cy38h1.2

- Radar data exchange
  - An operational data exchange among HIRLAM countries is established
  - Uses the BALTRAD servers with data sent to OPERA
  - Quality controlled with the BALTRAD toolbox before made available

BATOR

```
elif [ $base = "amsua" ] ; then
        # AMSU-A
        if [ $AMSUA_OBS -eq 1 ]; then
          echo "amsua    BUFR     amsua            ${YMD} ${HH}">>refdata
          ln -sf $WRK/oulan/amsua ./BUFR.amsua
          cp ${HM_LIB}/nam/param_bator.cfg.amsua.${ATOVS_SOURCE} ./param.cfg
        fi
elif [ $base = "amsub" ] ; then
        # AMSU-B
        if [ $AMSUB_OBS -eq 1 ]; then
          echo "amsub    BUFR     amsub            ${YMD} ${HH}">>refdata
          ln -sf $WRK/oulan/amsub ./BUFR.amsub
          cp ${HM_LIB}/nam/param_bator.cfg.amsub.${ATOVS_SOURCE} ./param.cfg
        fi
elif [ $base = "iasi"  ] ; then
        if [ $IASI_OBS -eq 1 ]; then
          echo "iasi     BUFR     iasi             ${YMD} ${HH}">>refdata
          ln -sf $WRK/oulan/iasi ./BUFR.iasi
          ln -sf ${HM_SAT_CONST}/iasichannels ./iasichannels
          cp ${HM_LIB}/nam/param_bator.cfg.${IASI_SOURCE} ./param.cfg
        fi
```

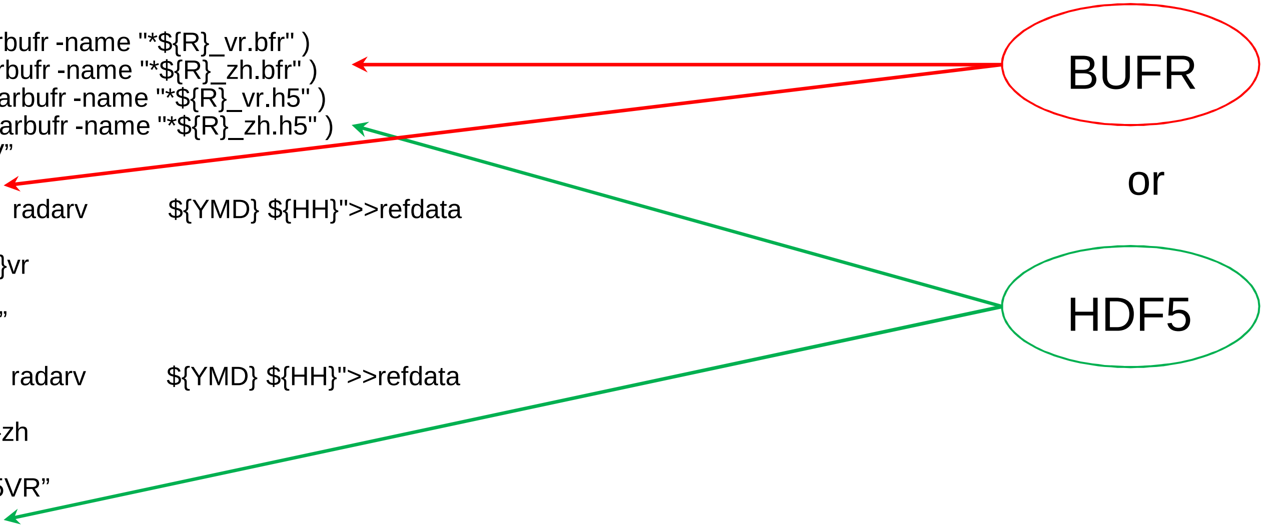
BATOR

```
elif [ $base = "radarv"  ] ; then
  if [ "$RADAR_OBS" -eq "1" ]; then
    cp ${HM_LIB}/nam/param_bator.cfg.radarv ./param.cfg
    echo -e "\n\n\n Radar\n\n\n”
    # Find available radars
    RADAR=''
    for F in $( ls -1 $WRK/radarbufr/ ) ; do
       R=$( echo $F | cut -d "_" -f1 )
       echo $RADAR | grep -q $R || RADAR="$RADAR $R”
    done
    for R in $RADAR ; do
       FV=$( find $WRK/radarbufr -name "*${R}_vr.bfr" )
       FR=$( find $WRK/radarbufr -name "*${R}_zh.bfr" )
       H5V=$( find $WRK/radarbufr -name "*${R}_vr.h5" )
       H5R=$( find $WRK/radarbufr -name "*${R}_zh.h5" )
       echo "Checking for $FV”
       if [ -e "$FV" ]; then
         echo "${R}vr  BUFR     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $FV ./BUFR.${R}vr
      fi
      echo "Checking for $FR”
       if [ -e "$FR" ]; then
        echo "${R}zh  BUFR     radarv           ${YMD} ${HH}">>refdata
        cat refdata
        ln -sf $FR ./BUFR.${R}zh
       fi
       echo "Checking for $H5VR”
       if [ -e "$H5V" ]; then
         echo "${R}vr  HDF5     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $H5V ./HDF5.${R}vr
       fi
      echo "Checking for $H5R”
      if [ -e "$H5R" ]; then
         echo "${R}zh  HDF5     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $H5R ./HDF5.${R}zh
      fi
      done
  else
     echo -e "\n\n\n No radar\n\n\n”
   fi #RADAR_OBS
```

BATOR

Wind and reflectivity in separete files (vr or zh)
Can also be in the same file

```
elif [ $base = "radarv"  ] ; then
  if [ "$RADAR_OBS" -eq "1" ]; then
    cp ${HM_LIB}/nam/param_bator.cfg.radarv ./param.cfg
    echo -e "\n\n\n Radar\n\n\n”
    # Find available radars
    RADAR=''
    for F in $( ls -1 $WRK/radarbufr/ ) ; do
       R=$( echo $F | cut -d "_" -f1 )
       echo $RADAR | grep -q $R || RADAR="$RADAR $R”
    done
    for R in $RADAR ; do
       FV=$( find $WRK/radarbufr -name "*${R}_vr.bfr" )
       FR=$( find $WRK/radarbufr -name "*${R}_zh.bfr" )
       H5V=$( find $WRK/radarbufr -name "*${R}_vr.h5" )
       H5R=$( find $WRK/radarbufr -name "*${R}_zh.h5" )
       echo "Checking for $FV”
       if [ -e "$FV" ]; then
         echo "${R}vr  BUFR     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $FV ./BUFR.${R}vr
      fi
      echo "Checking for $FR”
       if [ -e "$FR" ]; then
        echo "${R}zh  BUFR     radarv           ${YMD} ${HH}">>refdata
        cat refdata
        ln -sf $FR ./BUFR.${R}zh
       fi
       echo "Checking for $H5VR”
       if [ -e "$H5V" ]; then
         echo "${R}vr  HDF5     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $H5V ./HDF5.${R}vr
       fi
      echo "Checking for $H5R”
      if [ -e "$H5R" ]; then
         echo "${R}zh  HDF5     radarv           ${YMD} ${HH}">>refdata
         cat refdata
         ln -sf $H5R ./HDF5.${R}zh
      fi
      done
  else
     echo -e "\n\n\n No radar\n\n\n”
   fi #RADAR_OBS
```



I want to use radardata! What do I need to do?

- scr/include.ass

  ```
  export RADAR_OBS=1          #Radar
  ```

- scr/Radar2bufr
  - Point to the location of the radarfiles
  - Copy to the work directory with the correct naming (wmoid_zh.h5 or wmoid_vr.bfr)

- HDF5 libraries
  - Point to the libraries and includes in Env_system
  - Include these in the makeup config file

Env_system

```
module load hdf5/1.8.9

# Run time librariesexport
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$NETCDF_DIR/lib:/software/apps/hdf5/1.8.9/i1214/lib/


# HDF5
export HDF5_DIR=/software/apps/hdf5/1.8.9/i1214/
export HDF5_INC="-I$HDF5_DIR/include”
export HDF5_LIB="-L$HDF5_DIR/lib -lhdf5_fortran -lhdf5 -lz"
```

util/Makeup/config.mymachine

```
DEFS=-DLINUX -DLITTLE -DLITTLE_ENDIAN -DHIGHRES -DADDRESS64 -DPOINTER_64 -D_ABI64 -DBLAS \
    -DSTATIC_LINKING -DINTEL -D_RTTOV_DO_DISTRIBCOEF -DINTEGER_IS_INT \
    -DREAL_8 -DREAL_BIGGER_THAN_INTEGER -DUSE_SAMIO -D_RTTOV_DO_DISTRIBCOEF -DNO_CURSES \
    -DFA=fa -DLFI=lfi -DARO=aro -DOL=ol -DASC=asc -DTXT=txt -DUSE_HDF5

SYSLIBS= $(LD_SYS01) \
        $(LD_LANG01) $(LD_LANG02) $(LD_LANG03) $(LD_LANG04) $(LD_MPI01) \
        $(GRIB_API_LIB) \
        $(NETCDF_LIB) $(HDF5_LIB)

INCLDIRS=$(GRIB_API_INCLUDE) $(NETCDF_INC) $(HDF5_INC)

EXTMODS=grib_api,netcdf,hdf5,ifport
```

Radar2bufr

```
 # Input radar data
RADARECFS=ec:/smb/radardata




for COUNTRY in SE NO ; do
   case $COUNTRY in
      SE)
         # Just copy the files to the workdirectory since we read
         # the HDF5-files directly

         if [ -s $RADARECFS/$COUNTRY/*${DATE}T${HH}*.h5 ] ; then
            for RADARFILE in `ls $RADARECFS/$COUNTRY/*${DATE}T${HH}*.h5`; do
               radarID=$( echo $(basename $RADARFILE) | cut -d "_" -f1 )

               if [ $radarID = seang ] ; then
                  myWMOID=02606
                  cp $RADARFILE $WRK/radarbufr/${myWMOID}_zh.h5
               elif [ $radarID = sease ] ; then
                  myWMOID=02588
                  cp $RADARFILE $WRK/radarbufr/${myWMOID}_zh.h5
               fi
            done
         else
            touch $WRK/radarbufr/00000_zh.h5
         fi
      ;;
```

Radar2bufr

```
MET)
  # Just copy the files to the workdirectory since they are
  # already converted to mf-bufr

  if [ -s $RADARECFS/$COUNTRY/*_${DATE}${HH}*.bfr.gz ] ; then
    for RADARFILE in `ls $RADARECFS/$COUNTRY/*_${DATE}${HH}*.bfr.gz`; do
      myDATATYPE=$( echo $(basename $RADARFILE) | cut -d "_" -f2 )
      myWMOID=$( echo $(basename $RADARFILE) | cut -d "_" -f1 )
      if [ $myDATATYPE == "vr" ]; then
        cp $RADARFILE $WRK/radarbufr/${myWMOID}_vr.bfr.gz
      elif [ $myDATATYPE == "dbz" ]; then
        cp $RADARFILE $WRK/radarbufr/${myWMOID}_zh.bfr.gz
      else
        echo "Unknown datatype: $myDATATYPE"
        exit 1
      fi
    done
    gunzip $WRK/radarbufr/*.gz
  else
    touch $WRK/radarbufr/00000_zh.bfr
  fi
;;
```