

Quality Assurance Activities for VISTAS BC Processing

December 31, 2004

PROJECT TITLE: Processing of GEOS-CHEM output to generate CMAQ 2002 Boundary Conditions.

PROJECT PERIOD: September 15 - December 31, 2004

CONTRACTOR: Dr. Daewon W. Byun

CONTRACT: Subcontract No.: 06-12482D

SUBMITTED TO: Ralph E. Morris
Title: Principal
ENVIRON INTERNATIONAL CORPORATION

1. INTRODUCTION

This quality assurance document describes tasks performed by the Dr. Daewon Byun for the Environ International Corp in support of the CMAQ model simulation for the VISTAS. The objective is to ensure correct processing of the GEOS-CHEM global scale atmospheric chemistry model output to generate boundary conditions (BCs) for the CMAQ boundary conditions for year 2002 for the National RPO 36-km grid and the VISTAS layer heights. This document lists computer codes and scripts applied for the project.

2. DESCRIPTION OF DATA AND CODES

2.1 GEOS-CHEM Data Processed

Several versions of GEOS-CHEM outputs for year 2002 were processed in this project. First, the "STD" output includes the 22-layer output for January – August, 2002. This version of GEOS-CHEM output is called "GEOS_3". The STD output also includes the 20-layer output for September – September, 2002 and is called "GEOS_4". It seems that the main reason for the use of different formats of the GEOS-CHEM outputs are due to the change in the meteorological datasets available from the [Global Modeling and Assimilation Office](#) (GAMO) for year 2002. The vertical layer structures of the GAMO global scale meteorology data for GEOS_3 and GEOS_4 are different,

Further, Harvard provided additional simulation results obtained with the corrected NH3 emissions. The new output "STD_v02" includes the 20-layer output for GEOS_3 and GEOS_4 datasets.

2.2 GEOS2CMAQ Codes

For processing of the STD data, we utilized the original GEOS2CMAQ code version 1, developed by University of Houston Institute for Multidimensional Air Quality Studies (IMAQS).

For processing of the STD data for GEOS_3, GEOS_4, we re-compiled the GEOS2CMAQ version 1 with the new height data for the vertical layer structure, provided by Rokjin Park.

GOES-CHEM data and prepare GEOS2CMAQ codes for the processing of the GOES-CHEM output, to be provided by Dr. Rokjin Park.

The 2002 GOES-CHEM data are in two different formats, so-called GEOS_3 from January to August, 2002, and GEOS_4 from September to December 2002. The GEOS2CMAQ codes need to be modified to handles these two different formats and additional species. Different executable programs are generated for each chemical mechanism, for each data format and data source using separate “bldit” scripts. The program bldit script and run scripts used are provided at the end of this document. A simple User’s Guide for GEOS2CMAQ version 1 is also available.

There were problem processing the STD_v02 data with the original GEOS2CMAQ program because of changes in both the number of layers and the vertical layer structures because of the default assumptions used in the original codes.

Rojin Park updated the code and produced the GEOS2CMAQ Version 2 to resolve these problems. The new program is included at the in an accompanying “tar” file as well.

3. BOUNDARY CONDITON FILES

3.1 Output for CB-4 mechanism

Processing of STD GEOS_3 and GEOS_4 data were performed separately using different processors. Separate run scripts were generated for each month to clearly identify the starting and ending Julian date of the GEOS_CHEM output, which was stored in a separate directory for each month. Because of the need to refer to the file in another directory, separate run scripts were used. Next, STD_v02 GEOS_3 and GEOS_4 data were processed to generate day-specific 3-hourly BC files for each datasets. The two datasets were compared to ensure generation of proper BC files. There were various possibilities of making mistakes with the original GEOS2CMAQ data because they relied on the use of different include files during the compilation and there were no safety check if proper processors were used for different datasets. The GEOS2CMAQ version 2 reduced many of these risks by checking if the model configurations and input data were well coordinated.

3.2 Output for CB-4 mechanisms

Only STD_v02 GEOS_3 and GEOS_4 outputs were processed with GEOS2CMAQ Version 2.. GEOS-CHEM output will be processed to generate day-specific 3-hourly CMAQ BC output for SAPRC99 species. The output was further reviewed and compared with CB-4 BC files with PAVE.

3.3 Generation of monthly average BC files


```

    exit 1
  endif
  set platf = `uname -i | cut -d", " -f2`

#setenv FORTRAN f90
# set FC = /galaxy/local/forte/7/SUNWspro/bin/f90
set FC = /galaxy/local/studio/8/SUNWspro/bin/f90
set FP = /galaxy/local/forte/7/SUNWspro/bin/fpp

set xflag0 = "-xtarget=ultra3"
set xflag1 = "-xtarget=ultra3"
set xflag2 = "-xcache=16/32/1:1024/64/1"
set xflag3 = "-xcache=16/32/1:4096/64/1"
#set xopt = "-fast -O4"
set xopt = "-fast -xtarget=ultra3"
if ( $?Sun_debug ) then
  set XF_FLAGS = "-g -C -e -fnonstd"
  set LINK_FLAGS = "-g -C -e -fnonstd -xildoff"
else
  if ( $platf == "Ultra-60" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag3"
    set LINK_FLAGS = "-e $xopt $xflag1"
  else if ( $platf == "Ultra-30" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag2"
    set LINK_FLAGS = "-e $xopt $xflag1"
  else if ( $platf == "Ultra-2" ) then
    set XF_FLAGS = "-e $xopt $xflag0"
    set LINK_FLAGS = "-e $xopt $xflag0"
  else if ( $platf == "Sun-Blade-1000" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag3"
    set LINK_FLAGS = "-e $xopt $xflag1"
  else
    set XF_FLAGS = "-e $xopt"
    set LINK_FLAGS = "-e $xopt."
  endif
endif

set F_FLAGS = "$XF_FLAGS -I."
set CPP_FLAGS = "-fixed -e"

set MOD = ${M3TOOLS}/stenex/$BLD_OS

set F_FLAGS = "$XF_FLAGS -I. -M${MOD}"
set CPP_FLAGS = "-fixed -e"
#####

set LIB1 = "-L${M3TOOLS}/m3io/$BLD_OS -lm3io"
set LIB2 = "-L${M3TOOLS}/netCDF/$BLD_OS -lnetcdf"
set LIB3 = "-L${MOD} -lsef90_noop"
set LIBS = "$LIB1 $LIB2 $LIB3"

```

```

set Bldr = $M3TOOLS/build/$BLD_OS/m3bld

set Domain = d${Reso}_${Layers}
set ICL_CONST = $GlobInc
set ICL_IOAPI = "${M3TOOLS}/m3io"
set ICL_FILES = $GlobInc
set ICL_GRID = $GlobInc/cmaq_v1_grids/$Domain
set ICL_GRID1 = /galaxy/home/aqm/nkmoon/GEOS-CHEM/LINK_CODE/GRID/$Domain
set ICL_MECH = $GlobInc/$Mechanism

set Popt = NOOP
set STX = ( -DF90\
    -DSUBST_MODULES=${Popt}_MODULES\
    -DSUBST_GLOBAL_SUM=${Popt}_GLOBAL_SUM )
#set STX =

## setenv CVSROOT $Project

# make the config file

cat >! $CFG.bld << E+O+X

model    $MODEL;

FPP      $FP;

cpp_flags "$CPP_FLAGS $STX";

f_compiler $FC;

f_flags  "$F_FLAGS";

link_flags "$LINK_FLAGS";

libraries "$LIBS";

global   $Opt;

// domain and mechanism: ${Domain} ${Mechanism}
// project archive: ${Project}

include SUBST_CONST    $ICL_CONST/CONST.EXT;
include SUBST_IOPARMS  $ICL_IOAPI/PARMS3.EXT;
include SUBST_IOFDESC  $ICL_IOAPI/FDESC3.EXT;
include SUBST_IODECL   $ICL_IOAPI/IODECL3.EXT;
include SUBST_XSTAT    $ICL_IOAPI/XSTAT3.EXT;
include SUBST_COORD_ID $ICL_GRID1/COORD_${Grid}.EXT;
include SUBST_HGRD_ID  $ICL_GRID1/HGRD_${Grid}_${Decomp}.EXT;
include SUBST_VGRD_ID  $ICL_GRID1/VGRD_${Layers}.EXT;
include SUBST_MECHANISM ${Base}/src/GEOS2_${Mechanism}_BC.EXT;

```

```

misc ;
  ${Base}/src/geos2cmaq.F
  ${Base}/src/bcon.F
  ${Base}/src/comheader.F
  ${Base}/src/lat_lon.F
  ${Base}/src/get_tau.F
  ${Base}/src/errmsg.F
  ${Base}/src/geos_tools.F
  ${Base}/src/read_chem.F

E+O+X

# make the model executable

if ( $?MakeOpt ) then
  $Blder -make ${CFG}.bld
else
  set NoMake
  $Blder ${CFG}.bld
endif
if ( $status != 0 ) then
  echo " *** failure in $Blder ***"
  exit 1
endif
if ( -e "$Base/${CFG}" ) then
  echo " >>> previous ${CFG} exists, re-naming to ${CFG}.old <<<"
  unalias mv
  mv $Base/${CFG} $Base/${CFG}.old
endif
cp ${CFG}.bld $Base/${CFG}
if ( ( $Opt != no_compile ) && \
    ( $Opt != no_link ) && \
    ( $Opt != parse_only ) && \
    ( $Opt != show_only ) && \
    $?NoMake ) then
  mv $MODEL $Base
endif

exit

```

4.2 Sample run script for GEOS2CMAQ version 1: run.geos2cmaq_01

```

#!/bin/csh -f

# Sample GridEngine submission script for MPI
# -----
# my own name for this job (output filenames prefixed with this)
#$ -N geos2cmaq

```

```
# hold the execution of this process until the previous if done
##$ -hold_jid

# don't send me mail when it's finished
#$ -M dwbyun
#$ -m n

# I want to use csh as my processing shell
#$ -S /usr/bin/csh

# ccount to charge for this job (doesn't do anything yet)
#$ -A tester

# Execute the job from the current working directory
#$ -cwd

# Parallel environment request
#$ -l cre

# Here is where you set then number of CPU's you want!
# PE_name CPU_Numbers_requested
##$ -pe hpc 2
#$ -pe core 1

# initialize the local environment
source /galaxy/home/aqm/nkmoon/setenv_m3_kodos
env

# -----
# RCS file, release, date & time of last delta, author, state, [and locker]
# $Header$

date; set timestamp; echo " "; set echo
#-----
# 2001 | 01:001, 02:032, 03:060, 04:091
# 2001 | 05:121, 06:152, 07:182, 08:213, 09:244, 10:274, 11:305, 12:335
#-----

#set Base = $cwd
set Base = /home/dwbyun/rjkdir/BCON_rjp
# set Mechanism = cb4
set Mechanism = saprc99
#
set year = 2002
set month = 01
set monthn = 01
```

```

set firstday = 01
set NDATE = 00

# while ( $NDATE <= 02 )
while ( $NDATE <= 31 )

    @ NDATE1 = $NDATE + 1

    if ( $NDATE < 10 ) then
        set day = ${month}0${NDATE}
        set nextday = ${month}0${NDATE1}
        if ( $NDATE == 9 ) then
            set nextday = ${month}${NDATE1}
        endif
    else
        set day = $month$NDATE
        set nextday = $month${NDATE1}
        if( $day == 0131 ) then
            set monthn = 02
            set nextday = 02
        endif
    endif
endif

    @ julianday = ($firstday - 1) + $NDATE

    if ( $julianday < 10 ) then
        set julianday = 00$julianday
    else if ( $julianday < 100 ) then
        set julianday = 0$julianday
    endif

#-----
# set executable
# Note -- do not forget to change the mechanism name
#-----
# set EXEC = BCON_VISTAS_${Mechanism}_v3
# set EXEC = BCON_VISTAS_${Mechanism}_v3_120
#set EXEC = BCON_VISTAS_SAPRC_v3_120

#-----
# OUTDIR : Output directory
#-----
#set OUTDIR = /cxfs/work/aqm/nkmoon/LINK/out/EPA36km/${year}${month}_TEST
#set OUTDIR = /cxfs/work/aqm/nkmoon/LINK/out/ICAP108km/${year}${month}
#set OUTDIR = /cxfs/work/aqm/rjp/LINK/out/VISTAS/SPRING_new/${year}${month}
set OUTDIR = ${Base}/OUT/${year}${month}

```

```

if ( ! -d "$OUTDIR" ) mkdir -p $OUTDIR
#-----
# Output file name
#-----

set OUTFL1 = BCON.VISTAS36_CCTM_${Mechanism}.${year}${julianday}.ncf

#-----
# Setup GEOS_CHEM BCON output
#-----
setenv BCON      $OUTDIR/$OUTFL1

#-----
# JUDATE : INITIAL JULIAN STUDY DATE (YYDDD)
# HSTRT  : STARTING HOUR OF JUDATE FOR STUDY
# HTORUN : NUMBER OF ' HOURS ' TO USE FROM MM5
# HTOSKIP: NUMBER OF ' HOURS ' TO SKIP FIRST
#-----
setenv JUDATE ${year}${julianday}
setenv HSTRT  00
setenv HTOSKIP 00
setenv HTORUN 09

#-----
# INPDIR : Input directory
#-----
#set INPDIR1 = /galaxy/home/aqm/nkmoon/GEOS-
CHEM/LINK_CODE/DATA/conc/${year}${month}
#set INPDIR1 = /galaxy/home/aqm/nkmoon/GEOS-CHEM/LINK_CODE/DATA/conc
set INPDIR1 = /galaxy/home/aqm/rjp/VISTAS/STD_v02/GEOS_3/${year}${month}
set INPDIRN = /galaxy/home/aqm/rjp/VISTAS/STD_v02/GEOS_3/${year}${month}
set INPDIR2 = /galaxy/home/aqm/nkmoon/GEOS-CHEM/LINK_CODE/DATA/domain
set INPDIR3 = /galaxy/home/aqm/nkmoon/GEOS-
CHEM/LINK_CODE/DATA/ver_geos
set INPDIR4 = /galaxy/home/aqm/nkmoon/GEOS-CHEM/LINK_CODE/DATA/aird

#-----
# Set up point emission binary and ascii file
#-----
setenv GEOSCHEM1  $INPDIR1/ts${year}${day}.bpch
setenv GEOSCHEM2  $INPDIRN/ts${year}${nextday}.bpch
#setenv VERTICAL_OUT $INPDIR2/GRIDCRO3D_020101
#setenv VERTICAL_OUT $INPDIR2/GRIDCRO3D_ICAP_Pacific_2001001
setenv VERTICAL_OUT
$INPDIR2/GRIDCRO3D.Jan02_19L.px_acm2.36km.20020101

```



```

set Bld = $Base/BLD_$APPL
#unset echo
if ( ! -e "$Bld" ) then
  mkdir $Bld
else
  if ( ! -d "$Bld" ) then
    echo " *** target exists, but not a directory ***"
    exit 1
  endif
endif
#set echo
cd $Bld

set BLD_OS = `uname -s` `uname -r | cut -d"." -f1`
if ( $BLD_OS != 'SunOS5' ) then
  echo " $BLD_OS -> wrong bldit script for host!"
  exit 1
endif
set platf = `uname -i | cut -d"," -f2`

#setenv FORTRAN f90
# set FC = /galaxy/local/forte/7/SUNWspro/bin/f90
set FC = /galaxy/local/studio/8/SUNWspro/bin/f90
set FP = /galaxy/local/forte/7/SUNWspro/bin/fpp

set xflag0 = "-xtarget=ultra3"
set xflag1 = "-xtarget=ultra3"
set xflag2 = "-xcache=16/32/1:1024/64/1"
set xflag3 = "-xcache=16/32/1:4096/64/1"
#set xopt = "-fast -O4"
set xopt = "-fast -xtarget=ultra3"
if ( $?Sun_debug ) then
  set XF_FLAGS = "-g -C -e -fnonstd"
  set LINK_FLAGS = "-g -C -e -fnonstd -xildoff"
else
  if ( $platf == "Ultra-60" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag3"
    set LINK_FLAGS = "-e $xopt $xflag1"
  else if ( $platf == "Ultra-30" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag2"
    set LINK_FLAGS = "-e $xopt $xflag1"
  else if ( $platf == "Ultra-2" ) then
    set XF_FLAGS = "-e $xopt $xflag0"
    set LINK_FLAGS = "-e $xopt $xflag0"
  else if ( $platf == "Sun-Blade-1000" ) then
    set XF_FLAGS = "-e $xopt $xflag1 $xflag3"
  endif
endif

```



```

FPP      $FP;

cpp_flags "$CPP_FLAGS $STX";

f_compiler $FC;

f_flags  "$F_FLAGS";

link_flags "$LINK_FLAGS";

libraries "$LIBS";

global  $Opt;

// domain and mechanism: ${Domain} ${Mechanism}
// project archive: ${Project}

include SUBST_CONST      $ICL_CONST/CONST.EXT;
include SUBST_IOPARMS   $ICL_IOAPI/PARMS3.EXT;
include SUBST_IOFDESC   $ICL_IOAPI/FDESC3.EXT;
include SUBST_IODECL    $ICL_IOAPI/IODECL3.EXT;
include SUBST_XSTAT     $ICL_IOAPI/XSTAT3.EXT;
include SUBST_COORD_ID  $ICL_GRID1/COORD_${Grid}.EXT;
include SUBST_HGRD_ID   $ICL_GRID1/HGRD_${Grid}_${Decomp}.EXT;
include SUBST_VGRD_ID   $ICL_GRID1/VGRD_${Layers}.EXT;
include SUBST_MECHANISM ${Base}/src_v02/GEOS2_${Mechanism}_BC.EXT;

misc ;
  ${Base}/src_v02/geos2cmaq.F
  ${Base}/src_v02/bcon.F
  ${Base}/src_v02/comheader.F
  ${Base}/src_v02/lat_lon.F
  ${Base}/src_v02/get_tau.F
  ${Base}/src_v02/errmsg.F
  ${Base}/src_v02/geos_tools.F
  ${Base}/src_v02/read_chem.F

E+O+X

# make the model executable

if ( $?MakeOpt ) then
  $Blder -make ${CFG}.bld
else
  set NoMake
  $Blder ${CFG}.bld

```

```
endif
if ( $status != 0 ) then
  echo " *** failure in $Blder ***"
  exit 1
endif
if ( -e "$Base/${CFG}" ) then
  echo " >>> previous ${CFG} exists, re-naming to ${CFG}.old <<<"
  unalias mv
  mv $Base/${CFG} $Base/${CFG}.old
endif
cp ${CFG}.bld $Base/${CFG}
if ( ( $Opt != no_compile ) && \
  ( $Opt != no_link ) && \
  ( $Opt != parse_only ) && \
  ( $Opt != show_only ) && \
  $?NoMake ) then
  mv $MODEL $Base
endif

exit
```