

# NWPSAF 1D-Var User Manual

Fiona Smith, Met Office, Exeter, UK

NWPSAF-MO-UD-032

Version 1.1: 9<sup>th</sup> December 2016

*This software and documentation was developed within the context of the EUMETSAT satellite Application Facility (NWP SAF). The partners in the NWP SAF are the Met Office, ECMWF, KNMI, and Météo-France.*

## Appendix E. Instructions for using the observation simulation code

With release 1.1 of the NWPSAF-1DVar, support has been provided for simulating observations, and for converting existing observations between brightness temperatures or radiances and PC scores and back to reconstructed spectra.

The relevant code can be found in the directory `src/sim_spec`. Four programs are provided:

- `sim_spec_rttov11.f90`
- `sim_spec_rttov12.f90`
- `spec_to_pc.f90`
- `pc_to_spec.f90`

### Simulating observations

The two `sim_spec` routines read in profiles in the standard format for the 1D-Var, and run RTTOV (or PC-RTTOV) to produce observation files in the standard format. Outputs are controlled by a series of switches set in the code itself. The relevant settings that can be changed are:

```

LOGICAL, parameter :: add_error=.true.           !This controls whether noise from the Rmatrix is used
LOGICAL, parameter :: add_bias=.false.          !This controls whether bias is added per chan (BTs only)
LOGICAL, parameter :: usepcs=.true.             !This controls whether PC-RTTOV is used
LOGICAL, parameter :: produceradiances=.false. !This sets the RTTOV variable switchrad
LOGICAL, parameter :: reconstruct=.true.        !If PC-RTTOV is used, set this true to produce reconstructed radiances/BTs
Integer, parameter :: surftype=1                !0=land 1=sea 2=snow/ice Only one setting per input profile file at the moment

CHARACTER(len=200) :: biasfile='/path/to/bias_for_BTs_only.txt'
CHARACTER(len=200) :: noisefile='/path/to/1DVar/Rmatrix'
CHARACTER(len=200) :: profilefile='/path/to/1DVar/background_profiles.dat'
CHARACTER(len=200) :: outfile='/path/to/SimObs.dat'
    
```

Note that the output directory should already exist.

In addition, the following example information should be modified to match the input files:

```

integer(kind=jpim), parameter :: sat_id=4
integer(kind=jpim), parameter :: max_iasi_channels=8461 ! Total number of IASI channels
integer(kind=jpim), parameter :: idim=4348             ! Max. no. of profiles to read
integer(kind=jpim), parameter :: nwlevels=70          ! No. of model levels
integer(kind=jpim), parameter :: nwlayers=69          ! No. of model levels - 1
integer(kind=jpim), parameter :: nslay = 4            ! No. of model surface layers
!If PC-RTTOV is used
INTEGER(kind=jpim), PARAMETER :: ipcreg=3             !Number of PC predictors
INTEGER(kind=jpim), PARAMETER :: npcscs=200           !number of PC scores
    
```

### Types of data output

Output quantity	Setting for usepcs	Setting for produceradiances	Setting for reconstruct
Brightness Temperature	.false.	.false.	.false.
Radiance	.false.	.true.	.false.
PC Score	.true.	N/A	.false.
Reconstructed BT (looks like BT in the output file)	.true.	.false.	.true.
Reconstructed Rad (looks like Rad in the output file)	.true.	.true.	.true.

Note that Reconstructed spectra are not yet supported in the 1D-Var: although you could use these quantities as input anyway, your observation error covariance structure is likely to be incorrect relative to the observations and/or forward calculations, and this could cause problems with minimisation.

If `add_noise=.true.`, a Gaussian error is added for each channel separately, consistent with the input R-matrix file. For PC score simulations, a random Gaussing error with `s.d.=1` is applied. Note that this is just to allow experimental use of this code; the effects may not be realistic. The best way to create a noisy PC score simulation is to simulate radiances, add noise in radiance space, and then convert to PC Scores (using code provided as described below). Note that in a simulated experimental setting, where only instrument noise is assumed in the R-matrix, you should simulate Reconstructed Radiances with PC-RTTOV before adding the noise; using normal RTTOV adds an additional forward model error that can cause erratic minimisation.

### Converting between spectra and PC scores

The routine `spec_to_pc.f90` converts observations in standard 1D-Var input format from spectra (BT, Radiance, Reconstructed BT or Reconstructed Rad) to PC scores. The input units are determined by the "Units: " line in the header of the input file. If no "Units: " line is present, BTs are assumed.

If the input is in BT, these are first converted to radiance units, then to PC scores. The settings that may need to be changed in this file are:

```
CHARACTER(LEN=*), PARAMETER :: infile = '/path/to/infile.dat'
CHARACTER(LEN=*), PARAMETER :: outfile = '/path/to/outfile.dat'
INTEGER, PARAMETER :: NPCScores=200
INTEGER, PARAMETER :: ipcreg=3
INTEGER, PARAMETER :: ipcbnd=1
CHARACTER(LEN=*), PARAMETER :: coeffile = '/path/to/rtcoef_metop_2_iasi.H5'
CHARACTER(LEN=*), PARAMETER :: pcfile = '/path/to/pccoef_metop_2_iasi.dat'
```

The routine `pc_to_spec.f90` converts observations in standard 1D-Var input format from PC scores to Reconstructed BT or Reconstructed Radiances.

The following settings may need to be changed:

```
CHARACTER(LEN=*), PARAMETER :: infile = '/path/to/infile.dat'
CHARACTER(LEN=*), PARAMETER :: outfile = '/path/to/outfile.dat'
INTEGER, PARAMETER :: ipcreg=3
INTEGER, PARAMETER :: ipcbnd=1
CHARACTER(LEN=*), PARAMETER :: coeffile = '/path/to/rtcoef_metop_2_iasi.H5'
CHARACTER(LEN=*), PARAMETER :: pcfile = '/path/to/pccoef_metop_2_iasi.dat'
LOGICAL :: convert_to_bt=.false.
```

If `convert_to_bt=.false.`, the output will be reconstructed radiances.

### Compiling and running the code

The four programs do not at present have namelist inputs, so options need to be set in the code, which means it needs to be recompiled, but in other senses the procedure is like the main 1D-Var code, using the same makefile to compile the code and putting the executables in the same build directory.

To compile, simply go into the `build` directory, and type:

```
make x
```

where `x` is the name of the program you want to compile.

Example scripts are provided in `WorkDir` to run the routines (namely `Run_Simspec.ksh` and `Run_Convert.ksh`) but these are very simple since all settings are done inside the source code itself.

Example output from these routines can be found in `Sample_ObsFiles/Sim_RTT0V11` and `Sample_ObsFiles/Sim_RTT0V12`