

National Earth Observation Group

Landsat NBAR Product Generation System

― Software design and implementation notes

|  |  |
| --- | --- |
| Author: | Frank Q. Fu |
| Date Created: | 04/03/2010 |
| Date Modified: | 27/05/2010 |
| Current Version: | 0.2 |
| TRIM Number: | D2010-36176 |

Table of Contents

[1. Introduction 1](#_Toc263336486)

[2. Description of Landsat5/7 datasets 1](#_Toc263336487)

[3. Business Requirements 2](#_Toc263336488)

[2.1. Programming Languages and Platform 2](#_Toc263336489)

[2.2. Inputs 2](#_Toc263336490)

[2.3. Dataset ID 3](#_Toc263336491)

[2.4. Outputs 3](#_Toc263336492)

[4. System Design and Development 4](#_Toc263336493)

[3.1. Object and Classes Diagrams 4](#_Toc263336494)

[a) Data Structure 4](#_Toc263336495)

[b) NBAR Parser 4](#_Toc263336496)

[c) NBAR Processor 4](#_Toc263336497)

[d) NBAR Package 4](#_Toc263336498)

[3.2. Main Components 5](#_Toc263336499)

[3.3. Future Extension of the packaging system 5](#_Toc263336500)

[3.4. Physical Structure 6](#_Toc263336501)

[3.5. Python Modules and classes 6](#_Toc263336502)

[3.6. C Programs 7](#_Toc263336503)

[3.7. Shell Scripts 7](#_Toc263336504)

[4. Maintenance Notes 8](#_Toc263336505)

[4.1. Installation of the program 8](#_Toc263336506)

[4.2. Usage and execution of the program 11](#_Toc263336507)

[4.3. Input and output directories 11](#_Toc263336508)

[4.4. Log file 12](#_Toc263336509)

[4.5. Lock File 12](#_Toc263336510)

[4.6. Post-implementation bugs and problems 12](#_Toc263336511)

[5. Appendix 12](#_Toc263336512)

[A. Temporary/intermediate files extracted from inputs 12](#_Toc263336513)

[B. Final output files of a NBAR dataset packaged 14](#_Toc263336514)

# Introduction

Landsat nadir BRDF-adjusted reflectance (NBAR) product generation system is part of NEO NBAR project, aiming to deliver an NBAR capability that can be run in a fully automated environment for images that have been acquired during the ‘MODIS era’ (post April 2000).

The scientific algorithm of the system and the corresponding FORTRAN scripts were developed by Fuqin Li of NEO Science Team. The model has been manually tested against ground based reflectance measurements for selected validation sites. In order to make this system operational on the production site, new supplementary modules have been developed and the whole system has been integrated successfully for automatic processing.

The design and development of the system was initially conducted by Paul Gardner and Lan-Wei Wang starting from late 2009. Paul Gardner completed the first part of the system for parsing and extracting parameters from various inputs.

# Description of Landsat5/7 datasets

Geoscience Australia receives and processes data from the Landsat series of satellites. However, at current stage, the Landsat NBAR generation system only processes Landsat-5 TM and Landsat-7 ETM+ data. The Thematic Mapper (TM) is a sensor onboard the Landsat 4 and Landsat-5 spacecrafts. It provides information on the Earth's surface in the visible, near, middle and thermal infrared regions of the electromagnetic spectrum. Landsat-7 has the new Enhanced Thematic Mapper Plus (ETM+) sensor. This sensor has the same 7 spectral bands as its predecessor, TM, but has an added panchromatic band with 15-metre resolution and a higher resolution thermal band of 60 metres. The ETM+ sensor also has a five percent absolute radiometric calibration. The satellite and image characteristics of the TM and ETM+ are summarized in Table 1 and 2.

Table 1. The radiometric characteristics of the ETM+ and TM sensors

|  |  |  |  |
| --- | --- | --- | --- |
| **Band No** | **Spectral Range (in Microns)** | **EM Region** | **Generalised Application Details** |
| **1** | 0.45 - 0.52 | Visible Blue | Coastal water mapping, differentiation of vegetation from soils |
| **2** | 0.52 - 0.60 | Visible Green | Assessment of vegetation vigour |
| **3** | 0.63 - 0.69 | Visible Red | Chlorophyll absorption for vegetation differentiation |
| **4** | 0.76 - 0.90 | Near Infrared | Biomass surveys and delineation of water bodies |
| **5** | 1.55 - 1.75 | Middle Infrared | Vegetation and soil moisture measurements; differentiation between snow and cloud |
| **6** | 10.40- 12.50 | Thermal Infrared | Thermal mapping, soil moisture studies and plant heat stress measurement |
| **7** | 2.08 - 2.35 | Middle Infrared | Hydrothermal mapping |
| **8** | 0.52 - 0.90 (panchromatic) | Green, Visible Red, Near Infrared | Large area mapping, urban change studies |

Table 2. Other satellite and image characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | | **Landsat-7 ETM+** | **Landsat-5 TM** |
| **Ground Sampling Interval (GSI) (pixel size)** | **Bands 1-5 & 7 Band 6 Band 8** | 30 - 30 m 60 - 60 m 15 - 15 m pixel size (18 - 18 m GSI)\* | 30 - 30 m 120 - 120 m N/A |
| **Swath width** | | 185 km | 185 km |
| **Repeat coverage interval** | | 16 days (233 orbits) | 16 days (233 orbits) |
| **Altitude** | | 705 km | 705 km |
| **Quantisation** | | Best 8 of 9 bits | 8 bits (256 levels) |
| **On-board data storage** | | 375 Gb (solid state) | Magnetic tape failed |
| **Orbit type** | | Sun-synchronous | Sun-synchronous |
| **Inclination** | | 98.2 | 98.2 |
| **Equatorial Crossing** | | Descending node: 10:00 am | Descending node: 10:10 am |
| *\* ETM+ band 8 (panchromatic) was designed to be acquired at 15m resolution, but post-launch testing shows a ground sampling interval closer to 18m.* | | | |

# Business Requirements

The main business requirements for Landsat NBAR generation system were defined in the project scope document “NBAR Project Approach” (Trim: D2010-16918). More detailed requirements came from discussions, emails and decisions made on a series of project meetings. However, from IT technique point of view, these initial requirements are not sufficient enough for designing and development of the system. Further detailed requirements were gradually finalised through development, testing and validation. As a result, the output data formats and physical structure of the system have been changed and re-structured in according with the changes in requirements.

## Programming Languages and Platform

The key scientific algorithm of the system was implemented in FORTRAN by Fuqin Li while the main driver and controlling modules were in Python. Nine Shell scripts, developed by Fuqin Li and Lan-Wei Wang, were used to communicate between the Python and FORTRAN modules. A couple of C programs were developed for the extraction of ancillary parameters and the generation of quick-look JPG images. The system can be run on a stand-alone Linux environment and has been successfully tested on acr111, ausimages4 and ANU NCI Linux machines.

## Inputs

The inputs to the NBAR processing system are Landsat-5 TM and Landsat-7 ETM+ datasets packaged in NEO (ACRES) format. Each dataset contains 7 image data files (bands 1-7), a quick-look JPG image, a report text file (ACRES old version), a readme text file and other ancillary files. All files are stored in a directory named by a “work order” number as they are manually ordered through NEO Operations. The detailed folder structure and files contained are shown in Figure 1.

Figure 1. Physical structure and contents of an input dataset with a work order number of “07889\_01”

**07889\_01**

label.txt

md5sum.txt

QACheckList.txt

readme.txt

**scene01**

07889\_01.jgw 07889\_01.jpg

band1.dat band2.dat

band3.dat band4.dat

band5.dat band6.dat

band7.dat header.dat

report.txt

## Dataset ID

The dataset ID, different from the work order number, is a unique string used to identify a dataset. It is a predefined format of

*sss\_nnn\_ fff\_ ggg\_hhh\_ ppp\_rrr\_yyyymmdd*

where

*sss* is the satellite ID, eg., LS5 stands for Landsat-5 and LS7 for Landsat-7,

*nnn* is the sensor ID, whose value is either “TM” or “ETM”

*fff* is the product code, eg., NBAR

*ggg* is the process code, eg., P54

*hhh* is the product source/group, eg., GAV1

*ppp* is the path number,

*rrr* is the row number, and

*yyyymmdd* is the acquisition date of the dataset.

For examples, LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522 and LS7\_ETM\_OTH\_P54\_NBAR\_094\_084\_20090522 are valid dataset IDs, representing a Landsat-5 TM and a Landsat-7 ETM+ ortho-corrected NBAR datasets respectively. For both datasets, the path numbers are 094, row numbers 084 and acquisition dates are 22 May 2009.

## Outputs

The outputs of Landsat NBAR processing system for each dataset are stored in a newly-created directory named by its dataset ID. A label.txt, readme.txt, metadata.xml, jpg image and md5sum.txt are residing in the dataset directory. The thermal Band 6 image data files are not included in the final NBAR products as it was a decision made by NEO management. Therefore, the rest six NBAR bands (Band 1, 2, 3, 4, 5, and 7) are outputted to a multiple-band ENVI file, which, together with a ENVI header file and a report.txt file, are stored in the “scene01/” subdirectory. The coloured quick-look JPEG images are created with bands 7, 4, and 1 as the RGB colour components respectively. The physical structure of output datasets is similar to that of inputs (Figures 2).

However, in order to consistent with other operation systems such as ODPJM, it is required that the final directory structure of a packaged dataset should included within a root directory named by the dataset ID plus “\_FOR\_ARCHIVE”. Both the metadata.xml and the quick-look jpg image are copied to this root directory to easy the subsequent archiving processes. The extra part of the directory structure is coloured in green in Figure 2 as it seems to be redundant and may need to be removed later.

**scene01**

LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522.envi

LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522.hdr

report.txt

**LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522**

label.txt

readme.txt

metadata.xml

LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522.jpg

md5sum.txt

Figure 2. Physical structure and contents of an output NBAR dataset after packaged

**LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522\_FOR\_ARCHIVE**

metadata.xml

LS5\_TM\_OTH\_P54\_NBAR\_094\_084\_20090522.jpg

# System Design and Development

Based on the business requirements, the system was designed and developed using the object-oriented methodology. However, the design and development of the system was not completed in a single cycle as more requirements and specifications were added or changed during the development life cycle, resulting in the re-structure and optimisation of the code accordingly. We here used the *Reverse Engineering of Software* methodology to derive UML class diagrams from existing python code for a better understanding the system. A collection of classes were identified and their relationships are drawn in UML.

## Object and Classes Diagrams

### Data Structure

There is only one data structure “NbarData” class which models the properties of a NBAR dataset object. It defines a set of metadata and parameters that are used to describe a Landsat 5/7 scene. Most of these metadata and parameters are defined in the EODS tables.

### NBAR Parser

The NbarParser implements a set of methods for parsing and extracting metadata from various data image and ancillary input files. The metadata and parameters parsed are stored in an NbarData object. Meanwhile, the parser also generates a set of temporary text files which are the inputs to some core FORTRAN programs invoked by NbarProcessor to conduct the NBAR generation processing.

### NBAR Processor

The NbarProcessor controls the main procedure of nine core FORTRAN programs. The failure of any FORTRAN programs will result in the exit of the whole NBAR processing. In some cases it may force the system to re-run certain FORTRAN programs with different inputs. Upon successful in generating NBAR products, the NbarProcessor will also conduct clean up jobs: remove temporary files, rename data image files and change file access permissions.

### NBAR Package

The NbarPackage implements a set of methods for generating ancillary files and packaging all final products in a directory structure shown in Figure 2. The six NBAR bands will be incorporated into a multi-band ENVI file and an ENVI header is to be generated accordingly. A metadata.xml and a report.txt files will be created using the parameters extracted or predefined by the system.

## Main Components

1. Input

The inputs to the system are a series of datasets identified by unique work orders. These datasets may be either a Landsat-5 TM scene or a Landsat-7 ETM+ scene (see Fig 1).

1. Parsing and Extraction

For each input work order, the system will invoke a parser which extracts metadata and parameters from relevant input files, such as ozone, MODIS BRDF HDF files. These metadata and parameters will be written into temporary text files or stored in the NbarData object (data structures).

1. NBAR Processing

After successful metadata extraction and input preparation, the program will invoke the 9 Shell scripts, each of which executes a Fortran program accordingly. Note that the third FORTRAN program is Modtran4.exe, commercial software purchased from external company. This program may take up to 30-45 minutes to run the whole process for a single dataset. It also needs inputs from a pre-provided directory “DATA/” and a text file “modroot.in”.

1. Dataset Packaging

After the generation of NBAR image files in the dataset directory named by its dataset ID, the system will package relevant outputs in specified format so that it can be archived into EODS. The packaging processes include the following steps:

1. create a new directory ‘scene01/’;
2. create jpeg image over band 7, 4, and 1 as RGB components;
3. copy readme.txt;
4. generate label.txt;
5. generate metadata.xml;
6. generate report.txt storing in directory ‘scene01/’;
7. generate multi-band ENVI file and header file storing in directory ‘scene01/’;
8. generate md5sum.txt.

## Future Extension of the packaging system

Although the current version of the NBAR processing and packaging system is designed and implemented for Landsat 5/7 datasets, it has the potential to be extended to include other satellite data as well, as it was designed and developed using object-oriented methodology.

## Physical Structure

NBAR\_PRJ

**src**

**script**

runangle\_ortho\_bin

runbinear\_l5\_ortho

runbrdf\_allband\_ortho\_bin

runcoeffiecient

runflux

runmodtraninput\_netcdf

runmodtran\_netcdf\_lanwei

runmodtranoutput

runtrans

md5checksum.bsh

bin2jpg.c

bin2jpg.h

extractDemHeight.c

extractDemHeight.h

extractVapour.c

extractVapour.h

JpegUtil.c

JpegUtil.h

extractOzone.c

extractOzone.h

HdfExtractor.c

Makefile.ffu

Makefile

**lib**

Figure 4. Physical structure of Landsat NBAR processing and package system

**bin**

**log**

runNbarProcessor

(main driver)

angle\_ortho\_bin

coefficient

input\_modtran\_ortho

read\_modtrancor\_ortho

bin2jpg

extractDEMhight

Mod4v3r1\_F90.exe

refort\_tp5\_ga

binear\_ortho

extractOzone

read\_flx\_ga refort\_tp5\_ga\_trans

brdf\_sim\_bin

extractVapour

read\_flx\_ga\_trans

hdfExtractor

read\_modtran

NbarProcessor.py

NbarPackage.py

NbarParser.py

NbarData.py

NbarUtils.py

NbarLog.py

NbarConfig.py

XmlWriter.py

isotime.py

julian.py

**python**

landsat\_nbar.lck

problem.log

workorder.log ... ...

hdf\_c\_4.2r1-linux

jpeg

libgeotiff-1.2.4

**DATA**

**config**

**fortran**

**tmp**

angle\_ortho\_bin.f binear\_ortho.f

brdf\_sim\_bin.f coefficient.f

input\_modtran\_ortho.f read\_flx\_ga.f

read\_flx\_ga\_trans.f read\_modtran.f

read\_modtrancor\_ortho.f refort\_tp5\_ga.f

refort\_tp5\_ga\_trans.f Makefile

The Landsat 5/7 Package System is in *“/net/acr111/image10/stripe/NBAR\_PRJ/*”. The main driver python file “*runNbarProcessor*” is in this directory. There are at least 10 sun-directories (Fig 4): (a) “*python*/” contains all Python source codes ; (b) “*bin*/” contains 17 executable C or FORTRAN mini-programs; (c) “*log/”* is used to store a problem.log and a lock file (p6\_packaging.lck), all other run-time log files are to be stored in this subdirectory; (d) “*shell*/” contains 10 Shell scripts that are used to invoke Fortran programs, or to run C programs to extract parameters and create JPEG images; (e) “*src/*” contains relevant C source codes, and the executable programs compiled from these source codes are resited in “*bin*\” sub-directory.

## Python Modules and classes

Python is the main programming language used in this system. Six Python modules were implemented and other two modules “*julian.py*” and “*isotime.py*” are borrowed from other system (Table 3).

Table 3. A list of Python modules and classes implemented and included

|  |  |  |
| --- | --- | --- |
| **package** | **Module/Class** | **Notes** |
|  | runNbarProcessor | The main driver file which runs the NBAR processing and packaging processes. Option “-l” specifies log file name. |
| python | \_\_init\_\_.py | An empty file which is required to make Python treat the directory “python” as containing packages. It can execute initialization code for the package or set the \_\_all\_\_ variable. |
| NbarProcessor.py | The class which controls the NBAR processing work flow in running the 9 Fortran scripts. |
| NbarPackage.py | A class which packages NBAR products generated into NEO standard datasets packages. |
| NbarParser.py | A class used to parse metadata/parameters from various input files. |
| NbarData.py | A data structure that contains all parameter parsed from input files. |
| NbarConfig.py | defines a set of input/output directories |
| NbarUtils.py | implements a set of functions for I/O, string and date format, parsing arguments, etc |
| isotime.py | Two ancillary python files that are “borrowed from other system and contain functions for calculating Landsat 5/6 orbit number. |
| julian.py |

## C Programs

Six C programs (Table 4) were developed for extracting inputs, generating JPEG images and converting flat binary files into GeoTIFF. The main reason for implementing C programs is simply because that the JPEG and GeoTIFF libraries were implemented in C language as well.

A Makefile in “src/” directory is provided and used to compile relevant C source code. The executable C programs will be output to the directory “*bin/*”.

Table 4. A list of C programs and corresponding course code

|  |  |  |  |
| --- | --- | --- | --- |
| **C Programs**  **in bin/** | **Source Code in src/** | | **Notes** |
| **C Files** | **H Files** |
| extractDEMhight | extractDemHeight.c | extractDemHeight.h | Extract DEM height for the scene centre |
| extractOzone | extractOzone.c | extractOzone.h | extract ozone value for that month |
| extractVapour | extractVapour.c | extractVapour.h | Extract water vapour for that day of year and hour from a GeoTIFF file |
| bin2jpg | bin2jpg.c | bin2jpg.h | Create a JPEG image from three 16 bits flat binary files as RGB components. |
| *JpegUtil.c* | *JpegUtil.h* |
| hdfExtractor | HdfExtractor.c |  | Extract average BRDF values from a subset of HDF files. |

## Shell Scripts

Shell scripts (Table 5) are used to run a series of executable Fortran programs and Linux/Unix commands.

Table 5. A list of Shell scripts

|  |  |
| --- | --- |
| **Shell Script** | **Notes** |
| *runangle\_ortho\_bin* | invoke the FORTRAN program bin/angle\_ortho\_bin |
| *runmodtraninput\_netcdf* | invoke 4 FORTRAN programs: bin/read\_modtrancor\_ortho, bin/input\_modtran\_ortho, bin/refort\_tp5\_ga and bin/refort\_tp5\_ga\_trans |
| *runmodtran\_netcdf\_lanwei* | invoke the commercial program bin/Mod4v3r1\_F90.exe |
| *runflux* | invoke the FORTRAN program bin/read\_flx\_ga |
| *runtrans* | invoke the FORTRAN program bin/read\_flx\_ga\_trans |
| *runcoeffiecient* | invoke the FORTRAN program bin/coefficient |
| *runmodtranoutput* | invoke the FORTRAN program bin/read\_modtran |
| *runbinear\_l5\_ortho* | invoke the FORTRAN program bin/binear\_ortho |
| *runbrdf\_allband\_ortho\_bin* | invoke the FORTRAN program bin/brdf\_sim\_bin |
| *md5checksum.bsh* | Run the Linux command “*md5sum*” to generate checksum for all input files under given directory. |

# Maintenance Notes

For purposes of recording, maintenance, and future extension of the system, a few questions and possible answers are listed below:

## Installation of the program

Installation of the Landsat NBAR processing system includes the following steps:

* 1. copy the program home directory “NBAR\_PRJ” to the new machine;
  2. copy the built-in input directory “NBAR\_INPUTS” to the new machine;
  3. modify “python/NbarConfig.py”

open the “python/NbarConfig.py” in a file edit application

set NBAR\_INPUTS = the full path where the directory “NBAR\_INPUTS” is

* 1. compile all Fortran codes

> cd ./fortran

> make

* 1. install Modtran4 program (see /tmp/Mod4v3r1/READMEinstall.txt)
     + Untar the file: Mod4v2r1.tar in the ‘/tmp/’ directory

cd ./tmp

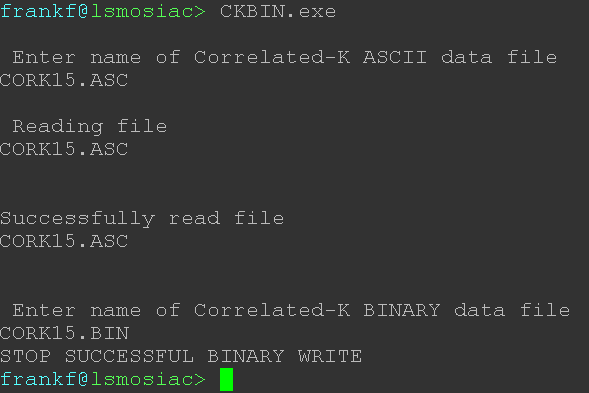
tar xvf Mod4V3R1\_unix.tar

* + - Create Correlated-k binary data files in the DATA/ subdirectory

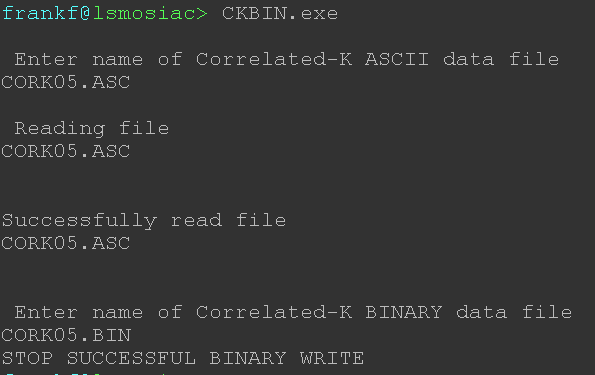
cd ./ Mod4v1r1/DATA/

gfortran -Wall -g -fbounds-check CKBIN.f -o CKBIN.exe

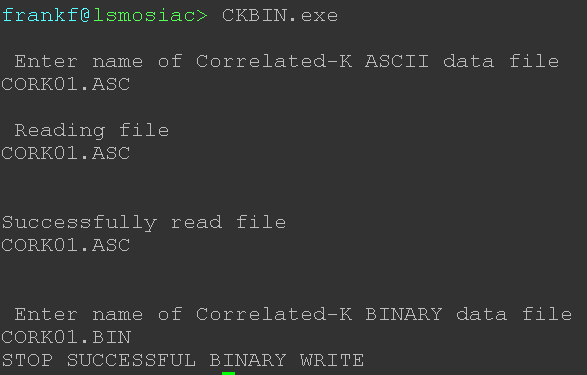
* + - Run CKBIN.exe to generate Correlated-k binary files from ASCII files
      * Run CKBIN.exe and enter input file CORK15.ASC, and then will generate the binary file: CORK15.BIN.



* + - * Run CKBIN.exe and enter input file CORK05.ASC, and then will generate the binary file: CORK05.BIN.



* + - * Run CKBIN.exe and enter input file CORK01.ASC, and then will generate the binary file: CORK01.BIN.



* + - Compile MOLBMP.f to build MOLBMP.exe

gfortran -Wall -g -fdefault-double-8 -fdefault-integer-8 -fdefault-real-8 -fbounds-check MOLBMP.f -o MOLBMP.exe

* + - Run MOLBMP.exe to create band model parameter files

Run MOLBMP.exe and enter “1” to select an ASCII-to-binary conversion (Option 1), and then enter “2”, and “2” to create the direct-access, binary (unformatted) band model file: B2001\_15.BIN.

|  |
| --- |
| frankf@lsmosiac> MOLBMP.exe  PROGRAM MOLBMP: CONVERTS BETWEEN THE SEQUENTIAL-ACCESS ASCII  (FORMATTED) AND THE DIRECT-ACCESS BINARY (UNFORMATTED)  MODTRAN4 MOLECULAR BAND MODEL PARAMETER FILES  (THE FILE FROM WHICH DATA IS READ IS NOT DELETED).  ENTER 1 TO CREATE BINARY (UNFORMATTED) FILE FROM ASCII (FORMATTED) FILE  2 TO CREATE ASCII (FORMATTED) FILE FROM BINARY (UNFORMATTED) FILE  1  ENTER ASCII BAND MODEL FILE NAME (MAX 150 CHARACTERS)  [ENTER 0 FOR NAME = "B2001\_01.ASC"]  [ENTER 1 FOR NAME = "B2001\_05.ASC"]  [ENTER 2 FOR NAME = "B2001\_15.ASC"]  2  ENTER BINARY BAND MODEL FILE NAME (MAX 150 CHARACTERS)  [ENTER 0 FOR NAME = "B2001\_01.BIN"]  [ENTER 1 FOR NAME = "B2001\_05.BIN"]  [ENTER 2 FOR NAME = "B2001\_15.BIN"]  2  AN ERROR OCCURRED WHEN A RECORD LENGTH OF 15 WAS USED.  A RECORD LENGTH OF 30 WILL NOW BE TRIED.  AN ERROR OCCURRED WHEN A RECORD LENGTH OF 30 WAS USED.  A RECORD LENGTH OF 60 WILL NOW BE TRIED.  NTEMP,(TBAND(IT),IT=1,NTEMP)  6 180. 205. 230. 255. 280. 305.  IBNDWD= 15 IF1ST= 0 IFLAST= 22680  LSTREC= 5862 NDIVID= 1 DEDGE= 1.200  THE FIRST TWO BAND MODEL RECORDS ARE:  0 1 3.141E-04 2.601E-04 2.171E-04 1.828E-04 1.554E-04 1.331E-04  1009 9.885E-02 1.019E-01 1.052E-01 1.089E-01 1.130E-01 1.177E-01  0 3 1.863E-03 1.462E-03 1.180E-03 9.743E-04 8.196E-04 7.004E-04  826 4.098E+00 4.620E+00 5.200E+00 5.828E+00 6.489E+00 7.170E+00  THE LAST RECORD, 5862, IS:  22680 13 6.509E-11 4.907E-11 3.821E-11 3.053E-11 2.491E-11 2.068E-11  0 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00  TESTING BINARY TAPE  BAND MODEL PARAMETER FILE CONTAINS 5862 RECORDS (LAST FREQUENCY IS 22680 CM-1)  SUCCESSFULLY CREATED THE DIRECT-ACCESS, BINARY (UNFORMATTED) BAND MODEL FILE:  B2001\_15.BIN  STOP Success! |

Similarly, enter “1”, “1” and “1” to create B2001\_05.BIN and “1”, “0” and “0” to create B2001\_01.BIN.

* + - Modify make file “Mod4v3r1/Make\_F90”

in Line 5-6 of Make\_F90:

f90 = f90

opt = -O

changed to:

f90 = gfortran

opt = -Wall -g -fbounds-check

* + - Compile and Build Mod4v2r1\_F90.exe

Run make command to build the FORTRAN90 version of the MODTRAN4 Version2 Revision1 executable file, Mod4v2r1\_F90.exe. The FORTRAN90 object files are placed in the “Mod4v3r1/obj90/” directory.

cd ../

make -f Make\_F90

* + - Copy relevant files to the program

cp -af Mod4v3r1\_F90.exe ../../bin/

cp -ar DATA ../../

cp flts/\*.flt ../DATA/

## Usage and execution of the program

usage: runNbarProcessor [-l logfile] <input\_dir> [output\_dir] <sensor>

[-l logfile]: optional flag to specify the log file name,

default is “log/work-order.log”

“-l stdout” or “-l std” for stdout

<input\_dir>: input directory of a dataset package specified

by a work-order number

[output\_dir]: optional output directory for storing NBAR

products generated, default output directory

is defined by "NBAR\_OUTPUTS" in NbarConfig.py

<sensor>: sensor of the dataset

## Input and output directories

Basically, there are two types of input directories: built-in input directory and work order input directory. The built-in input directory is defined as the constant variable “NBAR\_INPUTS” in the NbarConfig.py (Table 8). It contains 8 sub-directories such as “aerosol/aeronet/”, “brdfDB/MODISBRDF/”, “DEM/”, “earthsun\_distanceLUT/”, “ozone\_LUT/”, “satellite\_filter/”, “solar\_irradianceLUT/”, “water\_vapour/data/”, each of which contains important input files needed to feed the system. The work order input directory is to be specified in the command-line. It is a full path pointing to an input dataset package named by a work order number.

The output directory is used to store all output NBAR data images and ancillary files. The users may need to specify the output directory in the command-line. Otherwise the system will use the default output directory which is defined by the variable “NBAR\_OUTPUTS” in “NBbarConfig.py” (see Table 8).

Table 8. A list of global settings defined in NbarConfig.py

|  |
| --- |
| class NbarConfig:    # main built-in input directory containing predefined inputs for NBAR processing  NBAR\_INPUTS = ' /image10/stripe/NBAR\_INPUTS'    # default output directory for storing outputs  NBAR\_OUTPUTS = ' /image10/stripe/NBAR\_OUTPUTS'    # A set of sub-directories which contain  DIR\_Aerosol = NBAR\_INPUTS + "/aerosol/aeronet"  DIR\_BRDF = NBAR\_INPUTS + "/brdfDB/MODISBRDF"  DIR\_DEM= os.path.join(NBAR\_INPUTS, "DEM")  DIR\_EarthSun\_LUT = NBAR\_INPUTS + "/earthsun\_distanceLUT"  DIR\_Ozone\_LUT = NBAR\_INPUTS + "/ozone\_LUT"  DIR\_SatFilter = NBAR\_INPUTS + "/satellite\_filter"  DIR\_SolarIrradianceLUT = NBAR\_INPUTS + "/solar\_irradianceLUT"  DIR\_WaterVapour = NBAR\_INPUTS + "/water\_vapour/data" |

## Log file

The Landsat NBAR processing system automatically maintains log files to record all run-time information, including time of execution, work orders processed, in/output directories, warning and error messages if any, as well as CPU time duration. These log files are stored in the “log/” directory and named by their input work order number, eg. log/07889\_01.log. Alternatively, users can specify the name of a log file in the command-line flag. For instance, if enter a command-line like “runNbarProcessor -l myworkorder.log ... ...”, then all runtime info will be output to the log file “log/ myworkorder.log”.

## Lock File

As mentioned before that ModTran4 is commercial scientific software that GA currently has only one licence, it is essential to make sure that at any time only one instance of the Landsat NBAR processing system is running. To achieve this, a lockup mechanism from the Python fcntl module is implemented. When executing “runNbarProcessor”, the program will try to lock a lock file named “landsat\_nbar.lck” in the directory “log/” for writing. If the attempt fails, which means that the program is being run by previous process, the program will then quit, otherwise go ahead to process NBAR datasets.

The lockfile mechanism was implemented between Line 35-39 in runNbarProcessor:

|  |
| --- |
| nbarLog = os.path.join(progDir, "log")  lockfile = os.path.join(nbarLog, 'landsat\_nbar.lck')  if isLocked(lockfile):  fpLog.write(' Can\'t run the NBAR Processor as it is locked by other process. Exit!\n')  sys.exit(8) |

## Post-implementation bugs and problems

...

# Appendix

## Temporary/intermediate files extracted from inputs

brdf\_modis\_band1/2/3/4/5/7.txt

|  |
| --- |
| 0.100051 0.034705 0.003607  -7.38074 1.180710 1997.0 0.98717 |

coordinator\_ppp\_rrr\_20090212\_ortho.txt

|  |
| --- |
| 8800 8800  1 1541  1 5544  1 8800  4400 347  4400 4400  4400 8444  8800 1  8800 3237  8800 7332 |

centerline\_ppp\_rrr\_20090212\_ortho.txt

|  |
| --- |
| 7.690000  8800 8800  1 5544 -29.1836231715703 140.469471769687  2 5544 -29.1838731715557 140.469407294477  3 5543 -29.1841231715411 140.469342819031  4 5543 -29.1843731715525 140.469278343345  5 5543 -29.1846231715119 140.469213867436  6 5543 -29.1848731715754 140.469149391267  7 5542 -29.1851231715347 140.469084914889  8 5542 -29.1853731714940 140.469020438276  9 5542 -29.1856231714534 140.468955961429  10 5542 -29.1858731716211 140.468891484293  11 5541 -29.1861231715805 140.468827006977  … … … … … … … …  8799 3237 -31.3831230425705 139.892715170974  8800 3237 -31.3833731982351 139.892648444226 |

header\_angle\_ppp\_rrr\_20090212.txt

|  |
| --- |
| 43 0.364313805556  8800 8800  -29.183625 139.083625  -30.2835 140.183499  7083160.0 98.2 0.001059  7.690000 |

input\_modtraninput\_ppp\_rrr\_20090212.txt

|  |
| --- |
| -29.183625 139.083625  0.257  1.080000  DATA/landsat7\_vsir.flt  97.050440  0.035000  Annotation, 2009-02-12  705  43  0.364314 |

satellite\_filter\_input\_ppp\_rrr\_20090212.txt

|  |
| --- |
| 6  /net/ausimages4/data1/projects/nbar\_landsat/NBAR\_INPUTS/satellite\_filter/landsat  7\_vsir.flt |

startend\_ppp\_rrr\_20090212\_ortho.txt

|  |
| --- |
| 1 1670 5214 8757 7088  2 1676 5220 8764 7089  3 1683 5227 8771 7089  4 1689 5234 8778 7090  5 1696 5241 8785 7090  6 1702 5247 8792 7091  7 1709 5254 8799 7091  8 1619 5210 8800 7182  9 1619 5210 8800 7182  10 1619 5210 8800 7182  11 1618 5209 8800 7183  … … … … … … … …  8799 59 3692 7325 7267  8800 66 3644 7221 7156 |

## Final output files of a NBAR dataset packaged

Below are outputs of report.txt, label.txt, readme.txt and licence.txt from a packaged NBAR dataset. Text in blue colour shows the values of parameters which may be different from datasets.

B-1. report.txt

|  |
| --- |
| GA NBAR PROCESSING REPORT  -------------------------  Dataset ID: LS5\_TM\_OTH\_P54\_NBAR\_090\_084\_20090117  Satellite: Landsat-5 Sensor: TM  Camera Number: N/A Sensor Mode: N/A  Ground Station: ALICE  Map Projection: EQUIRECT\_0\_0 Zone: N/A  Earth Ellipsoid: GDA94  Path/Strip No.: 090 Row No: 084  Image Lines: 9200 Image Pixels: 9200  Image Orientation: 0.00 deg from N Pixel Size: 0.00025 deg  Output Bands: 1 2 3 4 5 7  Scene Centre Lat: -34.600000 deg Scene Centre Long: 149.816749 deg  Scene Centre Date: 2009 01 17 Scene Centre Time: 23:34:40.5985  Product Format: N/A Interleaving: BSQ  Completion Date: 2010 02 24 Completion Time: 20:30:09  Termination Status: Successful Completion  PRODUCT FORMATTING  ------------------  Product Scene Center Location (lat/long) : -34.600000 149.816749  Product Scene Center Date/Time (yyyy mm dd): 2009 1 17 23:34:40.5985  Product Extent:  Lat: -33.450125---------------------------------Lat: -33.450125  Long: 148.666875 Long: 150.966625  | |  | |  | |  | |  | |  | |  Lat: -35.749875 Lat: -35.749875  Long: 148.666875---------------------------------Long: 150.966625 |

B-2. label.txt

|  |
| --- |
| Copyright (c) Commonwealth  of Australia 2010  Landsat-5 TM Acquired: 17-Jan-2009  Centre: S34:36:00/E149:49:00  Ortho Corrected Nadir BRDF-Adjusted  Reflectance (NBAR) Image |

B-3. readme.txt

|  |
| --- |
| README.TXT  This document explains the directories and files for GA  Landsat Nadir BRDF-Adjusted Reflectance (NBAR) products.  The image has been scaled to 2 byte integer with a scale facter  of 10000 (reflectance = DN / 10000).  root directory  --------------  nnn\_nnn\_fff\_ggg\_hhh\_ppp\_rrr\_yyyymmdd.jpg - Browse image  readme.txt - this document  label.txt - the label information associated with this product  metadata.xml - scene metadata in XML format  md5sum.txt - MD5 checksum is checking cryptographic message  digests(or check values) of files. You can use this  file to check the integrity of your product  /scene01 Sub-directory containing all the files pertaining  -------- to the one scene/product.  nnn\_nnn\_fff\_ggg\_hhh\_ppp\_rrr\_yyyymmdd.env - 6 image bands containing  image pixels only  nnn\_nnn\_fff\_ggg\_hhh\_ppp\_rrr\_yyyymmdd.hdr - ENVI header file  report.txt - GA product report  Where,  nnn\_nnn: satellite and sensor ID  fff: product code  ggg: process code  hhh: process source/group  ppp: path  rrr: row  yyyymmdd: acquisition date  Copies of the GA NBAR Theoretical Basis Document is available through GA website. |

B-5. md5sum.txt

|  |
| --- |
| 1093e5b0af971abd5efde19d344a28b9 ./readme.txt  3f15f89bcda298ed1b82ac7fc795171e ./label.txt  f42e0cd5fc9054e0545f9b9f62de43ed ./metadata.xml  3b3c0817b1253725f8b00b949a50bc38 ./LS5\_TM\_NBAR\_P54\_GAV1\_093\_085\_20080916.jpg  8d27cb2d671b3774e488758694b7061a ./scene01/report.txt  fc901c9a3cea70d5dc14617ac70ddf63 ./scene01/LS5\_TM\_NBAR\_P54\_GAV1\_093\_085\_20080916.hdr  049a3dbdb99fa2192d68b9423958e9f0 ./scene01/LS5\_TM\_NBAR\_P54\_GAV1\_093\_085\_20080916.env |