Landsat
Collection 2 (C2) Level 3 (L3)
Burned Area (BA)
Algorithm Description Document (ADD)

October 2021

Document Owner: 
CHRISTOPHER ENGBRETSON

Digitally signed by
CHRISTOPHER ENGBRETSON
Date: 2021.10.20 14:22:14 -05'00'

Chris Engebretson
Landsat DPAS CCB Chair
U.S. Geological Survey

EROS
Sioux Falls, South Dakota
Executive Summary

This Algorithm Description Document (ADD) defines the algorithm used for the generation of Landsat Collection 2 (C2) Level 3 (L3) Burned Area (BA) Science Product created at the U.S. Geological Survey (USGS) Earth Resource Observation and Science (EROS) Center.

This document is under Data Processing and Archive System (DPAS) Configuration Control Board (CCB) control. Please submit changes to this document, as well as supportive material justifying the proposed changes, via Change Request (CR) to the Process and Change Management Tool.
## Document History

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Version</th>
<th>Publication Date</th>
<th>Change Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSDS-2083</td>
<td>Version 1.0</td>
<td>October 2021</td>
<td>20794</td>
</tr>
</tbody>
</table>
Contents

Executive Summary .................................................................................................................. iii
Document History ................................................................................................................ iv
Contents ................................................................................................................................. v
List of Figures ....................................................................................................................... v
List of Tables .......................................................................................................................... v
Section 1  Introduction ......................................................................................................... 1
  1.1  Background .................................................................................................................. 1
  1.2  Purpose and Scope ...................................................................................................... 1
  1.3  Document Organization ............................................................................................. 1
Section 2  Burned Area Algorithm Details ................................................................. 2
  2.1  Burned Area Details .................................................................................................. 2
  2.2  Algorithm Inputs ....................................................................................................... 2
  2.3  Burned Area Outputs ............................................................................................... 3
  2.4  Burned Area Flowchart ........................................................................................... 3
Section 3  Burned Area Algorithm Procedure ........................................................... 4
  3.1  Preprocessing (preprocessing.py) .............................................................................. 4
  3.1.1  Forward Processing ............................................................................................. 5
  3.2  Converting Pixel QA to Fmask bit arrangement (qa2Fmask.py) ............................... 5
  3.3  BA Processing ......................................................................................................... 6
  3.4  Subsetting the ARD tile stack for the current year (subset_stack.py) ..................... 7
  3.5  Copying the files associated to subset stack (subset_stack_copy_files.py)............. 8
  3.6  Generating the burn probabilities (sample_and_predict_open_by_block.py) ... 8
  3.7  Generating binary burn classification (threshold_stack_by_scene.py) .......... 10
  3.8  Writing XML metadata (scene_EarthExplorer_metadata.py) .............................. 11
  3.9  Generating PNG browse files (scene_visuals.py) ................................................. 11
Section 4  Citation Information ....................................................................................... 13
Appendix A  Acronyms ......................................................................................................... 14
References .............................................................................................................................. 16

List of Figures

Figure 2-1. Burned Area Algorithm Flow Diagram ............................................................. 3

List of Tables

Table 2-1. Burned Area Algorithm Inputs ........................................................................... 3
Table 2-2. Burned Area Algorithm Outputs ......................................................................... 3
Section 1  Introduction

1.1  Background
The Burned Area (BA) algorithm identifies burned areas in historical time series of Landsat image archive to produce the Landsat Collection 2 (C2) Level 3 (L3) BA Science Product. The algorithm is based on the Landsat Burned Area Essential Climate Variable algorithm developed by Todd Hawbaker (USGS Geosciences and Environmental Change Science Center) with modifications to handle Landsat Analysis Ready Data (ARD) and incorporate Landsat 8 data, and is refined to use spectral indices developed specifically for mapping burned areas. The BA algorithm requires the Pixel Quality Assessment (QA) band, Top of Atmosphere (TOA) Brightness Temperature (BT), and Surface Reflectance (SR) as input, in addition to several static inputs described in Section 2.2. The Collection 2 Level 3 BA products are generated for Landsat 4-5 Thematic Mapper (TM), Landsat 7 Enhanced Thematic Mapper Plus (ETM+) and Landsat 8 Operational Land Imager (OLI) / Thermal Infrared Sensor (TIRS) acquisitions.

For additional information about the characteristics of the BA product, please see the LSDS-2041 Landsat Collection 2 (C2) Level 3 (L3) Burned Area (BA) Data Format Control Book (DFCB).

1.2  Purpose and Scope
The primary purpose of this document is to provide technical details and information on the Landsat Collection 2 Level 3 BA algorithm.

1.3  Document Organization
This document contains the following sections:

- Section 1 introduces BA
- Section 2 provides technical details on the inputs and outputs of BA algorithm
- Section 3 provides details on the BA algorithm procedure
- Section 4 contains citation information
- Appendix A contains a list of acronyms
- The References section contains a list of reference documents and supporting webpages
Section 2  Burned Area Algorithm Details

2.1  Burned Area Details

The BA algorithm involves multiple steps: pre-processing of the input data stacks (and preparing for the modeling), modeling burned area probabilities for each acquisition, and generation of the annual composites (future addition TBD).

From the Landsat Collection 2 (C2) Analysis Ready Data (ARD) tiles, the algorithm makes use of predictors derived from individual ARD tiles, lagged reference conditions, and change metrics between the tile and reference predictors.

The algorithm output products are the BA tiles, which consist of burn probabilities, a binary burn classification, a quick-look browse file, and a metadata file.

The Landsat Burned Area algorithm is implemented as a series of python scripts that are executed in sequence with a bash shell script (pBAMA_v2_ARD.sh). The python scripts and shell script are run on the cloud environment as part of Collection 2. The BA algorithm involves the following steps, each performed by an individual python script, in the order that they are executed within pBAMA_v2_ARD.sh:

1. Preprocessing (preprocessing.py)
2. Converting Landsat Level 2 Pixel QA to a quality band with Fmask bit arrangement (qa2Fmask.py)
3. BA Processing
   a. Subsetting the ARD tile stack for the current year (subset_stack.py)
   b. Copying the files associated to subset stack (subset_stack_copy_files.py)
   c. Generating the burn probabilities (sample_and_predict_open_by_block.py)
   d. Generating binary burn classification (threshold_stack_by_scene.py)
   e. Writing XML metadata (scene_EarthExplorer_metadata.py)
   f. Generating PNG browse files (scene_visuals.py)

2.2  Algorithm Inputs

Table 2-1 lists the Inputs of the BA algorithm.

<table>
<thead>
<tr>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landsat Collection 2 ARD tile stack of Surface Reflectance (SR) and Top of</td>
<td>Used to create the spectral indices and tasseled cap variables for</td>
</tr>
<tr>
<td>Atmosphere (TOA) Brightness Temperature (BT)</td>
<td>generating the burn probabilities</td>
</tr>
<tr>
<td>Level 2 pixel QA Mask (QA_PIXEL)</td>
<td>Used to create a quality band with Fmask bit arrangement</td>
</tr>
<tr>
<td>Radiometric Saturation (QA_RADSAT)</td>
<td>Used to create the quality band with Fmask bit arrangement</td>
</tr>
<tr>
<td>Level I ecoregion raster file</td>
<td>Used to create variables that are input to the model</td>
</tr>
</tbody>
</table>
Table 2-1. Burned Area Algorithm Inputs

<table>
<thead>
<tr>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shapefile for Landsat ARD</td>
<td>Used to crop the ecoregion data to the current tile extents</td>
</tr>
</tbody>
</table>

2.3 Burned Area Outputs

Table 2-2 lists the outputs of the BA algorithm.

<table>
<thead>
<tr>
<th>Description</th>
<th>Units</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burn Probability</td>
<td>Percent/Flag</td>
<td>8-bit Unsigned Integer</td>
</tr>
<tr>
<td>Burn Classification</td>
<td>Flag</td>
<td>8-bit Unsigned Integer</td>
</tr>
<tr>
<td>Browse file</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 2-2. Burned Area Algorithm Outputs

2.4 Burned Area Flowchart

Figure 2-1 shows the overall flow diagram of the Burned Area algorithm.

Figure 2-1. Burned Area Algorithm Flow Diagram
Section 3  Burned Area Algorithm Procedure

3.1  Preprocessing (preprocessing.py)

Inputs:
1. Landsat ARD tile stack of SR and TOA BT products for the desired date ranges, including data for the previous 3 years (for lagged reference conditions)
2. Temporary directory for processing the data
3. Output directory for writing the preprocessed results
4. Name of .csv stack file for storing information about the ARD tile stack
5. ARD region to be processed (CU)
6. Horizontal tile number to be processed
7. Vertical tile number to be processed
8. Reference bounding box for the current region (static file)
9. Shapefile for the Landsat ARD Tile polygons in the current region (static file – used to crop the ecoregion data to the current tile extents)
10. Level 3 ecoregion raster file in the current region (static file-used for input to the models)

Outputs:
1. Multi-band Georeferenced Tagged Image File Format (GeoTIFF) file (8 bands) with TM/ETM+ bands 1-7 and a modified quality band with Fmask bit arrangement for each acquisition (stored in the {output_dir}/lndsr directory)
2. Original XML metadata file for each acquisition (stored in the {output_dir}/lndsr_metadata directory)
3. ARD tile polygon for the current ARD tile (stored in the {output_dir}/vectors directory)
4. Ecoregion shapefile cropped to the current ARD tile (stored in the {output_dir}/rasters directory)
5. stack.csv metadata file which contains a list of all the XML files, associated metadata, and an output field to designate whether or not they will get processed (stored in the output directory)
6. bb.csv bounding box file which contains the west, north, east, and south bounding box coordinates for this tile (stored in the output directory)

Description:
The preprocessing script is automated to set up the subdirectories for the various outputs which will be generated for the particular geographic region as well as the specific ARD Tile. Projection information for the tile is obtained and set for output products. Next, the bounding box coordinates for this tile are determined and written to the bb.csv file – these are used to ‘snap’ the cropped ecoregion raster to the Landsat data. All of the output raster files (ecoregions) and vectors (ARD tile polygons) are cropped and written to the output raster or vector directory, respectively.

The list of .xml files within the input directory is then generated. The TOA Brightness Temperature file (band 6 for TM / ETM+ and band 10 for TIRS) is extracted from the BT product. Data is stored into a temporary directory, and the pixel QA, Surface Reflectance bands, and XML metadata files are extracted from each SR product. The
OLI band names are renamed to match the TM/ETM+ band names based on their spectral range (e.g., the near-infrared OLI band 5 is renamed to band 4). The pixel QA band is converted to a modified quality band with Fmask bit arrangement (see Section 3.2 for more information). Then, all bands are stacked into a single multiband GeoTIFF file, in order of SR_band1, SR_band2, SR_band3, SR_band4, SR_band5, BT_band6, SR_band7, and a QA band with Fmask bit arrangement. The multiband GeoTIFF file is written with compression to the output directory. The temporary directory and contents are removed.

The last step of preprocessing is to loop through all the XML metadata files, extract the metadata, and write the metadata to a Comma Separated Values (CSV) file, which contains key information about each acquisition in the stack, such as satellite, sensor, acquisition date, percent cloud cover, percent snow and ice, percent fill, and the geometric Root Mean Square Error (RMSE). If the geometric RMSE is greater than 10 meters or the percent cloud cover is greater than 80%, then the acquisition is not processed. If the date of the acquisition is prior to 1984, then the acquisition is written to the CSV file but flagged to not be processed for burned area probabilities. After processing has completed, if the CSV file exists in the output directory, then it indicates that the preprocessing completed successfully.

This approach, in general, describes the "stack processing" of historical temporal stacks of data (e.g., 1984-2021).

3.1.1 Forward Processing

Once the historical archive has been processed, new data will also need to be processed as part of "forward processing". The forward processing code will only process new acquisitions. The code checks the preprocessing output directory. If an acquisition doesn’t exist in that directory, then the acquisition is processed as described above and written to a multiband GeoTIFF file. The associated XML metadata file is written to the metadata directory. The stack.csv file will contain metadata for each of the acquisitions in the ARD tile stack; however, the only acquisitions flagged for processing in the CSV file will be the new acquisitions. The forward processing code also expects that the ARD tile polygon and the ecoregion shapefile are available as part of the initial stack processing.

NOTE: All the data products written to the output directory are used in full as part of the BA processing. Forward processing and full stack processing will utilize data in this directory. Do NOT rerun pre-processing unless new files have been added to the input tile stack, otherwise the output stack.csv file will specify all products with a ‘0’ in the output column for not processing.

3.2 Converting Pixel QA to Fmask bit arrangement (qa2Fmask.py)

Inputs:
1. Pixel QA band
2. Radiometric Saturation band
**Outputs:**
1. Quality band with Fmask bit arrangement (unsigned 8-bit integer)

**Description:**
The qa2Fmask routine reads the pixel QA band (QA_PIXEL), then loops through each pixel in the QA band to convert them to values that were previously used in the Fmask band.

```python
If pixel_qa is fill
    Fmask = nodata (255)
Else if pixel_qa is clear and (pixel_qa cloud confidence <= 1 or pixel_qa cirrus confidence <= 1)
    Fmask = 0
Else if pixel_qa is cloud and (pixel_qa cloud confidence >= 1 or pixel_qa cirrus confidence >= 1)
    Fmask = 3
Else if pixel_qa is cloud shadow
    Fmask = 4
Else if pixel_qa is snow
    Fmask = 2
Else if pixel_qa is water
    Fmask = 1
Else if radiometric saturation is terrain occluded
    Fmask = nodata (255)
Else
    Fmask = 0
```

The data are written out as unsigned 8-bit integers to a GeoTIFF file using DEFLATE compression. The geolocation information from the input pixel QA band is used for this band and the fill value is written as 255. The file extension is _CFMASK.tif.

### 3.3 BA Processing

**Inputs:**
1. Multi-band GeoTIFF file (8 bands) with TM/ETM+ bands 1-7 and a quality band with Fmask bit arrangement for each acquisition (stored in the {input_dir}/lndsr directory)
2. Original XML metadata file for each acquisition (stored in the {input_dir}/lndsr metadata directory)
3. ARD tile polygon for the current ARD tile (stored in the {input_dir}/vectors directory)
4. Ecoregion shapefile cropped to the current ARD tile (stored in the {input_dir}/rasters directory)
5. stack.csv metadata file which contains a list of all the XML files, associated metadata, and an output field to designate whether they will get processed (stored in the input directory)

**Outputs:**
1. Burn probability image for each acquisition {unsigned 8-bit integer}
2. Binary burn classification image for each acquisition {unsigned 8-bit integer}
3. ESPA XML metadata file for each acquisition
4. PNG browse files for each acquisition (Landsat 7-4-2, burn probability, burn classification)

Description:
Processing of the burned area probabilities and classifications occurs one year at a time. The processing script loops through each year, from the specified start year to end year, and determines the input and output directories and files, based on datasets from the preprocessing. For year in the date range, using an annual time increment, the following steps are implemented:
1. Subset the stack.csv file to the stack needed for the current year
2. Copy the files associated to this subset stack for the ARD tile to the temporary processing directory
3. Generate the burn probabilities
4. Generate binary burn classifications from the burn probabilities
5. Write XML metadata file for the burn probabilities and burn classification images
6. Generate PNG browse files for each acquisition showing three adjacent browse images of the raw Landsat Shortwave Infrared (SWIR2) / Near-infrared (NIR) / Green color composite, burn probability, and the burn classification
7. Copy the results to the destination folder

3.4 Subsetting the ARD tile stack for the current year (subset_stack.py)

Inputs:
1. stack.csv metadata file which contains a list of all the XML files, using a ‘1’ in the output column to specify if the file should be processed
2. Start date for the metadata subset
3. End date for the metadata subset
4. Sensor (optional, all sensors are included in the subset if not specified)

Outputs:
1. Subset stack.csv which contains only the metadata needed to process burn probabilities from start date to end date (with lags applied), as well as the specified sensor (default is to process all the sensors). Acquisitions will be specified for burned area processing based on the output field value.

Description:
This script subsets the full temporal stack.csv file based on the specified start date, end date, and sensor (optional). The input stack.csv file is read. The start year and end year are pulled from the start and end dates, respectively. A time lag of 3 years is subtracted from the start year to provide enough data for the lagged reference conditions and change metrics.

If the sensor is not specified, then the sensor is not used in the subsetting and all files are processed. A mask is set up based on the acquisition date, time lag, and sensor in the stack.csv file. Any records that meet the start/end date are flagged with an output
field value of 1, indicating these acquisitions are processed for burn probabilities. Otherwise, the output value is set to 0 and the acquisition is used only for generating the summaries and metrics. The original stack.csv is then subset using the date range, time lag, and sensor.

### 3.5 Copying the files associated to subset stack (subset_stack_copy_files.py)

**Inputs:**

1. Subset stack.csv (from subset_stack) which contains only the metadata for start date to end date, using a ‘1’ in the output directory to specify files that need to be processed
2. Directory location of the multiband GeoTIFF files, metadata files, raster, and vector files for the ARD tile stack

**Outputs:**

1. Directory to copy the subset of files to for burned area processing

**Description:**

This script copies the ecoregion raster file to the output directory. It then opens the subset stack file and reads the names of the acquisitions in the subset. Any files which exist in the destination directory which don’t apply to the subset stack are removed (the same output directory is used for processing year after year of data and therefore previous years of data will have already been copied). The missing multiband GeoTIFF file and XML metadata files for the subset stack are copied from the input Lndsr and Lndsr_metadata directories to the same subdirectories in the output directory.

### 3.6 Generating the burn probabilities (sample_and_predict_open_by_block.py)

**Inputs:**

1. Subset stack.csv (from subset_stack) which contains only the metadata for start date to end date, using a ‘1’ in the output directory to specify files that need to be processed
2. Directory location of the multiband GeoTIFF files, metadata files, raster, and vector files for the ARD tile stack
3. Shelf file with gradient-boosted regression (GBR) burn classifier object (produced from training the GBR model)
4. Colormap template for the burn probability images
5. Number of rows and columns of blocks in pixels to process per CPU (tiled processing in a multi-threaded environment)
6. Number of CPUs for processing (tiled processing in a multi-threaded environment)

**Outputs:**

1. Burn probability for each of the acquisitions to be processed for the current subset stack {Byte}
Description:
This script reads the input parameters and validates the input files are available. It reads
the Gradient Boosted Regression Model (GBRM) file, which includes the tree classifier and the predictors, which currently are NBRT1 (normalized burn ratio with thermal band), nbrt1_mean_3yr, vi6t (spectral index with bands 4 and 6), vi45 (spectral ratio with band 4 and 5), vi45_mean_3yr, nbrt1_sd_3yr, NDVI (normalized difference vegetation index), and ndvi_mean_3yr. The script then sets up the 258 stack variables split into Input, Stack, Output, and other (generated from the inputs), many of which are deleted when running the GBRM in sampling mode. These variables are available for training the model, if needed.

Ultimately the only variables which are kept for modeling (keep_predictors) are the Input (raw bands, Fmask, and ecoregion file), Stack (year, Julian day, sensor), and any variables defined in the GBRM shelf variable list as well as associated difference, mean, and standard deviation variables. This stack is then sorted. A stack of unused predictors is identified as any of the variables from the initial stack which are not included in the keep_predictors stack. These unused variables are deleted from the initial stack, as are the ‘eco_l2’ and ‘eco_l3’ variables. The output burn probability response variable (BP) is added to the final stack of variables. Now the stack is ready for the modeling process.

The GBRM modeling is run in a multi-threaded environment and the data is processed in blocks, based on the size of the image and the specified lines/samples to be processed per CPU. The modeling first opens the ecoregion datasets to determine the image size. Each of the blocks are processed independently. The stack-related variables (i.e., year, sensor) are populated for the block. The ecoregion data is then read for the block, and from that file, the eco_l1, eco_l2, and eco_l3 stack variables are populated. Each of the Landsat bands are loaded into the stack next, including the Fmask band. In this section, the Fmask values are changed from 3 to 253 for clouds; 2 to 252 for snow/ice; 4 to 254 for cloud shadows; and 1 to 251 for water. This allows the Fmask values to be preserved in the burn probability outputs. From the current variables (ecoregion, bandX, and Fmask), a processing mask is created to skip the nodata and unclear pixels. If the pixel for any of the bands is fill, then the pixel is masked from processing. Furthermore, if the Fmask is non-zero (water, snow/ice, clouds, cloud shadows), then the pixel is also masked from processing. The stack is then “shrunk” to drop images with no useful data.

With the remaining data/variables in the stack, the spectral index (e.g., NDVI) variables are computed. The tasseled cap (function type of TC) variables are then computed (i.e., TCbrightness, TCgreenness, and TCwetness) as well as Soil Adjusted Vegetation Index (SAVI), Enhanced Vegetation Index (EVI), Mid-Infrared Burn Index (MIRBI), Burned Area Index (BAI), NBRT1, Global Environmental Monitoring Index (GEMI) and various spectral ratios. For the spectral indices and the tasseled cap variables, if any of the values are less than -1 or greater than 1, then they are set to -1.0 or 1.0, respectively. The annual/lagged summaries for the bands and the spectral indices are also calculated. Next, the difference predictors (difference between one variable and another) and the relative difference predictors (relative difference between one variable
and another) are calculated. From here, the GBRM burn probabilities are computed. Then, predictor variables are rearranged to match the order of the training data, and the model is run on the predictor variables for each tile in the stack. The output probabilities are converted to percentage by multiplication of 100. The burn probabilities are then combined with the Fmask values (251-255) and written to the output file. Finally, all the input/output files are closed and memory is cleaned up.

The last step of the processing, after all the blocks have been completed, is to copy the burn probability color template for the current burn probability acquisition file.

3.7 Generating binary burn classification (threshold_stack_by_scene.py)

Inputs:
1. Subset stack.csv (from subset_stack) which contains only the metadata for start date to end date, using a ‘1’ in the output directory to specify files that need to be processed
2. Input directory for the burn probabilities for this ARD tile
3. Output directory for the burn classifications for this ARD tile
4. Colormap file for the burn classification images

Outputs:
1. Binary burn classification based on the thresholding of burn probabilities, burn thresholds, region seeds, and flood filling of the burn areas {unsigned 8-bit integer}

Description:
This script reads the input stack CSV file and keeps the files that are flagged with a ‘1’ in the output column. For each file in the filtered stack, the burn probability (BP) and output burn classification (BC) filenames are determined. The flood-fill algorithm is then used for identifying the burn scars.

The algorithm first masks any burn probability values greater than 100% to be 0 – these are clouds, shadows, water or snow/ice pixels. The bp_regions and bp_seeds arrays are defined as the same shape as the burn probability image and initialized to zero. For each pixel, if the burn probability is greater than or equal to the seed probability threshold (96%), then flag it as a seed pixel in bp_seeds. Next, small seeds are removed using a seed size threshold of 22 pixels (which is approximately 5 acres). This helps clean up false positives but may also remove small fires.

If there are any seed pixels in the bp_seeds image, then region properties are obtained from the first pixel in each region. Regions are defined by pixels with a connectivity of 1. Now loop through each region, pulling the area (number of pixels) for the region and the line/sample coordinates of the region. If the size of the region is greater than or equal to the seed size threshold (22 pixels ~ 5 acres), then the region is filled in with surrounding pixels of lower probability. Thus, if any of the surrounding pixels are between 71% and 100% burn probability, then they are added as burn pixels in the region.
The binary burn classification is generated from pixels that were flagged in the burn regions. The Fmask information is added for pixels that were not classified as burned. The new burn classification image is written to the burn classification and the burn classification color file is copied to the output directory.

3.8 Writing XML metadata (scene_EarthExplorer_metadata.py)

Inputs:
1. Subset stack.csv (from subset_stack) which contains only the metadata for start date to end date, using a ‘1’ in the output directory to specify files that need to be processed
2. Directory for the burn probabilities for this ARD tile
3. Directory for the burn classifications for this ARD tile
4. Acquisition-based metadata template

Outputs:
1. Acquisition level XML metadata files for both burn probability and burn classification files for each of the acquisitions to be processed for the current subset stack

Description:
This script finds the acquisitions in the subset stack that are to be output for the forward processing. The script uses the input XML file for the SR product as the starting point for the output acquisition-based metadata, then appends the burn probability and burn classification bands to the original XML file. The new XML file is written to both the burn probability directory and burn classification directory.

A change in processing for the forward processing writes only the XML file, using _BA.xml as the extension, to the burn probability directory. There isn't a need to duplicate this XML file in multiple directories.

3.9 Generating PNG browse files (scene_visuals.py)

Inputs:
1. Subset stack.csv (from subset_stack) which contains only the metadata for start date to end date, using a ‘1’ in the output directory to specify files that need to be processed
2. Directory location of the multiband GeoTIFF files for this ARD tile
3. Directory for the burn probabilities for this ARD tile
4. Colormap file for the burn probability images
5. Directory for the burn classifications for this ARD tile
6. Colormap file for the burn classification images
7. Directory for output PNG images

Outputs:
1. Quick-look product (containing plots for raw Landsat bands 7-4-2, burn probability, and burn classification) for each of the acquisitions to be processed for the current subset stack
**Description:**
This script finds the acquisitions in the subset stack that are to be output for the forward processing. The script uses the burn probability color map to create a Red Green and Blue Color Value (RGB) burn probability plot. Similarly, it uses the RGB burn classification color map to create the quick-look plot for the burn classification. Next SWIR2, NIR, and Green bands are used from the input multi-band GeoTIFF to generate a false color plot. All three are plotted on a PNG quick-look product with associated title, subtitles, and axis information.
Section 4  Citation Information

There are no restrictions on the use of Landsat Science Products. It is not a requirement of data use, but the following citation may be used in publication or presentation materials to acknowledge the USGS as a data source and to credit the original research.

*Landsat Collection 2 Level 3 Burned Area (BA) Science Products courtesy of the U.S. Geological Survey.*


Reprints or citations of papers or oral presentations based on USGS data are welcome to help the USGS stay informed of how data are being used. These can be sent to the contact information listed below:

USGS EROS Customer Services
Email: custserv@usgs.gov
### Appendix A  Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>Algorithm Description Document</td>
</tr>
<tr>
<td>ARD</td>
<td>Analysis Ready Data</td>
</tr>
<tr>
<td>BA</td>
<td>Burned Area</td>
</tr>
<tr>
<td>BAI</td>
<td>Burned Area Index</td>
</tr>
<tr>
<td>BC</td>
<td>Burn Classification</td>
</tr>
<tr>
<td>BP</td>
<td>Burn Probability</td>
</tr>
<tr>
<td>BT</td>
<td>Brightness Temperature</td>
</tr>
<tr>
<td>C2</td>
<td>Collection 2</td>
</tr>
<tr>
<td>CCB</td>
<td>Configuration Control Board</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CR</td>
<td>Change Request</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values</td>
</tr>
<tr>
<td>DFCB</td>
<td>Data Format Control Book</td>
</tr>
<tr>
<td>DPAS</td>
<td>Data Processing and Archive System</td>
</tr>
<tr>
<td>EROS</td>
<td>Earth Resource Observation and Science</td>
</tr>
<tr>
<td>ESPA</td>
<td>EROS Science Processing Architecture</td>
</tr>
<tr>
<td>ETM+</td>
<td>Enhanced Thematic Mapper Plus</td>
</tr>
<tr>
<td>EVI</td>
<td>Enhanced Vegetation Index</td>
</tr>
<tr>
<td>Fmask</td>
<td>Function of Mask</td>
</tr>
<tr>
<td>GBR</td>
<td>Gradient Boosted Regression</td>
</tr>
<tr>
<td>GBRM</td>
<td>Gradient Boosted Regression Model</td>
</tr>
<tr>
<td>GEMI</td>
<td>Global Environmental Monitoring Index</td>
</tr>
<tr>
<td>GeoTIFF</td>
<td>Georeferenced Tagged Image File Format</td>
</tr>
<tr>
<td>L3</td>
<td>Level 3</td>
</tr>
<tr>
<td>LSDDS</td>
<td>Land Satellites Data System</td>
</tr>
<tr>
<td>MIRBI</td>
<td>Mid-Infrared Burn Index</td>
</tr>
<tr>
<td>NBRT</td>
<td>Normalized Burn Ratio Thermal</td>
</tr>
<tr>
<td>NDVI</td>
<td>Normalized Difference Vegetation Index</td>
</tr>
<tr>
<td>NIR</td>
<td>Near-infrared</td>
</tr>
<tr>
<td>OLI</td>
<td>Operational Land Imager</td>
</tr>
<tr>
<td>PIXELQA</td>
<td>Pixel Quality Assessment</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assessment</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green and Blue Color Value</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>SAVI</td>
<td>Soil Adjusted Vegetation Index</td>
</tr>
<tr>
<td>SR</td>
<td>Surface Reflectance</td>
</tr>
<tr>
<td>SWIR</td>
<td>Shortwave Infrared</td>
</tr>
<tr>
<td>TC</td>
<td>Tasseled Cap</td>
</tr>
<tr>
<td>TIRS</td>
<td>Thermal Infrared Sensor</td>
</tr>
<tr>
<td>TM</td>
<td>Thematic Mapper</td>
</tr>
<tr>
<td>TOA</td>
<td>Top of Atmosphere</td>
</tr>
<tr>
<td>USGS</td>
<td>U.S. Geological Survey</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
References

Please see https://www.usgs.gov/core-science-systems/nli/landsat/landsat-acronyms for a list of acronyms used in this document.
